MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 32, NO. 1, 2025
journal homepage: www.mais-journal.ru

ARTIFICIAL INTELLIGENCE

Comparison of Pre-trained Models for Domain-specific Entity Extraction from Student Report Documents

A. V. Melnikova[1], M. S. Vorobeva[1], A. V. Glazkova[1]

DOI: 10.18255/1818-1015-2025-1-66-79

[1]University of Tyumen, Tyumen, Russia

MSC2020: 68T50, 97B40
Research article
Full text in Russian

Received February 11, 2025
Revised February 21, 2025
Accepted February 26, 2025

The authors propose a methodology for extracting domain-specific entities from student report documents in Russian language using pre-trained transformer-based language models. Extracting domain-specific entities from student report documents is a relevant task since the obtained data can be used for various purposes, ranging from the formation of project teams to the personalization of learning pathways. Additionally, automating the document processing workflow reduces the labor costs associated with manual processing. As training material for training models, expert-annotated student report documents were used. These documents were created by students in information technology programs between 2019 and 2022 for project-based, practical disciplines, and theses. The domain-specific entity extraction task is approached as two subtasks: named entity recognition (NER) and annotated text generation. A comparative analysis was conducted among NER encoder-only models (ruBERT, ruRoBERTa), encoder-decoder models (ruT5, mBART), and decoder-only models (ruGPT, T-lite) for text generation. The effectiveness of the models was evaluated using the F1-score, along with an analysis of common errors. The highest F1-score on the test set was achieved by mBART (93.55 %). This model also showed the lowest error rate in domain-specific entity identification during text generation and annotation. The NER models demonstrated a lower tendency for errors but tended to extract domain-specific entities in a fragmented manner. The obtained results indicate the applicability of the examined models for solving the stated tasks, considering the specific requirements of the problem.

Keywords: domain-specific entities; digital footprint; information extraction; natural language processing; pre-trained language models

INFORMATION ABOUT THE AUTHORS

Melnikova, Antonina V.	ORCID iD: 0009-0006-1011-5225. E-mail: a.v.melnikova@utmn.ru Senior Lecturer of the Department of Software of the School of Computer Science
Vorobeva, Marina S.	ORCID iD: 0000-0002-1508-4089. E-mail: m.s.vorobeva@utmn.ru Head of the Department of Software of the School of Computer Science, Associate Professor, PhD
Glazkova, Anna V. (corresponding author)	ORCID iD: 0000-0001-8409-6457. E-mail: a.v.glazkova@utmn.ru Associate Professor of the Department of Software of the School of Computer Science, PhD

Funding: This study was supported by the Ministry of Science and Higher Education of the Russian Federation within the framework of a State assignment (FEWZ-2024-0052).

For citation: A. V. Melnikova, M. S. Vorobeva, and A. V. Glazkova, "Comparison of pre-trained models for domain-specific entity extraction from student report documents", *Modeling and Analysis of Information Systems*, vol. 32, no. 1, pp. 66–79, 2025. DOI: 10.18255/1818-1015-2025-1-66-79.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 32, № 1, 2025
сайт журнала: www.mais-journal.ru

Сравнение предварительно обученных моделей для извлечения предметно-ориентированных сущностей из студенческих отчетных документов

А. В. Мельникова[1], М. С. Воробьева[1], А. В. Глазкова[1]

DOI: 10.18255/1818-1015-2025-1-66-79

[1]Тюменский государственный университет, Тюмень, Россия

УДК 004.912+378.1
Научная статья
Полный текст на русском языке

Получена 11 февраля 2025 г.
После доработки 21 февраля 2025 г.
Принята к публикации 26 февраля 2025 г.

Авторы предлагают методику извлечения предметно-ориентированных сущностей (ПОС) из русскоязычных текстов студенческих отчетных документов с использованием предварительно обученных языковых моделей на основе трансформеров. Извлечение ПОС из студенческих работ представляет собой актуальную задачу, так как полученные данные могут использоваться для различных целей — начиная от формирования проектных групп и заканчивая персонализацией учебных маршрутов, а также автоматизация процесса обработки документов снижает затраты труда на ручную обработку. В качестве материала для дообучения исследуемых моделей использовались размеченные экспертами отчетные документы студентов, обучающихся по направлениям информационных технологий и поступивших в период с 2019 по 2022 год, по проектным, практическим дисциплинам и выпускным квалификационным работам. Задача извлечения ПОС рассматривается как две задачи: идентификация именованных сущностей и генерация размеченного текста. Сравнительный анализ проводился между моделями, основанными исключительно на энкодерах (ruBERT, ruRoBERTa), предназначенными для извлечения именованных сущностей, и моделями, использующими как энкодеры, так и декодеры (ruT5, mBART), а также моделями, базирующимися только на декодерах (ruGPT, T-lite), применяемыми для генерации текста. Для оценки эффективности сравниваемых моделей использовалась F-мера, а также проведен анализ типичных ошибок. Наиболее высокие показатели по F-мере на тестовом наборе данных продемонстрировала модель mBART (93.55 %). Эта же модель показала наименьший уровень ошибок при идентификации ПОС во время генерации текста и разметки. Модели для извлечения именованных сущностей проявляют меньшую склонность к ошибкам, однако имеют тенденцию к фрагментарному выделению ПОС. Полученные результаты свидетельствуют о применимости рассматриваемых моделей для решения поставленных задач с учетом специфики предъявляемых требований.

Ключевые слова: предметно-ориентированные сущности; цифровой след; извлечение информации; обработка естественного языка; предварительно обученные языковые модели

ИНФОРМАЦИЯ ОБ АВТОРАХ

Мельникова, Антонина Владимировна	ORCID iD: 0009-0006-1011-5225. E-mail: a.v.melnikova@utmn.ru Старший преподаватель кафедры программного обеспечения Школы компьютерных наук
Воробьева, Марина Сергеевна	ORCID iD: 0000-0002-1508-4089. E-mail: m.s.vorobeva@utmn.ru Заведующий кафедрой программного обеспечения Школы компьютерных наук, доцент, канд. тех. наук
Глазкова, Анна Валерьевна (автор для корреспонденции)	ORCID iD: 0000-0001-8409-6457. E-mail: a.v.glazkova@utmn.ru Доцент кафедры программного обеспечения Школы компьютерных наук, канд. тех. наук

Финансирование: Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации в рамках государственного задания (FEWZ-2024-0052).

Для цитирования: A. V. Melnikova, M. S. Vorobeva, and A. V. Glazkova, “Comparison of pre-trained models for domain-specific entity extraction from student report documents”, *Modeling and Analysis of Information Systems*, vol. 32, no. 1, pp. 66–79, 2025. DOI: 10.18255/1818-1015-2025-1-66-79.

Введение

При работе с текстовыми документами специалисту важно оценить степень погруженности исполнителя в конкретную область. В зависимости от специфики неструктурированных или полуструктурированных текстовых данных это могут быть элементы предметной области (термины, навыки, специализированные слова, ключевые фразы), встречающиеся в отчетах, вакансиях, описаниях проектов, портфолио, заявках на гранты и других документах [1].

Если рассматривать образовательные данные, на текущий момент в вузах о работах обучающихся накоплен большой объем информации, который все время увеличивается. В рамках учебной деятельности студенты в процессе выполнения заданий формируют свой цифровой след, включающий отчетные документы проектных практикумов, письменных практических заданий, промежуточных результатов по дисциплинам, практических подготовок, выпускных квалификационных работ [2]. Например, из студенческих работ можно получить базу знаний о том, что изучали и использовали студенты ИТ-направлений: технологии, языки программирования, библиотеки, фреймворки, алгоритмы, методы, подходы и др.

Систематизация таких данных очень важна для решения широкого спектра задач — от формирования проектных команд до персонализации образовательных траекторий. Анализ цифрового следа студентов может служить основой для рекомендаций по улучшению учебных планов и курсов, анализа востребованности навыков и знаний, мониторинга успеваемости и выявления направлений развития в соответствии актуальными требованиями индустрии.

При постоянном увеличении объема текстовой информации и работы с ней ручной сбор требует от сотрудников учебных заведений значительных ресурсов, автоматическое извлечение нужных данных становится все более востребованной задачей [3].

Инструменты обработки естественного языка позволяют автоматизировать процесс извлечения, исключая участие специалистов, работающих с текстовыми документами, к тому же можно проводить более глубокий анализ информации благодаря контексту. Большинство существующих методов в узкоспециализированных задачах, вроде извлечения из текстов навыков и компетенций, было проверено на англоязычных текстах, для анализа русскоязычного контента доступен лишь ограниченный набор инструментов и моделей.

В своем исследовании мы будем использовать термин «предметно-ориентированная сущность» (ПОС), который охватывает различные компоненты, связанные с конкретной областью знаний или исследований, отражая ее тематический или предметный контекст, и который включен в текст с учетом практического применения. Эта сущность может включать разнообразные элементы, такие как ключевые слова, фразы, именованные сущности, навыки и термины, которые способствуют более точной передаче информации и всестороннему описанию выбранных тем в корпусе документов.

Цель работы — проведение сравнительного анализа предварительно обученных моделей на основе трансформеров для задачи извлечения используемых предметно-ориентированных сущностей из студенческих отчетов. В частности, проведение дообучения и сравнение методов, основанных на энкодерах и предназначенных для распознавания именованных сущностей, и методов, основанных на энкодерах и декодерах или только декодерах, для генерации размеченных текстов на русском языке.

Для проведения исследования экспертами был размечен набор данных, полученный из 300 отчетных документов студентов, для сравнения качества и анализа ошибок выбранных моделей.

1. Обзор работ по тематике

В статье [4] дан небольшой обзор существующих методов извлечения терминов из текстов. Существуют несколько групп методов, основанных на правилах, на статистических признаках, на глубоком обучении.

Методы, основанные на правилах, используют регулярные выражения или лексико-грамматические шаблоны для выделения терминов. Правила составляются исследователями отдельно для специализированных корпусов [5]. Вторая группа методов использует статистические признаки для отбора терминов. К таковым методам относится, например, rutermextract, который вычисляет частотность использования отдельных слов или словосочетаний в текстах и на их основе определяет специфические для текстов слова, которые являются кандидатами в термины. Комбинированные методы, использующие для отбора терминов и шаблоны, и статистические показатели, показывают более высокий результат для извлечения терминов [6]. Методы глубокого обучения для извлечения терминов из русскоязычных текстов были применены в работе [7], где авторами предложена нейросетевая архитектура, использующая векторные представления текстов из мультиязычной модели BERT (mBERT) [8]. В работе [9]проводилось сравнение дообученных моделей, основанных на архитектуре Transformer, ruBERT [10] и mBERT.

Задача извлечения терминов является частным случаем задачи извлечения именованных сущностей [11, 12]. Для этой задачи зачастую используются языковые модели типа Transformer [13]. В последних работах сравниваются модели для извлечения именованных сущностей из русских текстов. В [14] лучший результат показала модель ruBERT (F-мера — 81.3 % на уровне сущностей). В работе [15] F-мера составила 68.82 % на уровне сущностей, данный результат был также получен моделью ruBERT, дополнительно обученной на текстах предметной области.

В последние годы также был опубликован ряд работ по извлечению упоминаний навыков из русскоязычных текстов вакансий. В статье [16] исследуется эффективность методов извлечения ключевых слов и модели BERT, дообученной для задачи распознавания именованных сущностей (named-entity recognition, NER). Авторами показано, что BERT значительно превосходит методы извлечения ключевых слов, однако демонстрирует достаточно низкое качество при извлечении навыков, отсутствующих в обучающей выборке. Работа [17] описывает подход на основе модели BERT и построения синтаксических деревьев. Автор подчеркивает, что в настоящее время перспективным инструментарием для решения рассматриваемой задачи являются предварительно обученные модели, основанные на архитектуре Transformer [18]. В работе [19] навыки извлекаются с помощью инструмента SkillNER. Поскольку данная модель предназначена для анализа англоязычных текстов, авторы предложили использовать инструменты машинного перевода русских текстов. Статья [20] исследует эффективность больших языковых моделей для извлечения навыков из русскоязычных текстов. Авторы сравнивают традиционные модели NER, основанные на энкодерах, с большими языковыми моделями, имеющими декодер-архитектуру. Результаты показали более высокую эффективность и более низкую вычислительную сложность моделей NER (F-мера — 81 %, доля правильно размеченных навыков — 73 %).

Другой близкой по тематике задачей является подбор ключевых слов. Большинство существующих исследований по подбору ключевых слов охватывает методы извлечения без учителя (в частности, статьи [21, 22]). В работах последних лет уделяется внимание генеративным методам подбора ключевых слов. В частности, в [23] показано, что мультиязычная энкодер-декодер модель способна генерировать ключевые слова в нормализованном виде и превосходить по метрикам (F-мера — 11.24 %, BERTScore — 76.89 %) традиционные подходы. В статье [24] сравниваются ключевые слова, подобранные с помощью ChatGPT, статистических подходов и подходов, основанных на машинном обучении. Результаты показывают низкую долю совпадений между полученными ключевыми словами. Однако наборы ключевых слов, составленные ChatGPT и экспертами, имеют относительно

высокую долю совпадений. Авторами работы [25] сравниваются модели генерации ключевых слов, основанные на энкодер-декодер и декодер-архитектуре. Лучший результат (F-мера — 20.16 %) был получен моделью, основанной на инструкциях, с помощью подхода с применением примеров.

2. Методы

2.1. Данные

Была собрана коллекция, включающая 300 отчетных документов по практическим подготовкам, проектным дисциплинам и выпускным квалификационным работам студентов, поступивших на ИТ-направление «Математическое обеспечение и администрирование информационных систем» в 2019–2022 годах. В каждом отчете экспертами были идентифицированы и аннотированы предметно-ориентированные сущности, относящиеся к предметной области ИТ-сферы, которые не просто упоминались, а использовались в процессе выполнения соответствующих работ. В текстах были представлены такие сущности, как названия языков программирования (Python, C++, C#), фреймворков и библиотек (Django, React, Next.js, Pandas, Sklearn), баз данных и технологий ORM (PostrgeSQL, MySQL, SQLAlchemy), инструментов для DevOps (Docker, Kubernetes, Nginx) и других групп инструментов разработки, названия алгоритмов (алгоритм k-ближайших соседей, алгоритм классификации, агломеративная кластеризация) и структур данных (граф, дерево, хэш-таблица). Для выделения начала и конца вхождения упоминания предметно-ориентированной сущности в тексте использовались теги [TAG*] и [*TAG] соответственно. Примеры размеченных текстов представлены в таблице 1.

Для последующего исследования из каждого студенческого отчета были извлечены абзацы — тексты (всего 2195), содержащие упоминания предметно-ориентированных сущностей. Всего выделено 1460 уникальных ПОС. Полученный датасет был разделен на обучающую и тестовую выборки в соотношении 70:30 (таблица 2). В тестовой выборке представлено 290 уникальных сущностей, что составляет 19 % от общего числа уникальных ПОС.

В большинстве случаев (72 %) в текстах присутствовало упоминание одной предметно-ориентированной сущности. Чаще всего упоминались сущности: «python» (4.9 % текстов), «база данных» (3.4 %), «javascript» (2.7 %) и «postgresql» (2.6 %). Остальные ПОС встречались меньше, чем в 2 % текстов. Подробные характеристики датасета визуализированы на рисунке 1, на котором отражены сущности, присутствующие не менее чем в 0.8 % текстов, и рисунке 2, демонстрирующем частоту появления в текстах определенного количества ПОС.

2.2. Модели

В данном исследовании были сравнены несколько подходов к извлечению предметно-ориентированных сущностей из студенческих отчетных документов. Поскольку анализ работ по тематике исследования показал эффективность моделей, основанных на архитектуре Transformer [18], для сравнения были выбраны предварительно обученные лингвистические модели, использующие эту архитектуру. В частности, были рассмотрены модели, основанные только на энкодере, на энкодере и декодере, только на декодере. Подробное описание используемых моделей представлено в таблице 3. Для обучения моделей использовались библиотеки SimpleTransformers[1] и Transformers [26].

- Модели, основанные только на энкодере:
  - ruBERT, русскоязычная версия модели BERT [8]. Предварительно обучена на русскоязычных текстах Википедии и новостей с помощью маскированного языкового моделирования (masked language modeling, MLM) с учетом контекста с обеих сторон, а также использует NSP (next sentence prediction), который определяет, следует ли одно предложение за другим, чтобы улучшить понимание связей между ними.

[1] https://simpletransformers.ai/

Table 1. Examles of texts in dataset **Таблица 1.** Примеры текстов в датасете

Описание ПОС	Размеченный текст
СУБД — понятие из области разработки баз данных PostgreSQL — название используемой СУБД	При выборе [TAG*]СУБД[*TAG] мы остановились на [TAG*]PostgreSQL[*TAG], так как каждый член нашей команды имел опыт работы с этой базой данных. Это облегчило командную работу и позволило использовать уже накопленные знания и навыки.
WPF — название системы, используемой в разработке	[TAG*]WPF[*TAG] — система для построения клиентских приложений Windows. С помощью данной системы был построен пользовательский интерфейс приложения.
FastHTTP — название фреймворка	[TAG*]FastHTTP[*TAG]: Высокопроизводительный HTTP сервер для Go. Использовался для обработки HTTP-запросов с минимальными задержками и высокой пропускной способностью [6].
HTML — название языка разметки CSS — название языка стилей JavaScript — название языка программирования	Клиентская часть веб-приложения, с которой осуществляет работу пользователь, использует [TAG*]HTML[*TAG] разметку для расположения элементов интерфейса, [TAG*]CSS[*TAG] для визуального представления страницы, [TAG*]JavaScript[*TAG] код для динамической работы со страницей и визуализацией. Именно с использованием данного языка происходит фильтрация таблицы данных с существующими элективными дисциплинами.
Алгоритм k-ближайших соседей — название используемого алгоритма k-NN — сокращенное название алгоритма	Полностью выполнить задачу классификации, опираясь лишь на правила, установленные заказчиком, не представляется возможным, из-за таких классов, как «шум», «тишина» и «музыка». Поэтому было принято решения использовать [TAG*]алгоритм k-ближайших соседей[*TAG] ([TAG*]k-NN[*TAG]), который был выбран в качестве первичной модели классификации аудиофайлов по нескольким причинам. Во-первых, это интуитивно понятный и простой в реализации метод. Во-вторых, он демонстрирует хорошие результаты на небольших объемах данных.
Стемминг и лемматизация — названия методов нормализации текстовых данных	Методы векторизации взяты из [4]. Из текста функций заранее удаляются знаки препинания, после этого он разбивается на токены (слова), если слово является стоп-словом (предлог, союз, местоимение и пр.) или содержит символы, не являющиеся буквами, то оно не используется. Разные формы одних и тех же слов приводятся к одной с помощью [TAG*]стемминга[*TAG] или [TAG*]лемматизации[*TAG]. Для стемминга используется стеммер, который отсекает от слов их части.

Table 2. Dataset statistics **Таблица 2.** Характеристики датасета

Характеристика	Обучающая выборка	Тестовая выборка
Количество текстов	1538	657
Средняя длина текстов (токенов)	28.04±19.54	27.32±19.59
Среднее количество предметно-ориентированных сущностей	1.51±1.07	1.48±0.98

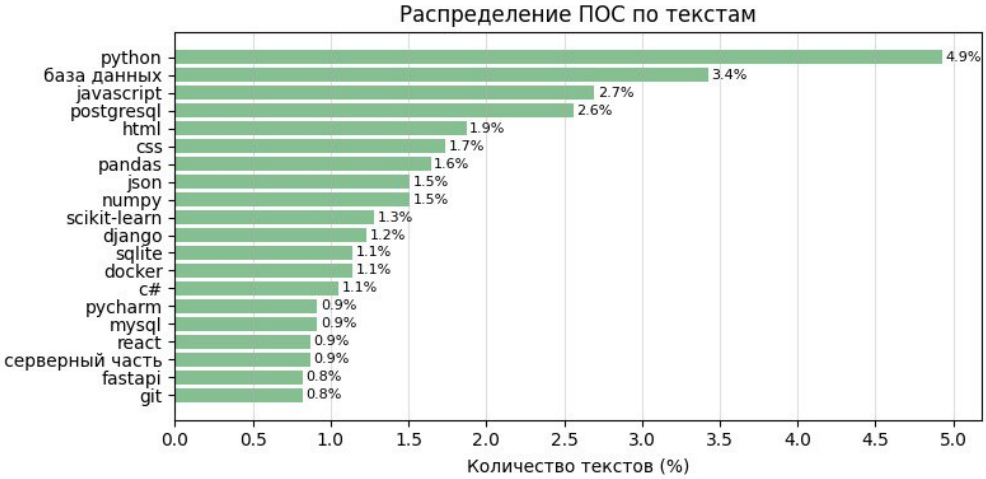

Fig. 1. Frequency of occurrence of object-oriented entities

Рис. 1. Частота встречаемости предметно-ориентированных сущностей

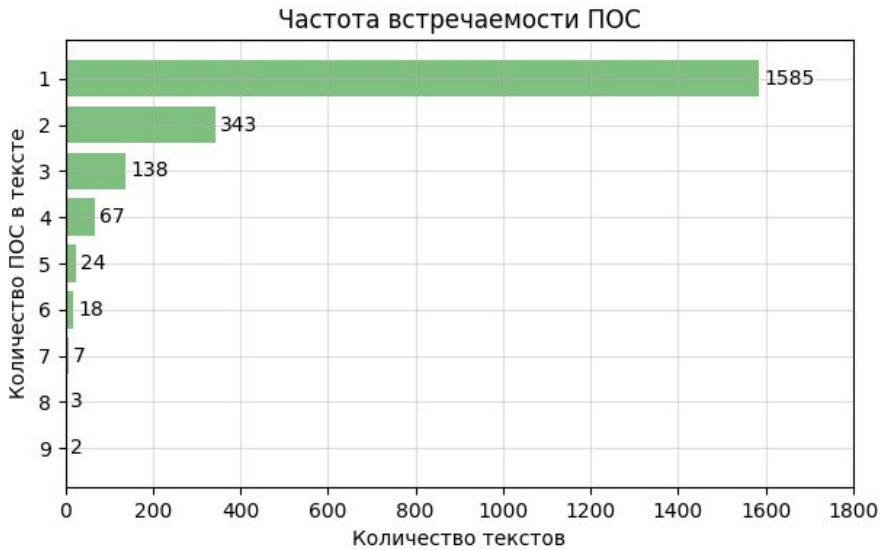

Fig. 2. Distribution of object-oriented entities across texts

Рис. 2. Распределение предметно-ориентированных сущностей по текстам

  - ruRoBERTa, улучшенная версия BERT с пересмотренным подходом к предобучению [27]. При обучении на русскоязычных текстах Википедии, новостей, книг, использовалось динамическое маскирование на каждой эпохе для больших батчей и более длинными последовательностями.
- Модели, основанные на энкодере и декодере:
  - ruT5, русскоязычная версия модели T5 [28] с архитектурой sequence-to-sequence. Предварительно обучена на русскоязычных текстах (книги, статьи, веб-страницы) для генерации текстов и использованием техники маскирования спангов (span corruption), где случайные фрагменты текста заменяются на маски, и модель учится их восстанавливать.
  - mBART, модель машинного перевода с архитектурой sequence-to-sequnce, построенная на базовой архитектуре BART [29]. Модель была предварительно обучена на более чем

Table 3. The models used in this study **Таблица 3.** Модели, используемые в работе

Модель	Версия	Архитектура	Количество параметров	Данные для обучения
ruBERT	DeepPavlov/rubert-base-cased [10]	Энкодер	178M	Википедия, новостные тексты
	ai-forever/ruBert-base [32]			Википедия, новостные тексты, Librusec, C4, OpenSubtitles
ruRoBERTa	ai-forever/ruRoberta-large [32]		355M	
ruT5	ai-forever/ruT5-base [32]	Энкодер-декодер	222M	
mBART	facebook/mbart-large-50 [33]		680M	Common Crawl (CC25), XLMR
ruGPT	ai-forever/rugpt3large_based_on_gpt2 [32]	Декодер	760M	Википедия, новостные тексты, Librusec, C4, OpenSubtitles
T-lite	t-bank-ai/T-lite-instruct-0.1		8B	Открытые англоязычные датасеты, переводы англоязычных датасетов, синтетический контекст для вопросно-ответных систем

50 языках с использованием комбинации техник маскирования фрагментов (span masking) и перемешивания предложений (sentence shuffling).

- Модели, основанные только на декодере:
  - ruGPT, русскоязычная модель, основанная на архитектуре GPT-3 [30] и использующая кодовую базу GPT-2 из библиотеки Transformers [26, 31].
  - T-lite, большая языковая модель, основанная на инструкциях, для обучения которой использовался набор данных, на 85 % состоящий из текстов на русском языке.

Модели, основанные на энкодере, обучались на задаче разметки последовательности токенов по аналогии с задачей распознавания именованных сущностей. ruBERT и ruRoBERTa были дообучены (fine-tuned) на обучающей выборке. При этом DeepPavlov/rubert-base-cased, ai-forever/ruBert-base, ruRoBERTa обучались в течение четырех, шести и пяти эпох соответственно. Количество эпох определялось экспериментальным путем: начиная с одной эпохи и постоянно увеличивая на одну эпоху, пока не будет достигнуто лучшее качество модели. Использовались следующие параметры для обучения: максимальная длина последовательности — 128 токенов, скорость обучения — 4e-5, размер батча — 16, оптимизатор — AdamW [34]. На вход каждой модели подавался текст, для каждого токена которого было необходимо предсказать одну из меток (B — токен является первым словом в ПОС, I — токен является не первым словом в ПОС, O — токен не является частью ПОС). Теги [TAG*] и [*TAG] были удалены перед процессом дообучения.

Поскольку модели ruT5 и mBART относятся к типу sequence-to-sequence, они обучались на задаче генерации последовательности токенов, а именно требовалось сгенерировать исходными текст с верно расставленными тегами [TAG*] и [*TAG]. Дообучение модели ruT5 проводилось в течение 50 эпох со следующими параметрами: 128 токенов, скорость обучения — 4e-5, размер батча — 16, оптимизатор — AdamW. Дообучение mBART на обучающей выборке проводилось в течение 20 эпох с использованием следующих параметров: максимальная длина последовательности — 256 токенов, скорость обучения — 4e-5, размер батча — 8, оптимизатор — AdamW. На вход моделей, основанных на архитектуре энкодер-декодер, подавался текст без тегов [TAG*] и [*TAG], сгенерированный текст

должен был представлять собой исходный текст с выделенными тегами [TAG*] и [*TAG], обозначающими вхождения предметно-ориентированных сущностей.

Особенностью процесса обучения моделей с декодер-архитектурой является то, что для заданного слова слои внимания могут получать доступ только к словам, расположенным перед ним в тексте [31]. Таким образом, обучение декодера сосредоточено на предсказании следующего слова в тексте. Языковая модель ruGPT была дообучена с использованием задачи причинного моделирования языка (causal language modeling, CLM) с максимальной длиной последовательности 1024 токена в течение десяти эпох. Количество эпох было установлено по аналогии с экспериментом по дообучению данной модели для одной из задач генерации текста на естественном языке, представленном в оригинальной статье [32]. В качестве остальных параметров дообучения взяты параметры, использующиеся в библиотеке SimpleTransformers по умолчанию. Входной текст подавался в следующем формате: *текст без тегов [TAG*] и [*TAG] + </s> + текст с выделенными тегами [TAG*] и [*TAG] + </end>*. Текст тестовой выборки подавался в виде: *текст без тегов [TAG*] и [*TAG] + </s>*.

Модель T-lite, которая относится к типу моделей, основанных на инструкциях, была обучена в технике обучения при помощи запросов (prompt-based learning) и подхода с применением примеров (few-shot learning). В качестве примеров были взяты десять случайных текстов из обучающей выборки. Для генерации текстов с выделенными тегами были установлены следующие параметры: температура, равная 0.1, и максимальная длина сгенерированного текста, равная 256 токенам. Выбор низкого значения параметра температуры обусловлен необходимостью модели повторять исходный текст с добавлением тегов и избегать перефразирования [35].

2.3. Метрики

Оценка результатов моделей проводилась с использованием двух метрик: посимвольной F-меры и F-меры на уровне сущностей. Использование метрик на уровне сущностей широко распространено в задаче извлечения именованных сущностей (в частности, в упомянутых выше работах [14, 15]). Авторы обзора [36] указывают, что помимо метрик, оценивающих точное совпадение сущностей, для оценки качества моделей также используют метрики, характеризующие частичное совпадение. Кроме того, посимвольные метрики широко используются в задачах, связанных с выделением фрагментов в тексте на естественном языке (например, в статье [37]).

Для оценки результатов каждый текст из тестовой выборки был разбит на токены (в зависимости от выбранной метрики — по словам или по символам). Также были разбиты соответствующие тексты, размеченные моделью. Для каждого текста в тестовом наборе рассчитывалось количество токенов, которые были верно предсказаны моделью. Это число использовалось для вычисления точности и полноты.

Точность (precision) и полнота (recall) вычислялись следующим образом:

$$Precision = \frac{TruePredicted}{PredictedTokens},$$

где *TruePredicted* — количество верно предсказанных токенов, *PredictedTokens* — общее количество токенов в тексте, размеченном моделью.

$$Recall = \frac{TruePredicted}{TestTokens},$$

где *TruePredicted* — количество верно предсказанных токенов, *TestTokens* — общее количество токенов в тексте из тестовой выборки.

Точность и полнота вычислялись для каждой пары текстов (текста из тестового набора и соответствующего текста, размеченного моделью). Затем средние значения точности и полноты использовались для вычисления F-меры:

Table 4. Results, %

Таблица 4. Результаты, %

Модель	Метрики на уровне сущностей			Посимвольные метрики		
	F1	Precision	Recall	F1	Precision	Recall
Модели, основанные только на энкодере						
ruBERT (DeepPavlov/rubert-base-cased)	72.48	73.43	71.56	67.14	70.16	64.36
ruBERT (ai-forever/ruBert-base)	73.43	73.39	73.47	68.42	69.15	67.70
ruRoBERTa	73.56	73.60	73.52	70.36	68.28	72.57
Модели, основанные на энкодере и декодере						
ruT5	84.58	86.48	82.76	57.21	57.21	56.34
mBART	93.55	93.53	93.56	70.88	70.84	70.92
Модели, основанные только на декодере						
ruGPT	87.32	87.31	87.34	50.35	50.70	50.00
T-lite	86.96	86.98	86.95	51.26	51.08	51.44

$$F1 = \frac{2 \cdot AvgPrecision \cdot AvgRecall}{AvgPrecision + AvgRecall},$$

где *AvgPrecision* — среднее значение точности по каждой паре текстов, *AvgRecall* — среднее значение полноты по каждой паре текстов.

Использование двух способов расчета F-меры позволило оценить результаты как с точки зрения полного совпадения вручную размеченных предметно-ориентированных сущностей и сущностей, выделенных моделью, так и с точки зрения их частичного совпадения на уровне посимвольного пересечения.

3. Результаты

Результаты сравнения моделей представлены в таблице 4. Полученные значения метрик показывают, что модель mBART в большинстве случаев демонстрирует лучшее качество извлечения предметно-ориентированных сущностей из студенческих отчетных документов. Наибольшее преимущество mBART демонстрирует в значениях метрик, рассчитанных на уровне сущностей. Это показывает превосходство mBART с точки зрения полного совпадения вручную размеченных и выделенных моделью предметно-ориентированных сущностей. Основываясь на результатах посимвольных метрик, можно сделать вывод о близком качестве моделей mBART, ruRoBERTa и ruBERT.

Модель mBART показывает лучшее значение посимвольной F-меры (70.88 %) и Precision (70.84 %), при этом второе по величине значение F-меры демонстрирует ruRoBERTa (70.36 %), значение Precision — DeepPavlov/rubert-base-cased (70.16 %). Лучшее значение посимвольной метрики Recall показывает ruRoBERTa (72.57 %), превосходя mBART (70.92 %) на 1.65 процентных пункта.

Был проведен анализ ошибок, которые совершали модели. Результаты представлены в таблице 5. Для проведения анализа было осуществлено сопоставление сущностей каждого текста тестового набора с сущностями, идентифицированными моделями в указанных текстах. Неправильными сущностями текста являются те, которые выделила модель, но в изначальной разметке их не было. Оценивалось также, насколько много было пропущено ПОС из размеченных текстов, а для генеративных моделей способность размечать текста в соответствии с форматом.

Среди моделей, предназначенных для генерации текста с разметкой в формате [TAG*] и [*TAG], наименьшее количество ошибок при идентификации предметно-ориентированных сущностей демонстрирует модель mBART, которая также характеризуется меньшим количеством ошибок при расстановке тегов. Для других моделей характерно появление случаев, когда в результирующий текст встраивается значительное число открывающих тегов [TAG*], не имеющих соответствующих закрывающих пар (до 10–20 несбалансированных тегов в некоторых примерах).

Table 5. Errors of the models **Таблица 5.** Ошибки моделей

Модель	Среднее количество неправильно выделенных сущностей в тексте	Среднее количество сущностей, которые модель не смогла выделить в тексте	Среднее количество непарных меток [TAG*] и [*TAG]
DeepPavlov/rubert-base-cased	0.45±0.69	0.42±0.68	—
ai-forever/ruBert-base	0.47±0.73	0.44±0.69	—
ruRoBERTa	0.47±0.69	0.45±0.68	—
ruT5	0.64±0.82	0.75±0.96	0.10±0.31
mBART	0.53±0.85	0.58±0.85	0.11±0.63
ruGPT	0.56±0.93	0.86±0.93	0.22±1.32
T-lite	0.98±1.27	0.84±0.86	0.19±1.04

Модель T-lite выделяет множество элементов текста, которые не соответствуют предметно-ориентированным сущностям, включая числовые значения, подписи к иллюстрациям и таблицам, а также общие лексические единицы вроде «студент», «сотрудник», «сделка». Модель ruGPT, в свою очередь, идентифицирует английские слова и выражения, являющиеся наименованиями объектов или функций в библиотеках программного обеспечения. Наблюдаются ситуации, когда происходит выделение слов или выражений, относящихся к сфере информационных технологий, однако они оказываются усеченными или с заменой отдельных букв (например, «ransformers» — в тексте описывается использование библиотеки transformers, но модель выделила ее как «t[TAG*]ransformers[*TAG]»). Модель ruT5 зачастую искажает выделенные слова или фразы (например, преобразуя «Jupiter Notebook» в «Jeeppiter Notebook», а «postgresql» в «pargresql») и усекает отдельные предметно-ориентированные сущности, подобно модели ruGPT. Модель mBART в основном допускает ошибки при распознавании наименований алгоритмов и методов, ограничиваясь идентификацией имен их разработчиков.

Модели, предназначенные для извлечения именованных сущностей, демонстрируют меньшую частоту ошибок при идентификации предметно-ориентированных сущностей благодаря сохранению всех слов оригинального текста в неизмененном виде. Тем не менее, они склонны выделять лишь фрагменты сущностей (например, «база» вместо «база данных», «бинарная» вместо «бинарная классификация»), а лучше всего справляются с сущностями, состоящими из одного слова.

Часть ПОС (290), идентифицированных в текстах тестового набора, отсутствовали в обучающей выборке. Для оценки способности моделей к обобщению был проведен анализ точности извлечения ранее неизвестных сущностей. Худшие результаты продемонстрировали модели генерации текста T-lite, ruGPT и ruT5, корректно распознавшие лишь 34–38 % новых сущностей. Модель mBART показала более высокую точность — 52 %. Модели ruBERT и ruRoBERTa немного превзошли предыдущие модели, выявив правильный процент новых ПОС в диапазоне от 56 до 60 %.

Заключение

В ходе проведенного исследования была выполнена оценка качества моделей для генерации текстов, основанных на энкодере и декодере (mBART, T5) и только на декодере (T-lite, ruGPT), и моделей для извлечения именованных сущностей, основанных только на энкодере (ruBERT, ruRoBERTa), в контексте идентификации используемых предметно-ориентированных сущностей в текстах отчетных документов студентов, обучающихся на ИТ-направлении. Модели были дополнительно обучены на корпусе текстов, размеченном экспертами, что обеспечило их адаптацию к специфическим характеристикам данной предметной области.

Максимальные показатели по метрике F-меры были достигнуты генеративной моделью mBART (93.55 %), что свидетельствует о высокой точности этой модели в идентификации ПОС в тексте. Тем не менее, существенным недостатком генеративного подхода является наличие артефактов в виде орфографических ошибок, неправильных вставок меток и фрагментации слов. В данном отношении модели для извлечения именованных сущностей проявили себя как более надежные решения, поскольку они обеспечивают сохранение целостности исходных данных, сводя к минимуму вероятность возникновения ошибок форматирования. Вместе с тем, NER-модели часто извлекают лишь частичные фрагменты сущностей.

Полученные результаты свидетельствуют о применимости обеих групп моделей для обработки отчетных документов. Выбор между ними должен осуществляться исходя из конкретных требований. Если первоочередное значение имеет высокая точность сохранения оригинального текста и способность к идентификации новых, ранее не встречавшихся ПОС, предпочтительнее использовать модели для извлечения именованных сущностей. Если же основным критерием выступает максимальная полнота выявления сущностей, несмотря на возможные искажения, целесообразнее выбрать модель mBART.

С целью улучшения качества выделения ПОС в текстах работ студентов необходимо дообучать модели дополнительными типами отчетных документов (отзывы, характеристики, дневники практики, технологические карты студенческих проектов), что позволит расширить возможности моделей и улучшить точность извлечения ключевых фрагментов текста.

Результаты извлечения ПОС можно применить для создания цифрового портрета студента, проверки студенческих отчетов, оценки текстовых характеристик, сравнительного анализа документов, анализа учебных материалов и разработки ИТ-онтологий, используемых в образовательном процессе.

References

[1] Q. Guohao, W. Bin, W. Bai, and Z. Baoli, “Competency analysis in human resources using text classification based on deep neural network”, in *Proceedings of the IEEE Fourth International Conference on Data Science in Cyberspace*, 2019, pp. 322–329. DOI: 10.1109/DSC.2019.00056.

[2] I. Zakharova, Y. V. Boganyuk, M. Vorobyova, and E. Pavlova, “Diagnostics of professional competence of IT students based on digital footprint data”, *Informatics and Education*, vol. 4, no. 313, pp. 4–11, 2020. DOI: 10.32517/0234-0453-2020-35-4-4-11.

[3] Z. Alami Merrouni, B. Frikh, and B. Ouhbi, “Automatic keyphrase extraction: A survey and trends”, *Journal of Intelligent Information Systems*, vol. 54, no. 2, pp. 391–424, 2020. DOI: 10.1007/s10844-019-00558-9.

[4] E. Bruches, A. Pauls, T. Batura, V. Isachenko, and D. Shcherbatov, “Semantic analysis of scientific texts: Experience in creating a corpus and building language models”, *Software & Systems*, vol. 34, no. 1, pp. 132–144, 2021. DOI: 10.15827/0236-235X.133.132-144.

[5] Y. I. Butenko, N. S. Nikolaeva, and T. D. Margaryan, “Structural models of terminological word combinations for marking up a corpus of scientific and technical texts”, *NSU Vestnik. Series: Linguistics and Intercultural Communication*, vol. 19, no. 3, pp. 45–56, 2021. DOI: 10.25205/1818-7935-2021-19-3-45-56.

[6] A. Novikova, “Comparison of Sketch Engine and TermoStat tools for terminology extraction”, *International Journal of Open Information Technologies*, vol. 8, no. 11, pp. 73–79, 2020.

[7] E. Bruches and T. Batura, “Method for automatic term extraction from scientific articles based on weak supervision”, *Vestnik NSU. Series: Information Technologies*, vol. 19, no. 2, pp. 5–16, 2021, in Russian. DOI: 10.25205/1818-7900-2021-19-2-5-16.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[9] Y. Dementyeva, E. Bruches, and T. Batura, "Terms extraction from texts of scientific papers", *Software & Systems*, vol. 35, no. 4, pp. 689–697, 2022. DOI: 10.15827/0236-235X.140.689-697.

[10] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language", in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2019, pp. 333–339.

[11] V. K. Pimeshkov, M. L. Nikonorova, and M. G. Shishaev, "A combined term extraction method for the problem of monitoring thematic discussions in social media", *Informatics and Automation*, vol. 23, no. 4, pp. 1110–1138, 2024. DOI: 10.15622/ia.23.4.7.

[12] M. D. Averina and O. A. Levanova, "Extracting named entities from Russian-language documents with different expressiveness of structure", *Modeling and Analysis of Information Systems*, vol. 30, no. 4, pp. 382–393, 2023, in Russian. DOI: 10.18255/1818-1015-2023-4-382-393.

[13] X. Liu, J. A. Erkoyuncu, J. Y. H. Fuh, W. F. Lu, and B. Li, "Knowledge extraction for additive manufacturing process via named entity recognition with LLMs", *Robotics and Computer-Integrated Manufacturing*, vol. 93, p. 102 900, 2025. DOI: 10.1016/j.rcim.2024.102900.

[14] T. Atnashev *et al.*, "Razmecheno: Named entity recognition from digital archive of diaries "Prozhito"", in *Proceedings of the Fifth International Conference on Computational Linguistics in Bulgaria (CLIB 2022)*, 2022, pp. 22–38.

[15] M. Tikhomirov, N. Loukachevitch, A. Sirotina, and B. Dobrov, "Using BERT and augmentation in named entity recognition for cybersecurity domain", in *Proceedings of the 25th International Conference on Applications of Natural Language to Information Systems*, 2020, pp. 16–24. DOI: 10.1007/978-3-030-51310-8_2.

[16] P. V. Korytov, Y. Y. Gribetskiy, E. A. Andreeva, and I. I. Kholod, "Analysis of approaches for identifying key skills in vacancies", in *Proceedings of the International Conference on Soft Computing and Measurement*, 2024, pp. 300–303. DOI: 10.1109/SCM62608.2024.10554269.

[17] I. E. Nikolaev, "Knowledge and skills extraction from the job requirements texts", *Ontology of Designing*, vol. 13, no. 2, pp. 282–293, 2023, in Russian. DOI: 10.18287/2223-9537-2023-13-2-282-293.

[18] A. Vaswani *et al.*, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, pp. 1–11, 2017.

[19] A. V. Melnikova, M. S. Vorobeva, E. V. Egorova, and E. D. Chekanova, "Development of an algorithm for the formation of IT project teams based on data from the digital footprint of students", *Proceedings of the Institute for System Programming of the RAS*, vol. 36, no. 3, pp. 213–224, 2024. DOI: 10.15514/ispras-2024-36(3)-15.

[20] N. Matkin *et al.*, *Comparative analysis of encoder-based NER and large language models for skill extraction from Russian job vacancies*, 2024. arXiv: 2407.19816 [cs.CL].

[21] M. Khokhlova and M. Koryshev, "Keyness analysis and its representation in Russian academic papers on computational linguistics: Evaluation of algorithms", in *RASLAN*, 2022, pp. 25–33.

[22] O. Mitrofanova and D. Gavrilic, "Experiments on automatic keyphrase extraction in stylistically heterogeneous corpus of Russian texts", *Terra Linguistica*, vol. 13, no. 4, pp. 22–40, 2022. DOI: 10.18721/JHSS.13402.

[23] A. V. Glazkova, D. A. Morozov, M. S. Vorobeva, and A. A. Stupnikov, "Keyword generation for Russian-language scientific texts using the mT5 model", *Automatic Control and Computer Sciences*, vol. 58, no. 7, pp. 995–1002, 2024. DOI: 10.3103/S014641162470041X.

[24] D. Guseva and O. Mitrofanova, "Keyphrases in Russian-language popular science texts: comparison of oral and written speech perception with the results of automatic analysis", *Terra Linguistica*, vol. 15, no. 1, pp. 20–35, 2024. DOI: 10.18721/JHSS.15102.

[25] A. Glazkova, D. Morozov, and T. Garipov, *Key algorithms for keyphrase generation: Instruction-based LLMs for Russian scientific keyphrases*, 2024. arXiv: 2410.18040 [cs.CL].

[26] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing", Association for Computational Linguistics, 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[27] Y. Liu *et al.*, *RoBERTa: A robustly optimized BERT pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].

[28] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer", *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020. DOI: 10.5555/3455716.3455856.

[29] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703.

[30] T. Brown *et al.*, "Language models are few-shot learners", *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020. DOI: 10.5555/3495724.3495883.

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners", *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[32] D. Zmitrovich *et al.*, "A family of pretrained transformer language models for Russian", in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, 2024, pp. 507–524.

[33] Y. Tang *et al.*, "Multilingual translation from denoising pre-training", in *Findings of the Association for Computational Linguistics*, 2021, pp. 3450–3466. DOI: 10.18653/v1/2021.findings-acl.304.

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization", in *International Conference on Learning Representations*, 2019, p. 53 592 270.

[35] A. Kartelj, M. Mladenović, and S. Vujičić Stanković, "Comparison of algorithms for the recognition of ChatGPT paraphrased texts", *Journal of Big Data*, vol. 12, no. 1, pp. 1–17, 2025. DOI: 10.1186/s40537-025-01082-0.

[36] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition", *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2022. DOI: 10.1109/TKDE.2020.2981314.

[37] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, and P. Nakov, "Fine-grained analysis of propaganda in news articles", in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 5636–5646. DOI: 10.18653/v1/D19-1565.