DOI: 10.6060/ivecofin.2026671.770
УДК: 004.82

СТРАТЕГИИ ФОРМИРОВАНИЯ ОБУЧАЮЩИХ ВЫБОРОК ДЛЯ ТРАНСФОРМЕРНОЙ МОДЕЛИ ИЗВЛЕЧЕНИЯ РЕЛЯЦИОННЫХ ТРОЕК RESC

А.В. Кузьменко, В.С. Киреев

Андрей Владимирович Кузьменко* (ORCID 0009-0007-5900-6676), Василий Сергеевич Киреев (ORCID 0000-0001-6315-163X)
Национальный исследовательский ядерный университет «МИФИ», Каширское ш., 31, Москва, 115409, Россия
E-mail: andrey_kuzmenko2907@mail.ru*, vskireev@mephi.ru

Настоящая статья посвящена исследованию методологии построения графов знаний (ГЗ), которые являются одним из ключевых инструментов структурированного представления семантической информации в таких прикладных областях, как рекомендательные системы, информационный поиск и вопросно-ответные системы. Фундаментальной задачей в процессе автоматизированной генерации ГЗ является извлечение реляционных троек (субъект-отношение-объект), представляющих собой формализованное описание взаимосвязей между выделенными сущностями.

Целью данной работы является комплексный анализ влияния объема и формата обучающих данных на производительность модели RESC, предложенной ранее авторами для решения указанной задачи. Основное внимание уделяется оценке эффективности подхода, основанного на использовании частично размеченных данных для обучения, что позволяет сократить зависимость от трудоемкого процесса полной аннотации.

Экспериментальная часть исследования выполнена на общедоступном корпусе NYT-11, широко применяемом для валидации методов извлечения отношений. В ходе работы были выявлены и проанализированы ключевые факторы, детерминирующие итоговое качество извлечения, такие как степень полноты разметки и стратегия формирования обучающих выборок. На основе проведенного анализа предложены и эмпирически обоснованы практические рекомендации по оптимизации процесса подготовки данных для обучения моделей извлечения реляционных троек, позволяющие достигать высокой точности при сокращении требований к объему полностью размеченных примеров.

Ключевые слова: реляционная тройка, нейронная сеть, обработка естественного языка, трансформеры, графы знаний.

TRAINING SAMPLE FORMATION STRATEGIES FOR THE TRANSFORMER-BASED RESC MODEL IN RELATIONAL TRIPLE EXTRACTION

A.V. Kuzmenko, V.S. Kireev

Andrey V. Kuzmenko* (ORCID 0009-0007-5900-6676), Vasiliy S. Kireev (ORCID 0000-0001-6315-163X)
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe sh., 31, Moscow, 115409, Russia
E-mail: andrey_kuzmenko2907@mail.ru*, vskireev@mephi.ru

This article explores the methodology for constructing knowledge graphs (KGs), which are a key tool for the structured representation of semantic information in application areas such as recommender systems, information retrieval, and question-answering systems. A fundamental task in the automated generation of KGs is the extraction of relational triples (subject-relationship-object), which represent a formalized description of the relationships between the extracted entities.

The goal of this paper is to comprehensively analyze the impact of volume and format of the training data on the performance of the RESC model, previously proposed by the authors for solving this prob-

lem. The focus is on evaluating the effectiveness of an approach based on the use of partially labeled training data, which reduces the reliance on the labor-intensive process of full annotation.

The experimental portion of the study has been conducted using the publicly available NYT-11 corpus, which is widely used for validating relation extraction methods. The study has identified and analyzed key factors determining the final quality of data extraction, such as the degree of annotation completeness and the strategy for generating training samples. Based on this analysis, practical recommendations for optimizing the data preparation process for training relational triple extraction models are proposed and empirically substantiated, enabling high accuracy while reducing the requirements for the volume of fully annotated examples.

Keywords: relational triple, neural network, natural language processing, transformers, knowledge graphs.

Для цитирования:
Кузьменко А.В., Киреев В.С. Стратегии формирования обучающих выборок для трансформерной модели извлечения реляционных троек RESC. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2026. № 01(67). С. 158-166. DOI: 10.6060/ ivecofin.2026671.770

For citation:
Kuzmenko A.V., Kireev V.S. Training sample formation strategies for the transformer-based RESC model in relational triple extraction. *Ivecofin.* 2026. N 01(67). P. 158-166. DOI: 10.6060/ivecofin.2026671.770 (in Russian)

ВВЕДЕНИЕ

Извлечение информации является фундаментальной задачей обработки текстов на естественном языке. Одним из наиболее востребованных форматов представления извлеченной информации является граф знаний [1]. Представленная в таком формате информация является базой для построения поисковых, рекомендательных и вопросно-ответных систем.

В эпоху цифровой трансформации и экспоненциального роста объемов неструктурированных текстовых данных (новостные ленты, научные публикации, корпоративная документация, социальные медиа) потребность в автоматических, точных и масштабируемых методах структурирования информации становится критически важной. Графы знаний, выступая семантическим каркасом для больших данных, позволяют не только хранить факты, но и выявлять скрытые связи, что является основой для систем искусственного интеллекта нового поколения, включая сложные сценарии логического вывода и предиктивной аналитики [2]. Параллельно графовые модели допускают оперирование более дискретными элементами – токенами (сублексическими единицами), обеспечивая репрезентацию семантико-синтаксических отношений между ними [3].

Ключевым технологическим ограничением для масштабируемого применения графов знаний является проблема их автоматической генерации из неструктурированных текстовых данных [4], а именно задача извлечения реляционных троек. Преодоление этого барьера позволяет рассматривать графы знаний как высокоструктурированную альтернативу векторным представлениям (эмбеддингам) в архитектурах RAG (Retrieval-Augmented Generation) [5].

Современные методы решения данной задачи демонстрируют устойчивый компромисс между эффективностью и ресурсозатратностью. С одной стороны, глубокие нейросетевые модели достигают высоких метрик качества, однако их обучение требует значительных объемов дорогостоящих размеченных данных. С другой стороны, менее ресурсоемкие подходы, как правило, не обеспечивают сопоставимого уровня точности и полноты извлечения связей.

Таким образом, исследование влияния объема и формата обучающих данных на эффективность моделей извлечения троек приобретает высокую научную и практическую значимость. Оптимизация этого процесса позволяет снизить стоимость и время развертывания систем, основанных на знаниях, и сделать технологии семантического анализа доступными для прикладных задач в условиях ограниченных ресурсов. Данная работа, фокусируясь на масштабировании обучения модели RESC, направлена на решение этой актуальной проблемы, предлагая стратегии для достижения высокого качества при минимизации затрат на подготовку данных.

ПОСТАНОВКА ЗАДАЧИ

Извлечение реляционных троек можно сформировать как задачу поиска тройных множеств $Y = \{(s_1, r_1, o_1), (s_2, r_2, o_2), \dots, (s_k, r_k, o_k)\}$, в текстовом документе $X = \{x_1, x_2, \dots, x_m\}$. Математическая постановка задачи выглядит следующим образом:

$$p(Y|X,\theta) = p(k|X)\prod_{i=1}^{K} p\left(Y_i \middle| X, Y_{i\neq j}, \theta\right), \quad (1)$$

где $s_\text{i} = [x_{s_i}^{start}, \dots, x_{s_i}^{end}]$ – субъект отношения;

$o_\text{i} = [x_{o_i}^{start}, \dots, x_{o_i}^{end}]$ –объект отношения;

r_i – тип семантической связи между s_i и o_i из фиксированного набора $r = \{r_1, r_2, \dots, r_n\}$;

$p(k|X)$ – моделирует размер набора реляционных троек;

$\text{p}\left(\text{Y}_\text{i} \middle| \text{X}, \text{Y}_{\text{i}\neq\text{j}}, \theta\right)$ – моделирует тройку Y_i, при условии, что она связана не только с предложением, но и с другими тройками $Y_{i\neq j}$;

θ – параметры модели.

Важным аспектом формализации задачи извлечения реляционных троек является учёт взаимозависимостей между сущностями и отношениями в рамках одного документа. В отличие от классических конвейерных подходов, где эти задачи решаются независимо, модель RESC интегрирует этапы извлечения сущностей и идентификации отношений, что позволяет учитывать контекстные связи и улучшать итоговое качество. Данная работа фокусируется на исследовании влияния объёма и структуры обучающих данных на сходимость и обобщающую способность модели, что является ключевым для её практического применения в условиях, ограниченных размеченных данных.

В работе [6] предлагается систематизация методов извлечения реляционных троек на: (1) конвейерные методы [7-9] – системы с двумя четко разграниченными стадиями извлечения сущностей и идентификации отношений между парой сущностей; (2) табличные [10, 11], в основе которых лежит идея заполнения треугольной таблицы отношений; (3) методы основанные разработке специальной маркировки тегов, позволяющей извлекать реляционные тройки напрямую [12–14]; (4) генеративные, методы основанные на генерации последовательности состоящей из элементов реляционных троек [15, 19].

ОПИСАНИЕ РАБОТЫ RESC

Рассматриваемая архитектура RESC является представителем конвейерной парадигмы. Возможны различные варианты ее реализации [20], поскольку многие части ее можно подвергнуть архитектурным изменениям. В настоящей работе рассматривается реализация, представленная на рис. 1.

У RESC, как и в других представителей конвейерных систем можно выделить две стадии. Первая является вспомогательной и необходима для извлечения сущностей кандидатов. Она реализуется на базе модели BERT [21], которая выступает в роли кодировщика последовательности текстовых токенов $X = \{x_1, x_2, \dots, x_m\}$ в последовательность векторов $H = \{h_1, h_2, \dots, h_m\}$. После, для каждого из векторов при помощи механизма принятия решения, реализованного в виде полносвязной нейронной сети с одним выходным слоем, определяется принадлежность к одному из трех классов: (0) исходный токен не является сущностью; (1) исходный токен является началом сущности; (2) исходный токен является последним в сущности.

Архитектура RESC сочетает преимущества предобученных трансформерных моделей и механизма кросс-внимания, что позволяет эффективно обрабатывать длинные последовательности и выявлять семантические связи между сущностями.

Стадия идентификации отношений выполняется в несколько этапов. Из векторных представлений $H = \{h_0, h_1, \dots, h_k\}$ формируются все возможные подпоследовательности вида $\left\{h_{start}^{sub} + e_0, \dots, h_{end}^{sub} + e_0, h_{start}^{obj} + e_1, \dots, h_{end}^{obj} + e_1\right\}$, где h_i^{sub} – вектор-кандидат в субъект отношения; h_j^{sub} – вектор-кандидат в объект отношения; e_0, e_1 - позиционные векторы. Сформированная последовательность векторов, вместе с последовательностью векторов $Q = \{q_0, q_1, \dots, q_n\}$, отражающей все возможные типы отношений передаются в блок «Cross Attention» в котором сформировывают информативные сигналы $Q' = \{q'_0, q'_1, \dots, q'_n\}$. Информативные сигналы содержат информацию наличие связей из представленной группы. Эти сигналы передаются в механизм принятия решений, реализованный в виде полносвязной нейронной сети с одним выходным слоем.

Модель извлечения реляционных троек RESC решает две связанные задачи. Первая – извлечение сущностей кандидатов. Вторая - идентификация отношения между парой сущностей. Оптимизация двух задач в RESC производится путем минимизации комбинированной функции потерь:

$$\mathcal{L} = \mathcal{L}_E + \lambda\mathcal{L}_R \rightarrow min. \quad (2)$$

Слагаемое $\mathcal{L}_E$ отвечает за извлечение сущностей-кандидатов. Для каждого из токенов

входной последовательности определяется метка класса из тройного набора: 0 – не является сущностью, 1 – начало сущности, 2 – продолжение сущности. Эта компонента является кроссэнтропийной функцией потерь, которую можно вычислить следующим образом:

$$\mathcal{L}_E = -\frac{1}{B_e \cdot S}\sum_i^{B_e}\sum_j^{S} log\,\frac{\exp\left(h_i^e\right)\cdot 1[y_i^e \neq ignore\_index]\cdot 1[y_i^e \neq pad\_index]}{\sum_{k=0}^{2}\exp\left(h_k^e\right)}, \quad (3)$$

где $1[y_i^e \neq ignore\_index]$ – игнорирование не целевых классов;

$1[y_i^e \neq pad\_index]$ – игнорирование вспомогательных классов, используемых для пакетирования примеров;

B_e – размер пакета;

S – длина максимальной последовательности в пакете.

Слагаемое $\mathcal{L}_R$ направлено на определение типа связей между парой сущностей. По своей структуре она является бинарной кроссэнтропией, выражение для которой можно записать в следующем виде:

$$\mathcal{L}_R = -\sum_{i}^{B_r}\sum_{j}^{N_i}\sum_{k}^{N_r}\{y_k^r \log \hat{y}_k^r + \\ +(1-y_k^r)\log(1-\hat{y}_k^r), \quad (4)$$

где B_r – размер пакета текстовых документов;

N_i – количество размещений, которые можно вычислить как $\frac{n!}{(n-2)!}$, где n – количество сущностей, определенных в документе;

N_r – количество отношений;

y_k^r – метка класса k: 0 – пара сущностей не связаны этим отношением, 1 – пара сущностей связана этим отношением;

$\hat{y}_k^r$ – оценка полученных вероятности принадлежности к классу k.

Чтобы контролировать влияния каждого из слагаемых вводится дополнительный гиперпараметр λ, который взвешивает значимость каждой из составляющих.

Обучение модели предлагается производить в две стадии. На первой стадии корректируются все веса модели кроме весов кодировщика. Это обеспечивает независимость обучения задач извлечения сущностей и идентификации отношений. Первая стадия выполняется пока значения показателей качества задач улучшаются. Вторая стадия состоит из двух этапов: (1) корректируются все веса модели, включая кодировщик с высокой скоростью обучения; (2) корректируются все веса модели, включая кодировщик с низкой скоростью обучения. Во всех экспериментах используется оптимизатор AdamW [20].

Список основных гиперпараметров выглядит следующим образом: B_e = 32; B_r = 32; λ = 1, скорость обучения (1 стадия) = 1e-4, скорость обучения (2 стадия, 1 этап) = 1e-4; скорость обучения (2 стадия, 2 этап) = 1e-5; коэффициент регуляризации=1e-2.

Предложенный двухэтапный подход к обучению модели RESC позволяет раздельно оптимизировать параметры для задач извлечения сущностей и идентификации отношений, что снижает риск переобучения и улучшает сходимость. Использование взвешенной функции потерь с гиперпараметром λ обеспечивает баланс между этими задачами и позволяет адаптировать модель под конкретные требования к точности или полноте. Эксперименты с различными стратегиями формирования пакетов данных демонстрируют устойчивость модели к изменению условий обучения и её применимость в сценариях с частичной разметкой.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ И ИХ ОБСУЖДЕНИЕ

Для экспериментов был использован открытый набор данных NYT [23]. Он содержит 24 уникальных отношений между сущностями, более 177 тыс. отношений и более 121 тыс. сущностей. Набор данных разбит на 56196/5000/5000 примеров на тренировочную, валидационную и тестовую части соответственно.

Направление исследований нацелено на определение достаточности объема обучающих данных путем масштабирования их количества. Обучающая выборка случайным образом перемешивается и отсекаются фрагменты из ряда {100, 500, 1000, 2000, 5000, 10000, 20000, 40000, 56196}, где числа в скобках означают размер нового обучающего фрагмента. Важно отметить, что при отсечении не производится повторное перемешивание. Это означает, что примеры, которые попали в меньшую подвыборку в полном объеме будут присутствовать в большей подвыборке. Целенаправленно не используется стратегия стратифицированного разбиения по типам отношений. Это сделано для моделирования более жестких условий обучения модели.

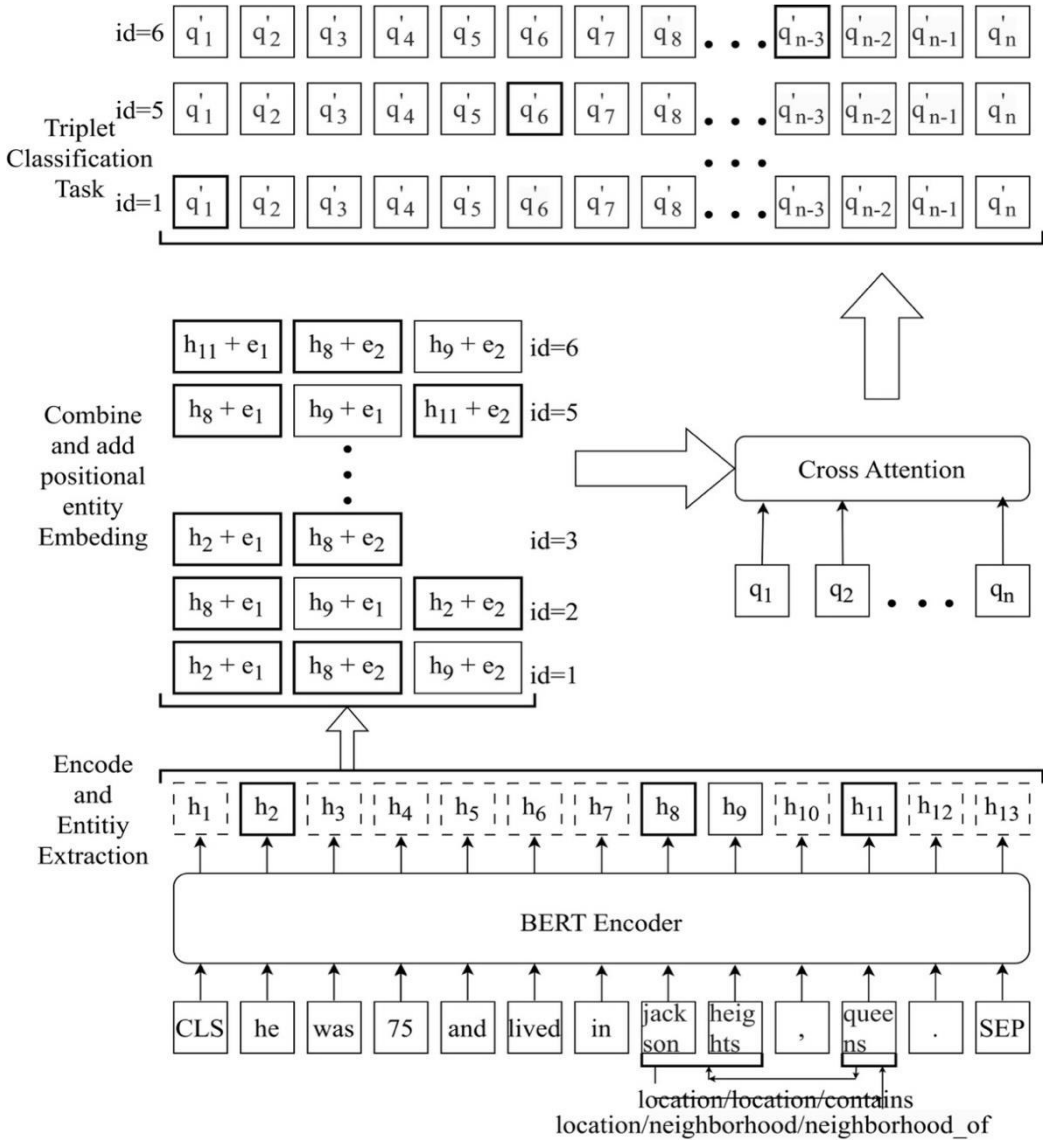


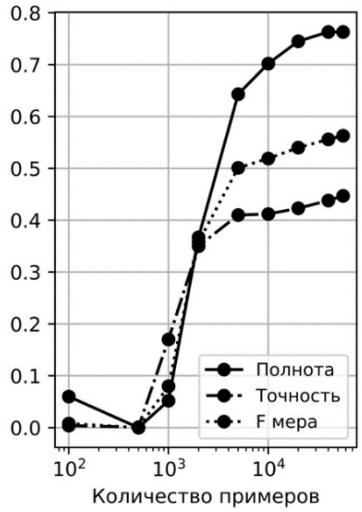
Рисунок 1. Архитектура RESC. Обучение модели
Figure 1. RESC Architecture. Model training

В экспериментах производится наблюдение за тремя величинами: (1) качество извлечение сущностей; (2) качество идентификации отношений при условии передачи набора истинных сущностей; (3) качество извлечения реляционных троек от начала до конца. Замеряются стандартные показатели качества: полнота (recall), точность (precision), и их среднее гармоническое (f-мера). На рис. 2а и 2б показано качество извлечения сущностей на первой и второй стадии обучения соответственно. На рис. 3а и 3б – качество идентификации отношений при передаче истинных сущностей. На рис. 4а и 4б – качество от начала до конца.

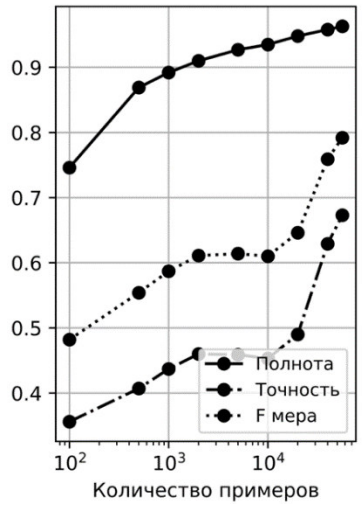
Можно наблюдать, что задача идентификации отношения для истинных сущностей оказывается существенно проще, чем задача извлечения сущностей. Даже при 100 примерах модель способна определять отношение с f мерой почти 0.6 на второй стадии обучения, в то же время извлечение сущностей на таком объеме данных показывает величину f-меры около 0.36.

Вторая интересная закономерность проявляется при сравнении результатов, полученных на небольшом объеме обучающих данных (до нескольких тысяч объектов), между первой и второй стадий обучения. Графики на рис. 2а, 3а демонстрируют полную неработоспособность моделей. Однако на рис. 2б, 3б можно пронаблюдать приемлемые показатели качества.

Третьей, по мнению авторов, наиболее важной закономерностью является довольно слабая зависимость между f-мерой извлечения сущностей и f-мерой извлечения реляционных троек. Результаты демонстрируют, что для наилучших показателей гораздо важнее на первой стадии извлекать сущности с высокими показателями полноты (recall), а не f-меры.



а)

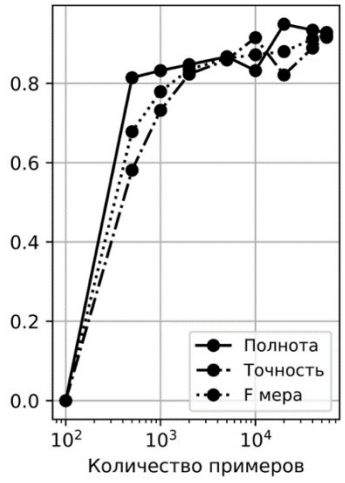


б)

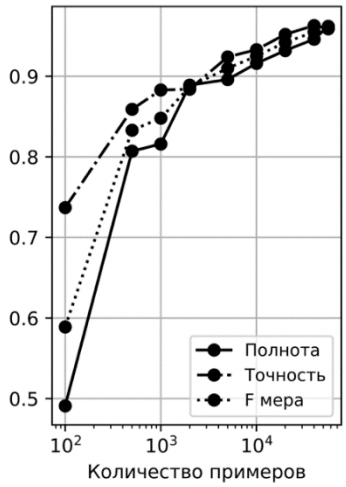
Рисунок 2. Качество извлечения сущностей
а) в конце первой стадии обучения;
б) в конце второй стадии обучения
Figure 2. Quality of entity extraction
a) at the end of the first stage of training;
b) at the end of the second stage of training

При совместном обучении двум задачам возникает вопрос выбора стратегии формирования обучающих примеров. В данной работе рассматривалось два варианта: (1) со случайным пакетом отношений, (2) с фиксированным пакетом отношений.

Рассмотрим первый вариант. Пусть дан пакет последовательностей $\boldsymbol{X} = \{X_1, X_2, \dots, X_{B_e}\}$, где B_e размер пакета. Тогда для каждого примера X_i необходимо определить m_i меток токенов, где m_i – длина -ой последовательности. Пусть известно, что в данном примере присутствует три сущности. Тогда для каждой из шести подпоследовательностей (количество сочетаний) необходимо предсказать r бинарных меток. В таком формате размерность последовательностей для извлечения сущностей всегда фиксирована и равна размеру пакета, а размерность подпоследовательностей для идентификации отношений случайна и зависит от количества сущностей у примеров, попавших в пакет.



а)

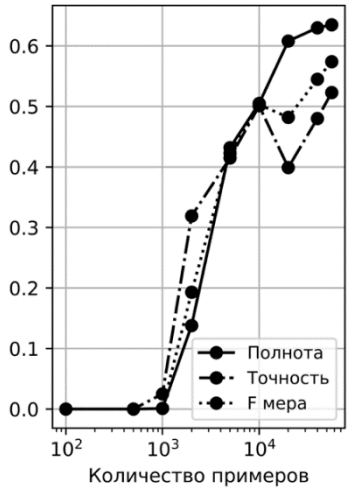


б)

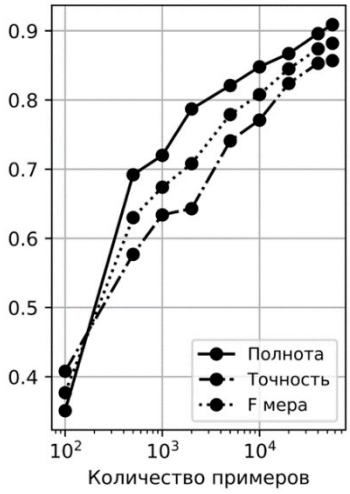
Рисунок 3. Качество идентификации отношений
а) в конце первой стадии обучения;
б) в конце второй стадии обучения
Figure 3. Quality of identification of the relationship
a) at the end of the first stage of training;
b) at the end of the second stage of training

Во втором варианте размер пакета последовательностей для извлечения сущностей и размер пакета подпоследовательностей для идентификации отношений равны. Это реализуется с помощью двух независимых баз примеров. Каждый пример из первой базы содержит последовательность токенов и соответствующую последовательность меток сущностей. Каждый пример второй базы содержит последовательность токенов, подпоследовательность для идентификации отношения и метки отношений соответствующие текущей подпоследовательности. Поскольку размеры вто-

рой базы всегда больше первой, объемы первой базы искусственно увеличиваются. Это производится путем случайного выбора с возвращением K примеров из первой базы, где K -размер второй базы. Для большего разнообразия такая процедура производится в начале каждой эпохи обучения.



а)



б)

Рисунок 4. Качество извлечения реляционных троек
а) в конце первой стадии обучения;
б) в конце второй стадии обучения
Figure 4. The quality of extracting relational triples
a) at the end of the first stage of training;
b) at the end of the second stage of training

Преимущество первой стратегии – это переиспользование результатов кодировщика. Достаточно один раз преобразовать последовательность токенов в последовательность векторов, которая далее используется для двух задач. Недостаток такой стратегии авторы видят в искусственном снижении дисперсии градиента, что может негативно отразиться в процессе обучение. Это связано с тем, что при вычислении комбинированной функции потерь (2) в рамках одного примеры будут одни и те же составляющие: метки токенов и множество подпоследовательностей. Для проверки этой гипотезы проводились эксперименты с различными коэффициентами взвешивания задач $\lambda = \{0.1, 0.25, 0.5, 1, 2, 5, 10\}$. На рис. 5 показаны значение f-меры в конце второй стадии обучения при различных λ двух стратегий формирования обучающих примеров. Можно пронаблюдать, что оптимальные значения λ отличаются у двух стратегий, однако пиковые показатели качества практически идентичны.

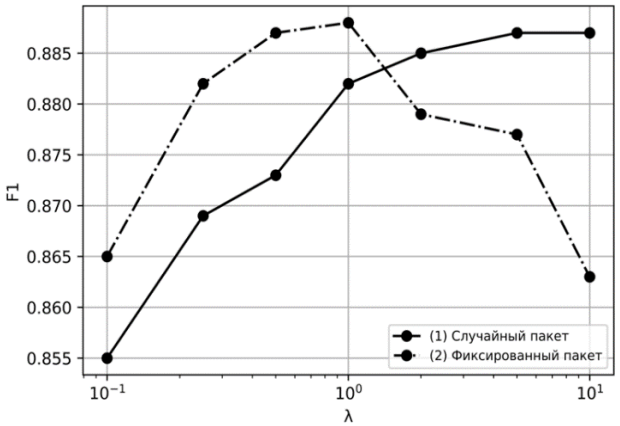


Рисунок 5. Сравнение стратегий обучения
Figure 5. Comparison of training strategies

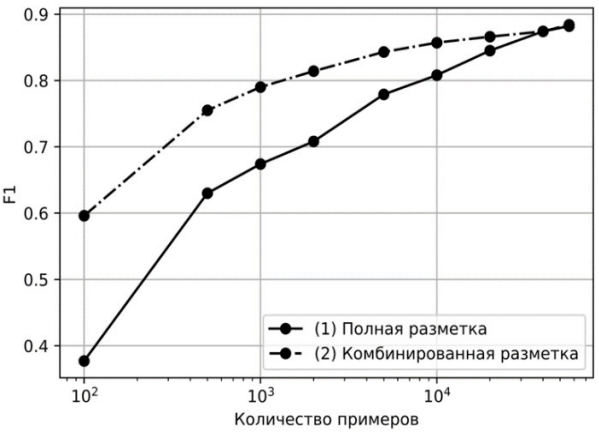


Рисунок 6. Сравнение типов разметки
Figure 6. Comparison of markup types

Хотя стратегия с выравниванием обучающих пакетов не улучшает показатели качества, она позволяет обучать модель в условиях смешанных примеров, когда в части примерах отсутствует разметка отношений, а есть только разметка сущностей. Аналогично методологии проверки достаточности данных, проводятся эксперименты с масштабированием размеченных отношений. Для этого при обучении используется разметка сущностей всего набора, при этом набор размеченных отношений выбирается из ряда {100, 500, 1000, 2000, 5000, 10000, 20000, 40000, 56196}. На рис. 6 иллюстрировано сравнение f меры извлече-

ния реляционных троек при масштабировании полной разметки (сущностей и отношений) и масштабировании комбинированной разметки (размечены все сущности и только часть отношений). Как видно из сравнения, для эффективного обучения RESC не требуется большое количество примеров с размеченными тройками. Гораздо эффективнее делать упор на разметку сущностей. Это очень важно в практических приложениях, поскольку разметка реляционных троек гораздо более трудоемкий и дорогостоящий процесс, чем разметка отдельных сущностей.

Наиболее значимым результатом является выявление зависимости между полнотой извлечения сущностей и итоговым качеством извлечения троек, что указывает на важность приоритизации этой задачи на ранних этапах обучения. Предложенная стратегия обучения с комбинированной разметкой позволяет существенно снизить затраты на аннотирование данных без значительной потери качества, что имеет высокую практическую ценность для развёртывания систем в промышленной среде.

ЗАКЛЮЧЕНИЕ

В данной работе проведено исследование обучения трансформерной модели RESC. Проведены эксперименты с масштабированием обучающего набора данных. В результате исследования были выявлены ключевые закономерности качества извлечения реляционных троек от размера и структуры обучающей выборки. Кроме того, были предложены две различные стратегии формирования обучающих примеров. Первая позволяет более эффективно, с точки зрения вычислительных затрат, производить обучение модели. Вторая открывает возможности обучать модель на комбинированных примерах.

Результаты исследования подтверждают, что модель RESC является эффективным инструментом для извлечения реляционных троек, особенно в условиях, ограниченных размеченных данных. Предложенные стратегии масштабирования обучающей выборки и формирования примеров позволяют оптимизировать процесс обучения и адаптировать модель под различные практические задачи. В перспективе планируется исследование применения RESC в предметно-ориентированных корпусах [24], а также интеграция модели в комплексные системы управления знаниями для поддержки принятия решений в реальном времени.

Авторы заявляют об отсутствии конфликта интересов.
The authors declare no conflict of interest.

ЛИТЕРАТУРА

1. **Zhang J.C. et al.** A review of recommender systems based on knowledge graph embedding. *Expert Systems with Applications*. 2024. N 250. P. 123876. DOI: 10.1016/j.eswa.2024.123876.
2. **Ji S., Pan S., Cambria E., Marttinen P., Yu P.S.** A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*. 2022. N 33(2). P. 494-514. DOI: 10.1109/TNNLS.2021.3070843.
3. **Зимнуров М.Ф., Астраханцева И.В.** Методология создания многосвязных структур данных с применением LLM в рабочих проектах. *Современные наукоемкие технологии. Региональное приложение*. 2025. № 1(81). С. 76-83. DOI: 10.6060/snt.20258101.0009.
4. **Dai Y., Wang S., Xiong N., Guo W.** A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics*. 2020. N 9(5). P. 750. DOI: 10.3390/electronics9050750.
5. **Герасимов А.С., Бобков. С.П.** Разработка самообучающейся системы обработки естественного языка с динамической реорганизацией знаний. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2025. № 3(65). С. 77-84. DOI: 10.6060/ivecofin.2025653.734.
6. **Кузьменко А.В., Киреев В.С.** Классификация методов извлечения реляционных троек из текстов на естественном языке. *В сб. «Нейроинформатика-2023» XXV Межд. н.-техн. конференции.* М.: НИЯУ МИФИ. 2023. С. 302-311.
7. **Zenelnko D., Aone C., Richardella A. et al**. Kernel methods for relation extraction. *Journal of Machine Learning Research*. 2003. N. 3. P. 1083-1106. DOI: 10.5555/944919.944964.
8. **Chan S.Y., Roth D.** Exploiting syntactico-semantic structures for relation extraction. *Materials of the 49th Annu. Meet. Assoc. Comput. Linguistics: Human Lang. Technol*. 2011. P. 551-560.

REFERENCES

1. **Zhang J.C. et al.** A review of recommender systems based on knowledge graph embedding. *Expert Systems with Applications*. 2024. N 250. P. 123876. DOI: 10.1016/j.eswa.2024.123876.
2. **Ji S., Pan S., Cambria E., Marttinen P., Yu P.S.** A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*. 2022. N 33(2). P. 494-514. DOI: 10.1109/TNNLS.2021.3070843.
3. **Zimnurov M.F., Astrakhantseva I.A.** Usage of large language models for building multi-vectored and multi-linked data model of work process. *Modern high technologies. Regional application*. 2025. N 1(81). P. 76-83. DOI 10.6060/snt.20258101.0009. (in Russian).
4. **Dai Y., Wang S., Xiong N., Guo W.** A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics*. 2020. N 9(5). P. 750. DOI: 10.3390/electronics9050750.
5. **Gerasimov A.S., Bobkov. S.P.** Development of a self-learning system for natural language processing with dynamic knowledge reorganization. *Ivecofin*. 2025. N 3(65). P. 77-84. DOI: 10.6060/ivecofin.2025653.734. (in Russian).
6. **Kuzmenko A.V., Kireev V.S.** Classification of methods for extracting relational triples from natural language texts. *Materials of the XXV International Scientific and Technical Conference "Neuroinformatics-2023"*. Moscow: NRNU MEPhI. 2023. P. 302-311. (in Russian).
7. **Zenelnko D., Aone C., Richardella A. et al**. Kernel methods for relation extraction. *Journal of Machine Learning Research*. 2003. N. 3. P. 1083-1106. DOI: 10.5555/944919.944964.
8. **Chan S.Y., Roth D.** Exploiting syntactico-semantic structures for relation extraction. *Materials of the 49th Annu. Meet. Assoc. Comput. Linguistics: Human Lang. Technol*. 2011. P. 551-560.

9. **Zhong Z., Chen D.A.** A frustratingly easy approach for entity and relation extraction. *Materials of the Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol*. 2021. P. 50-61.
10. **Zhang M., Zhang Y., Fu G.** End-to-End Neural Relation Extraction with Global Optimization. *Materials of the Conf. Empirical Methods in Natural Lang. Process*. 2017. P. 1730-1740.
11. **Gupta P., Schütze H., Andrassy B.** Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. *Materials of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016. P. 2537-2547.
12. **Wei Z., Su J., Wang Y., et al.** A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. *Materials of the 58th Annu. Meet. Assoc. Comput. Linguistics*. 2020. P. 1476-1488.
13. **Zheng S., Wang F, Bao H. et al.** Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *Materials of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017. N. 1. P. 1227-1236.
14. **Dai D., Xiao X., Lyu Y. et al.** Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling. *Materials of the AAAI Conference on Artificial Intelligence*. 2019. P. 6300-6308.
15. **Sui D., Chen Y., Liu K., et al.** Joint entity and relation extraction with set prediction networks. *IEEE Trans. Neural Networks Learn. Syst*. 2024. P. 12784-12795. DOI: 10.1109/TNNLS.2023.3264735.
16. **Zeng X., Zeng D., He S. et al.** Extracting relational facts by an end-to-end neural model with copy mechanism. *Materials of the 56th Annu. Meet. Assoc. Comput. Linguistics*. 2018. N 1. P. 506-514.
17. **Zeng D., Zhang H., Liu Q.** CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. *Materials of the AAAI Conference on Artificial Intelligence*. 2020. N 34(05). P. 9507-9514.
18. **Yuan C., Xie Q., Ananiadou S.** Zero-shot Temporal Relation Extraction with ChatGPT. *Materials of the 22nd Workshop Biomedical Natural Language Processing and BioNLP Shared Tasks*. 2023. P. 92-102.
19. **Hu Y., Ameer I., Zou X.** Zero-shot Clinical Entity Recognition using ChatGPT. https://www.researchgate.net/publication/369623655_Zero-shot_Clinical_Entity_Recognition_using_ChatGPT.
20. **Delvin J., Chang M.W., Lee K., Toutanova K.** BERT: pre-training of deep bidirectional transformer for language understanding. *Materials of the Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol*. 2019. P. 4171-4186.
21. **Кузьменко А.В., Киреев В.С.** Абляционное исследование модели извлечения реляционных троек RESC. *Современная наука*. 2025. № 2(2) С. 108-116. DOI: 10.37882/2223-2966.2025.02-2.22.
22. **Loshchilov I., Hutter F.** Decoupled weight decay regularization. https://openreview.net/pdf?id=Bkg6RiCqY7.
23. **Riedal S., Yao L., McCallum A.** Modeling relations and their mentions without labeled text. *Mach. Learn. Knowl. Discov. Databases*. 2010. P. 148-163.
24. **Kun X., Liwei W., Mo Y. et al.** Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. *Materials of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. P. 3156–3161.

9. **Zhong Z., Chen D.A.** A frustratingly easy approach for entity and relation extraction. *Materials of the Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol*. 2021. P. 50-61.
10. **Zhang M., Zhang Y., Fu G.** End-to-End Neural Relation Extraction with Global Optimization. *Materials of the Conf. Empirical Methods in Natural Lang. Process*. 2017. P. 1730-1740.
11. **Gupta P., Schütze H., Andrassy B.** Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. *Materials of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016. P. 2537-2547.
12. **Wei Z., Su J., Wang Y., et al.** A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. *Materials of the 58th Annu. Meet. Assoc. Comput. Linguistics*. 2020. P. 1476-1488.
13. **Zheng S., Wang F, Bao H. et al.** Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *Materials of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017. N. 1. P. 1227-1236.
14. **Dai D., Xiao X., Lyu Y. et al.** Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling. *Materials of the AAAI Conference on Artificial Intelligence*. 2019. P. 6300-6308.
15. **Sui D., Chen Y., Liu K., et al.** Joint entity and relation extraction with set prediction networks. *IEEE Trans. Neural Networks Learn. Syst*. 2024. P. 12784-12795. DOI: 10.1109/TNNLS.2023.3264735.
16. **Zeng X., Zeng D., He S. et al.** Extracting relational facts by an end-to-end neural model with copy mechanism. *Materials of the 56th Annu. Meet. Assoc. Comput. Linguistics*. 2018. N 1. P. 506-514.
17. **Zeng D., Zhang H., Liu Q.** CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. *Materials of the AAAI Conference on Artificial Intelligence*. 2020. N 34(05). P. 9507-9514.
18. **Yuan C., Xie Q., Ananiadou S.** Zero-shot Temporal Relation Extraction with ChatGPT. *Materials of the 22nd Workshop Biomedical Natural Language Processing and BioNLP Shared Tasks*. 2023. P. 92-102.
19. **Hu Y., Ameer I., Zou X.** Zero-shot Clinical Entity Recognition using ChatGPT. https://www.researchgate.net/publication/369623655_Zero-shot_Clinical_Entity_Recognition_using_ChatGPT.
20. **Delvin J., Chang M.W., Lee K., Toutanova K.** BERT: pre-training of deep bidirectional transformer for language understanding. *Materials of the Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol*. 2019. P. 4171-4186.

21. **Kuzmenko A.V., Kireev V.S.** An ablative study of the relational triple extraction model RESC. *Modern science.* 2025. N 2(2). P. 108-116. DOI: 10.37882/2223-2966.2025.02-2.22. (in Russian).

22. **Loshchilov I., Hutter F.** Decoupled weight decay regularization. https://openreview.net/pdf?id=Bkg6RiCqY7.
23. **Riedal S., Yao L., McCallum A.** Modeling relations and their mentions without labeled text. *Mach. Learn. Knowl. Discov. Databases*. 2010. P. 148-163.
24. **Kun X., Liwei W., Mo Y. et al.** Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. *Materials of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. P. 3156–3161.

Поступила в редакцию 17.12.2025
Принята к опубликованию 29.12.2025

Received 17.12.2025
Accepted 29.12.2025