



Образовательные ресурсы и лучшая практика ИТ-образования

<https://doi.org/10.25559/SITITO.021.202503.585-592>

УДК 004.912

Цифровой сервис для подготовки аттестационных листов обучающихся

А. В. Зыкина, В. В. Мунько*, Е. О. Шикин

Оригинальная статья

ФГАОУ ВО «Омский государственный технический университет», г. Омск, Российская Федерация

Адрес: 644050, Российская Федерация, г. Омск, пр. Мира, д. 11

* vvmunko@omgtu.ru

Аннотация

В статье представлен конвейер автоматизированной обработки учебной документации. Для создания числовой модели текстовой информации используется векторное представление данных, для повышения качества извлечения и предобработки данных из документов используются технологии OCR и регулярные выражения. Особое внимание уделено исследованию возможностей Chroma DB, которая предоставляет эффективные средства для хранения и поиска информации по векторным данным. На основе семантического сопоставления справок об обучении и детального сравнения по дисциплинам учебного плана реализован механизм рекомендации наиболее подходящего направления подготовки. Реализация представлена как автономный микросервис, развёрнутый в Docker и доступный через FastAPI. В результате создано гибкое решение, которое значительно сокращает время ручной сверки учебных планов и справок об обучении, обеспечивает высокую точность обнаружения сходства в наименованиях дисциплин и объёмах часов, а также легко масштабируется и интегрируется в существующие корпоративные информационные системы, благодаря стандартизированному REST-интерфейсу. Использование разработанного цифрового сервиса позволяет, с одной стороны, повысить объективность анализа учебных планов, сравнения дисциплин и выделения ключевых компетенций, что обеспечивает максимально корректное распределение обучающихся по направлениям подготовки. С другой стороны, цифровой сервис значительно ускоряет процесс подготовки документов и уменьшает нагрузку на учебные отделы образовательных организаций.

Ключевые слова: OCR, sentence-transformers, векторизация, ChromaDB, регулярные выражения, FastAPI, Docker, справка об обучении, учебный план

Финансирование: Работа выполнена в рамках Государственного задания № FSGF-2024-0006 «Разработка новых моделей и методов принятия решений в многоуровневых системах управления».

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Зыкина А. В., Мунько В. В., Шикин Е. О. Цифровой сервис для подготовки аттестационных листов обучающихся // Современные информационные технологии и ИТ-образование. 2025. Т. 21, № 3. С. 585-592. <https://doi.org/10.25559/SITITO.021.202503.585-592>

© Зыкина А. В., Мунько В. В., Шикин Е. О., 2025



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Educational Resources and Best Practices of IT-Education

Digital Service for Comparing Documents to Prepare Attestation Sheets of Students

A. V. Zykina, V. V. Munko^{*}, E. O. Shikin

Original article

Omsk State Technical University, Omsk, Russian Federation

Address: 11 Mira Ave, Omsk 644050, Russian Federation

^{*} vvmunko@omgtu.ru

Abstract

The article presents a pipeline for automated processing of educational documentation. A numerical model of textual information is created using vector data representation, while technologies such as OCR and regular expressions are employed to enhance the quality of extracting and preprocessing data from documents. Special attention is given to exploring the capabilities of Chroma DB, which provides efficient tools for storing and searching through vectorized data. Based on semantic matching between learning certificates and curricula, a recommendation mechanism has been implemented that suggests the most suitable training direction and performs detailed comparison across curriculum subjects. The implementation is presented as an autonomous microservice deployed in Docker and accessible via FastAPI. As a result, a flexible solution has been developed that significantly reduces time spent manually verifying study plans and learning certificates, ensures high accuracy in detecting similarities in subject names and hours volume, and can be easily scaled and integrated into existing corporate information systems due to its standardized REST interface. Utilizing this digital service allows, on one hand, increasing objectivity in analyzing study plans, comparing disciplines, and identifying key competencies, ensuring the most accurate allocation of learners to specific areas of study. On the other hand, it greatly accelerates document preparation processes and reduces workload on academic departments within educational institutions.

Keywords: OCR, sentence-transformers, vectorization, ChromaDB, regular expressions, FastAPI, Docker, academic transcript, curriculum

Funding: This work was carried out under State Assignment No. FSGF-2024-0006 "Development of new models and decision-making methods in multi-level control systems".

Conflict of interests: The authors declares no conflict of interest.

For citation: Zykina A.V., Munko V.V., Shikin E.O. Digital Service for Comparing Documents to Prepare Attestation Sheets of Students. *Modern Information Technologies and IT-Education*. 2025;21(3):585-592. <https://doi.org/10.25559/SITITO.021.202503.585-592>



1 Введение

В современном мире, характеризующемся стремительным развитием цифровых технологий, автоматизация обработки документов приобретает всё большую актуальность. Особенно в образовательной сфере, где накапливается огромное количество информации в виде учебных планов, справок об обучении и другой документации. Ручная проверка документации требует многократной проверки данных, что увеличивает время обработки и создает риск ошибок, связанных с человеческим фактором. Именно поэтому разработка алгоритмов, способных автоматически извлекать и сопоставлять ключевую информацию, становится важным направлением в практике использования информационных технологий. В работе предложен инструмент, который способен извлекать текстовую информацию из PDF-документов, осуществлять её предварительную обработку, а также очищать от лишних символов и структурировать для последующего анализа. Реализованный алгоритм интегрирует возможности OCR-технологий, обработки естественного языка и методов машинного обучения, что обеспечивает высокую точность распознавания текста и корректное сопоставление данных из разных источников [1, 2]. Использование современных библиотек, таких как *Langchain*, *HuggingFace* и *SentenceTransformer*, помогло разработке качественного решения. Для обеспечения гибкости и масштабируемости использован фреймворк *FastAPI* [3-5]. Это упрощает интеграцию решения в уже существующие информационные системы образовательных организаций.

2 Алгоритмическая модель

Обозначим множество учебных планов $P = \{p_1, \dots, p_k, \dots, p_m\}$. Каждый план p_k представлен набором дисциплин $D_k = \{d_{k1}, \dots, d_{kj}, \dots\}$, где у каждой дисциплины d_{kj} задан объём часов. Справка об обучении S содержит набор записей $R = \{r_1, \dots, r_i, \dots, r_n\}$, где каждая запись r_i – строка текста с наименованием дисциплины и числовым значением объема часов по этой дисциплине.

Необходимо найти учебный план (индекс k^*) такой, что семантическая близость между множеством дисциплин R и D_{k^*} максимальна. Формально, для каждого плана p_k для каждой пары (r_i, d_{kj}) вычисляется косинусное сходство эмбедингов [6, 7]:

$$\text{sim}(r_i, d_{kj}) = \frac{(v_r \cdot v_d)}{\|v_r\| \cdot \|v_d\|},$$

где v_r – это эмбединг записи r_i (строки из справки об обучении), v_d – эмбединг дисциплины d_{kj} из учебного плана, и для каждого плана p_k определяется агрегированная метрика $\text{score}(p_k, S)$ (например, сумма/среднее лучших соответствий с учётом веса по часам). Затем выбирается индекс k^*

$$k^* = \arg \max_k \text{score}(p_k, S).$$

Для экспериментов учебные планы взяты с официального сайта вуза (ОмГТУ). Использованы актуальные версии планов, доступные на момент проведения исследования. Эти планы представляют собой официальный эталонный корпус: они охватывают основные факультеты и направления подготовки, содержат структурированные перечни дисциплин и указания объёмов учебных часов, что делает их удобными для формирования базы для сопоставления данных. Справки об обучении студентов предоставлены деканатами соответствующих факультетов и использованы в качестве входных запросов для тестирования алгоритма. В выборке присутствовали как PDF-файлы, так и отсканированные документы различного качества. Это позволяет оценить устойчивость алгоритма к реальным условиям использования и к вариативности форматов документов.

3 Материалы и методы

Получение документов в формате PDF начальный этап работы с документацией. Студенты предоставляют справки об обучении в PDF файлах, который признан стандартом для обмена информацией обучающихся, так как корректно отображается на устройствах с различными операционными системами и экранами. Работа со справками об обучении связана с извлечением текста из отсканированных изображений. Это возможно благодаря технологии OCR (*Optical Character Recognition*), которая преобразует графическое представление текста в цифровой формат. OCR особенно полезен при работе с неструктурированными документами или сканами, где текст не является машинно-читаемым. Для реализации OCR используется библиотека *Tesseract*, разработанная компанией *Hewlett-Packard* и поддерживаемая *Google*. Это мощный инструмент с открытым исходным кодом, предоставляющий возможности распознавания текста на более чем 100 языках. Процесс обработки текста методом оптического распознавания текста разделяется на следующие этапы [8]: Удаление шума, повышение кратности, бинаризация (преобразование изображения в черно-белый формат) и выравнивание. Разделение изображения на текстовые блоки, строки и символы. Это позволяет модели эффективнее обрабатывать информацию. Преобразование графических элементов в текстовый формат. После распознавания текст часто содержит «шум» – лишние символы, пробелы, переносы строк и другие элементы, которые мешают дальнейшему анализу. Для очистки текста применяются регулярные выражения. Это мощный инструмент для работы с текстом. Они позволяют находить, заменять и удалять текстовые элементы на основе заданных шаблонов. Например, регулярные выражения можно использовать для удаления лишних пробелов или для извлечения



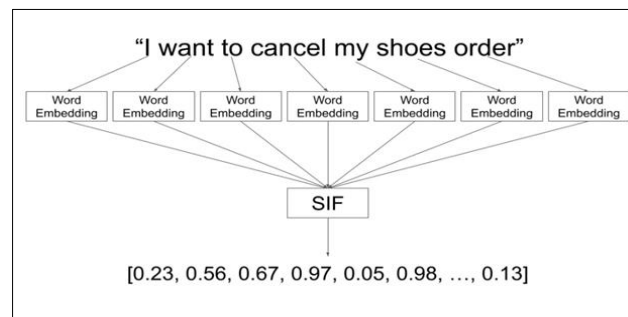
определённых структурированных данных, таких как даты или номера телефонов.

После очищения текста от лишнего мусора наступает этап векторизации. Векторизация слов и текстов представляет собой один из самых значимых методов преобразования текстовой информации в числовую форму. Векторизация преобразует слова, фразы, предложения и целые документы в математические объекты – векторы, что делает их пригодными для анализа, сравнения и обработки.

Идея векторизации основывается на гипотезе дистрибутивной семантики: «значение слова определяется контекстом, в котором оно используется» [9-12]. Это означает, что слова, которые часто встречаются в схожих контекстах, имеют схожие значения. Векторизация превращает эту идею в числовую форму: каждое слово представляется как точка в многомерном пространстве, где схожие слова находятся ближе друг к другу. Для решения таких задач существуют специальные модели *Sentence-BERT* и *Universal Sentence Encoder*. Они преобразуют тексты в компактные векторные представления (эмбединги), сохраняя их семантическую и синтаксическую структуру. Эти алгоритмы могут «понимать» смысл предложений, сравнивать их между собой и находить скрытые связи, в отличие от классических методов, таких как *Bag of Words (BoW)* или *TF-IDF*, которые представляют текст через статистику частот слов (игнорируя порядок и смысл). *Sentence Transformers* работают с контекстом и семантикой. Например, *BoW* превратит фразы «Кот бежит за собакой» и «Собака бежит за котом» в идентичные вектора, тогда как *Sentence Transformers* учтёт разный смысл, связанный с порядком слов. Даже продвинутые модели глубокого обучения, такие как *Word2Vec* или *GloVe*, фокусируются на эмбедингах отдельных слов, а не целых предложений. Они «не видят» разницы между «Я люблю банки» (финансы) и «Я открываю банки» (сосуды), если контекст не задан явно. *Sentence Transformers* решают эту проблему, анализируя предложения целиком через механизмы внимания трансформеров. Это позволяет им улавливать некоторые нюансы, такие как многозначность или скрытые связи между фразами. На рисунке 1 представлено создание векторного представления предложения с использованием *sentence embeddings*.

Базой *sentence-transformers* является стандартный трансформер, который сначала преобразует входной текст в последовательность токенов. Каждый токен затем проходит через несколько слоев само-внимания (*self-attention*) и преобразования, что позволяет модели учитывать контекстуальные взаимосвязи внутри предложения. Однако в отличие от обычного трансформера, который возвращает эмбединги для каждого токена, *sentence-transformers* добавляют дополнительный этап – агрегирование (*pooling*) токеновых представлений в один вектор фиксированной размерности. Для этого существует несколько методов [13-18]:

Mean Pooling: Вычисление среднего арифметического



Р и с. 1. Векторное представление предложения
F i g. 1. Vector representation of a sentence

Источник: здесь и далее в статье все рисунки составлены авторами.
Source: Hereinafter in this article all figures were drawn up by the authors.

эмбедингов всех токенов. Этот метод является простым, но часто эффективным, поскольку равномерно учитывает вклад всех слов.

Max Pooling: Выбор максимальных значений по каждому измерению, что может помочь выделить наиболее значимые признаки.

CLS-nooling: Использование эмбединга специального [CLS]-токена, который изначально предназначался для задач классификации. В некоторых случаях он оказывается достаточно информативным для представления всего предложения.

Работа с базами данных для векторных представлений является основным элементом современных задач в области обработки текстов, изображений и других типов данных [19]. Эти задачи включают поиск сходства, кластеризацию и семантический анализ, которые становятся возможными благодаря представлению данных в виде векторов [20-22]. *ChromaDB* является мощным и универсальным инструментом для работы с такими векторными данными. База данных оптимизирована для хранения, обработки и поиска векторов, что делает ее незаменимым в самых разных приложениях. Её основная задача заключается в эффективном сохранении векторных данных и реализации механизма быстрого поиска ближайших соседей (*nearest neighbors search*). Векторизация данных позволяет представить их в форме, где объекты с похожими характеристиками располагаются близко друг к другу. Такой подход открывает возможности анализа и сравнения данных с применением математических методов, таких как косинусное сходство или евклидово расстояние. Это ключевая функция для задач, где требуется определить степень схожести между объектами, такими как текстовые документы, изображения или аудиофайлы. Одной из особенностей *ChromaDB* является ее интеграция с современными языковыми моделями. База данных способна работать с векторами, которые генерируются такими популярными инструментами, как *BERT*, *Sentence Transformers* или *OpenAI Embeddings* [23-26]. Эта гибкость делает *ChromaDB* подходящей для разнообразных задач, начиная от анализа текста и заканчивая обработкой мультимедийного контента. Масштабируемость также является важным преимуществом: *ChromaDB* обеспечивает высокую производительность при работе



с большими объемами данных, сохраняя высокую скорость обработки запросов. Гибкость использования достигается за счет предоставления API, который позволяет самостоятельно настраивать методы работы с данными и поиска.

Современные образовательные системы сталкиваются с необходимостью внедрять новые технологии, сохраняя совместимость с существующей инфраструктурой. Именно здесь FastAPI становится ключевым инструментом, позволяя превратить даже сложные алгоритмы, такие как NLP-модели на базе *Sentence Transformers*, в удобные сервисы, доступные для других компонентов вузовской экосистемы. FastAPI обеспечивает бесшовную интеграцию алгоритмов в вузовские системы через REST API. Например, сервис анализа учебных материалов, использующий семантические эмбединги, можно превратить в эндпоинт, который возвращает тематические кластеры для LMS. Автодокументация (*Swagger/OpenAPI*) ускоряет взаимодействие между разработчиками и интеграторами, а валидация данных через Pydantic защищает систему: API автоматически проверяет входные параметры (тип текста, длину строки), минимизируя ошибки. Асинхронная обработка предотвращает зависания при высокой нагрузке – даже во время массовой проверки выпускных работ. Совместимость со стандартными протоколами (*HTTP/WebSockets*) и облачными решениями позволяет

развернуть сервис как в университетском дата-центре, так и в гибридном облаке, сохраняя гибкость для цифровой трансформации вуза. Таким образом, FastAPI не просто предоставляет «обертку» для алгоритма, а становится частью стратегии цифровой трансформации, позволяя вузам внедрять инновации без перестройки всей инфраструктуры.

4 Визуализация алгоритма

На рисунках 2, 3 представлена простая визуализация работы алгоритма, выполненная с помощью библиотеки *Jinja2* (библиотека *Python* для создания динамических *HTML*-страниц). На главной странице (Рис. 2) находятся две кнопки, сначала необходимо выбрать файл студента, то есть его справку об обучении, затем отправить ее на обработку алгоритму.

Отправьте файл

Р и с. 2. Главная страница
F i g. 2. The main page

После ожидания обработки файла с помощью *OCR* и выбора плана обучения с помощью *sentence embeddings*, страница обновится и появится соответствующая информация (на рисунке 3 представлен пример обработки).

Возможное направление студента

Название файла: Иванов справка об обучении.pdf
Наиболее подходящий план обучения: 02.03.03 2024.pdf
Предмет из учебной карточки: Алгоритмы и структуры данных - 144 часов
Предмет из учебного плана: Алгоритмизация и программирование - 324 часов, Схожесть: 0.80
Разница учебных часов: 180
Предмет из учебной карточки: Высокопроизводительная обработка и анализ данных - 180 часов
Предмет из учебного плана: Продвинутое методы работы с данными - 252 часов, Схожесть: 0.77
Разница учебных часов: 72
Предмет из учебной карточки: Деловая и межкультурная коммуникация - 72 часов
Предмет из учебного плана: Психология профессионального взаимодействия и саморазвития - 72 часов, Схожесть: 0.56
Разница учебных часов: 0
Предмет из учебной карточки: Математика для анализа данных - 216 часов
Предмет из учебного плана: Математический анализ - 360 часов, Схожесть: 0.79

Р и с. 3. Страница с ответом
F i g. 3. Answer page

Страница с ответом (Рис. 3) содержит следующую информацию:

- название загруженного файла;
- наиболее подходящий план обучения, выбранный алгоритмом;
- академическая разница обучающегося.

5 Обсуждение и заключение

Предложенное решение может быть встроено в корпоративную информационную систему в качестве внешнего сервиса. Для этого целесообразно воспользоваться встроенными возможностями системы по встраиванию сторонних приложений через *HTTP/REST* или *COM/ActiveX*-интерфейсы. Микросервис, упакованный в *Docker* и развернутый под FastAPI, предоставляет набор REST-эндпоинтов для приёма PDF-документов и возврата результатов анализа (рекомендованного студенту направления обучения, конкретный учебный план, похожие дисциплины и академическую разницу). В системе «Галактика Управление вузом» можно создать стандартный модуль интеграции, который при

определённых действиях (например, при загрузке нового учебного плана или при формировании отчёта) отправляет исходный PDF-документ на API и ожидает от него структурированный ответ в формате JSON. Преимуществом интеграции через FastAPI/Docker является независимость от конфигурации сервера «Галактика Управление вузом», это означает, что весь функционал по OCR, очистке регулярными выражениями, генерации эмбедингов и поиску в ChromaDB полностью инкапсулирован в контейнере. Это гарантирует, что любые отличия в версиях операционной системы или настройках баз данных в системе «Галактика Управление вузом» не повлияют на корректность работы алгоритма. Таким образом, объединяя REST-механизм (FastAPI) и возможности «Галактика Управление вузом» по работе с внешними сервисами, можно добиться частичной автоматизации подготовки аттестационных листов за счет автоматизированного сравнения учебных документов, сохраняя при этом единую точку входа для пользователей – интерфейс «Галактика Управление вузом».



Созданное решение значительно сокращает время ручной сверки учебных планов и справок об обучении, обеспечивает высокую точность обнаружения сходства в наименованиях дисциплин и объёмах часов, а также легко масштабируется и интегрируется в существующие корпоративные информационные системы, благодаря стандартизированному REST-интерфейсу.

Список использованных источников

1. OCR-RTPS: an OCR-based real-time positioning system for the valet parking / Z. Wu, X. Chen, J. Wang [et al.] // Applied Intelligence. 2023. Vol. 53, No. 14. P. 17920-17934. <https://doi.org/10.1007/s10489-022-04362-x>
2. Park Y., Shin Y. Gradual OCR: An Effective OCR Approach Based on Gradual Detection of Texts // Mathematics. 2023. Vol. 11, No. 22. Article number: 4585. <https://doi.org/10.3390/math11224585>
3. Богачев Р. Е., Зариковская Н. В. Реализация серверной части веб систем с использованием фреймворка FASTAPI // Актуальные научные исследования в современном мире. 2021. № 11-15(79). С. 368-373. EDN: QCWPAU
4. Pooja Bhat, Thanuja J. C. Implementing a Google Dialogflow Chatbot for Restaurant Websites â€“ A Serverless Approach with fastAPI // International Advanced Research Journal in Science, Engineering and Technology. 2024. Vol. 11, issue 7. P. 214-219. <https://doi.org/10.17148/IARJSET.2024.11730>
5. Воронин Е. Ю., Ластин В. И. Использование FASTAPI для создания высокопроизводительных REST API // Моя профессиональная карьера. 2025. Т. 2, № 72. С. 92-98. EDN: XUSYJK
6. Никулин В. С. Разработка графического инструмента для проектирования моделей нейронных сетей на базе TensorFlow // Информационные технологии в строительных, социальных и экономических системах. 2025. № 2(36). С. 251-258. EDN: RQCBSR
7. Мадаев Х. Х. Разработка архитектуры кроссплатформенного приложения для интеллектуальной оценки знаний школьников // Тенденции развития науки и образования. 2025. № 123-3. С. 106-110. EDN: BWRIKE
8. Кожевникова П. В., Колесникова Д. В., Сочко С. С. Единое хранилище данных системы поддержки принятия диспетчерских решений // Информационные технологии в управлении и экономике. 2025. № 1(38). С. 12-25. EDN: LGQKQH
9. Ничик С. А. Интеллектуальная система подбора собеседников на основе искусственного интеллекта // Инновации и инвестиции. 2025. № 7. С. 504-506. EDN: QAHXRT
10. Ведьманов И. С., Сивакова Т. В., Судаков В. А. Интеллектуальные технологии для формирования рекомендаций по запросам на естественном языке // Препринты ИПИМ им. М.В. Келдыша. 2025. № 3. С. 1-12. EDN: ALJDCU
11. Putra N. R., Rachmadi R. F., Nugroho S. M. S. Web Service untuk Ekstraksi Informasi KTP Menggunakan Google Cloud Vision // Jurnal Teknik ITS. 2023. Vol. 12, No. 3. P. A227-A232. <https://doi.org/10.12962/j23373539.v12i3.122400>
12. Sun J., Cao W., Yamanaka T. JustDeepIt: Software tool with graphical and character user interfaces for deep learning-based object detection and segmentation in image analysis // Frontiers in Plant Science. 2022. Vol. 13. Article number: 964058. <https://doi.org/10.3389/fpls.2022.964058>
13. Curriculum: curriculum tracking and validation website / A. J. Fajardo Quesada, R. Herrero Pacheco, R. Fajardo Alcalá [et al.] // Seminars in Medical Writing and Education. 2024. Vol. 3. Article number: 394. <https://doi.org/10.56294/mw2024394>
14. Халевин Т. А. Работа с формами и шаблонизатором Jinja2 фреймворка Flask в Python // Дневник науки. 2023. № 6(78). С. 1-7. https://doi.org/10.51691/2541-8327_2023_6_14
15. Django, The Python Web Framework / A. K. Srivastava [et al.] // International Journal of Scientific Research in Engineering and Management. 2022. Vol. 06, issue 05. P. 1-10. <https://doi.org/10.55041/ijrsrem13183>
16. APASview: A web-based platform for alternative polyadenylation analyses in hematological cancers / Xi Hu, J. Song, Ja. Chyr [et al.] // Frontiers in Genetics. 2022. Vol. 13. Article number: 928862. <https://doi.org/10.3389/fgene.2022.928862>
17. Коробко А. В., Шефтель К. С. Модельно-ориентированный подход к формированию оперативных печатных отчётов // Информационные и математические технологии в науке и управлении. 2022. № 4(28). С. 181-189. <https://doi.org/10.38028/ESI.2022.28.4.014>
18. Priyasta D., Hadiyanto, Septiawan R. Enabling EV Roaming Through Cascading WebSockets in OCPP 1.6 // Journal European des Systemes Automatises. 2023. Vol. 56, No. 3. P. 437-449. <https://doi.org/10.18280/jesa.560311>
19. Mazumder J., Mukhopadhyay P. Designing Question-Answer Based Search System in Libraries: Application of Open Source Retrieval Augmented Generation (RAG) Pipeline // Journal of Information and Knowledge. 2024. Vol. 61, issue 5. P. 255-260. <https://doi.org/10.17821/srels/2024/v61i5/171583>
20. Загородников М. В., Михайлов А. А. Восстановление текстового слоя PDF документов со сложным фоном // Труды Института системного программирования РАН. 2024. Т. 36, № 3. С. 189-202. [https://doi.org/10.15514/ISPRAS-2024-36\(3\)-13](https://doi.org/10.15514/ISPRAS-2024-36(3)-13)
21. Козлов С. В., Светлаков А. В. Применение регулярных выражений для обработки текстовых // International Journal of Open Information Technologies. 2022. Т. 10, № 9. С. 82-89. EDN: ZOSEER



22. Макарова Е. А. Обработка слабоструктурированных текстовых данных для использования в моделях анализа // Информационные и математические технологии в науке и управлении. 2023. № 1(29). С. 178-189. <https://doi.org/10.25729/ESI.2023.29.1.015>
23. Yang B., Cao Y., Ren Y. Integrating Vector Databases across Embedding Models // Proceedings of the ACM on Management of Data. 2025. Vol. 3, issue 6. Article number: 338. <https://doi.org/10.1145/3769803>
24. Kulshreshtha S., Garcia J. L. R., Chang C.-Y. Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study // Findings of the Association for Computational Linguistics: EMNLP 2020 ; ed. by T. Cohn, Y. He, Y. Liu. Association for Computational Linguistics, 2020. P. 933-942. <https://doi.org/10.18653/v1/2020.findings-emnlp.83>
25. Siddhanta A., Bhagat A. K. Sentiment Showdown – Sentence Transformers stand their ground against Language Models: Case of Sentiment Classification using Sentence Embeddings // Procedia Computer Science. 2025. Vol. 257. P. 1205-1212. <https://doi.org/10.1016/j.procs.2025.03.161>
26. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) ; ed. by K. Inui, J. Jiang, V. Ng, X. Wan. Hong Kong, China: Association for Computational Linguistics, 2019. P. 3982-3992. <https://doi.org/10.18653/v1/D19-1410>

References

1. Wu Z., et al. OCR-RTPS: an OCR-based real-time positioning system for the valet parking. *Applied Intelligence*. 2023;53(14):17920-17934. <https://doi.org/10.1007/s10489-022-04362-x>
2. Park Y., Shin Y. Gradual OCR: An Effective OCR Approach Based on Gradual Detection of Texts. *Mathematics*. 2023;11(22):4585. <https://doi.org/10.3390/math11224585>
3. Bogachev R.E., Zarikovskaya N.V. Implementation of the backend of web systems using the FASTAPI framework. *Aktual'nye nauçnye issledovaniâ v sovremennom mire*. 2021;(11-15):368-373. (In Russ., abstract in Eng.) EDN: QCWPAU
4. Pooja Bhat, Thanuja J.C. Implementing a Google Dialogflow Chatbot for Restaurant Websites â€“ A Serverless Approach with fastAPI. *International Advanced Research Journal in Science, Engineering and Technology*. 2024;11(7):214-219. <https://doi.org/10.17148/IARJSET.2024.11730>
5. Voronin E.Yu., Lastin V.I. Using FastAPI to create high-performance REST APIs. *Moâ professional'naâ kar'era*. 2025;2(72):92-98. (In Russ., abstract in Eng.) EDN: XUSYJK
6. Nikulin V.S. Development of a graphical tool for designing neural network models based on TensorFlow. *Information Technologies in Construction, Social and Economic Systems*. 2025;(2):251-258. (In Russ., abstract in Eng.) EDN: RQCBSR
7. Madayev Kh.Kh. Razrabotka arkhitektury krossplatformennogo prilozheniya dlya intellektualnoy otsenki znaniy shkolnikov. *Tendentsii razvitiya nauki i obrazovaniya*. 2025;(123-3):106-110. (In Russ., abstract in Eng.) EDN: BWRIKE
8. Kozhevnikova P.V., Kolesnikova D.V., Sochko S.S. Unified Data Storage of the Dispatch Decision Support System. *Information Technologies in Management and Economics*. 2025;(1):12-25. (In Russ., abstract in Eng.) EDN: LGQKQH
9. Nichik S.A. Intelligent Companion Recommendation System Based on Artificial Intelligence. *Innovation & Investment*. 2025;(7):504-506. (In Russ., abstract in Eng) EDN: QAHRXT
10. Vedmanov I.S., Sivakova T.V., Sudakov V.A. Intelligent Technologies for Generating Recommendations Based on Natural Language Queries. *Keldysh Institute Preprints*. 2025;(3):1-12. (In Russ., abstract in Eng) EDN: ALJDCU
11. Putra N.R., Rachmadi R.F., Nugroho S.M.S. Web Service untuk Ekstraksi Informasi KTP Menggunakan Google Cloud Vision. *Jurnal Teknik ITS*. 2023;12(3): A227-A232. <https://doi.org/10.12962/j23373539.v12i3.122400>
12. Sun J., Cao W., Yamanaka T. JustDeepIt: Software tool with graphical and character user interfaces for deep learning-based object detection and segmentation in image analysis. *Frontiers in Plant Science*. 2022;964058. <https://doi.org/10.3389/fpls.2022.964058>
13. Fajardo Quesada A.J., et al. Curriculum tracking and validation website. *Seminars in Medical Writing and Education*. 2024;3:394. <https://doi.org/10.56294/mw2024394>
14. Khalevin T.A. Working with Forms and the Jinja2 Template Engine of the Flask Framework in Python. *Dnevnik nauki*. 2023;(6):1-7. (In Russ., abstract in Eng) https://doi.org/10.51691/2541-8327_2023_6_14
15. Srivastava A.K., et al. Django, The Python Web Framework. *International Journal of Scientific Research in Engineering and Management*. 2022;06(05):1-10. <https://doi.org/10.55041/ijserm13183>
16. Hu X., et al. APAview: A web-based platform for alternative polyadenylation analyses in hematological cancers. *Frontiers in Genetics*. 2022;13:928862. <https://doi.org/10.3389/fgene.2022.928862>
17. Korobko A.V., Sheftel K.S. Model-driven approach to generating operational printed reports. *Informational and mathematical technologies in science and management*. 2022;(4):181-189. (In Russ., abstract in Eng) <https://doi.org/10.38028/ESI.2022.28.4.014>
18. Priyasta D., Hadiyanto, Septiawan R. Enabling EV Roaming Through Cascading WebSockets in OCPP 1.6. *Journal Europeen des Systemes Automatises*. 2023;56(3):437-449. <https://doi.org/10.18280/jesa.560311>



19. Mazumder J., Mukhopadhyay P. Designing Question-Answer Based Search System in Libraries: Application of Open Source Retrieval Augmented Generation (RAG) Pipeline. *Journal of Information and Knowledge*. 2024;61(5);255-260. <https://doi.org/10.17821/srels/2024/v61i5/171583>
20. Zagorodnikov M.V., Mikhaylov A.A. Recovering Text Layer from PDF Documents with Complex Background. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2024;36(3):189-202. (In Russ., abstract in Eng) [https://doi.org/10.15514/ISPRAS-2024-36\(3\)-13](https://doi.org/10.15514/ISPRAS-2024-36(3)-13)
21. Kozlov S.V., Svetlakov A.V. Using Regular Expressions to Process Text Data. *International Journal of Open Information Technologies*. 2022;10(9):82-89. (In Russ., abstract in Eng) EDN: ZOSEER
22. Makarova E.A. Processing of semi-structured text data for use in data analysis models. *Informational and mathematical technologies in science and management*. 2023;(1):178-189. (In Russ., abstract in Eng) <https://doi.org/10.25729/ESI.2023.29.1.015>
23. Yang B., Cao Y., Ren Y. Integrating Vector Databases across Embedding Models. *Proceedings of the ACM on Management of Data*. 2025;3(6):338. <https://doi.org/10.1145/3769803>
24. Kulshreshtha S., Garcia J.L.R., Chang C.-Y. Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study. In: Cohn T., He Y., Liu Y. (eds.) *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics; 2020. p. 933-942. <https://doi.org/10.18653/v1/2020.findings-emnlp.83>
25. Siddhanta A., Bhagat A.K. Sentiment Showdown – Sentence Transformers stand their ground against Language Models: Case of Sentiment Classification using Sentence Embeddings. *Procedia Computer Science*. 2025;257:1205-1212. <https://doi.org/10.1016/j.procs.2025.03.161>
26. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Inui K., Jiang J., Ng V., Wan X. (eds.) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics; 2019. p. 3982-3992. <https://doi.org/10.18653/v1/D19-1410>

Поступила 28.07.2025; одобрена после
рецензирования 14.09.2025; принята к публикации
03.10.2025.

Submitted 28.07.2025; approved after reviewing
14.09.2025; accepted for publication 03.10.2025.

Об авторах:

Зыкина Анна Владимировна, заведующий кафедрой прикладной математики и фундаментальной информатики факультета информационных технологий и компьютерных систем, ФГАОУ ВО «Омский государственный технический университет» (644050, Российская Федерация, г. Омск, пр. Мира, д. 11), доктор физико-математических наук, профессор, **ORCID:** <https://orcid.org/0000-0001-9697-8342>, avzykina@mail.ru

Мунько Виктория Васильевна, старший преподаватель кафедры прикладной математики и фундаментальной информатики факультета информационных технологий и компьютерных систем, ФГАОУ ВО «Омский государственный технический университет» (644050, Российская Федерация, г. Омск, пр. Мира, д. 11), **ORCID:** <https://orcid.org/0000-0003-0569-2459>, vvmunko@omgtu.ru

Шикин Егор Олегович, магистрант кафедры прикладной математики и фундаментальной информатики факультета информационных технологий и компьютерных систем, ФГАОУ ВО «Омский государственный технический университет» (644050, Российская Федерация, г. Омск, пр. Мира, д. 11), **ORCID:** <https://orcid.org/0009-0000-2882-339X>

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Anna V. Zykina, Head of the Chair of Applied Mathematics and Fundamental Informatics, Information Technology and Computer Systems Faculty, Omsk State Technical University (11 Mira Ave, Omsk 644050, Russian Federation), Dr. Sci. (Phys.-Math.), Professor, **ORCID:** <https://orcid.org/0000-0001-9697-8342>, avzykina@mail.ru

Viktoria V. Munko, Senior Lecturer of the Chair of Applied Mathematics and Fundamental Informatics, Information Technology and Computer Systems Faculty, Omsk State Technical University (11 Mira Ave, Omsk 644050, Russian Federation), **ORCID:** <https://orcid.org/0000-0003-0569-2459>, vvmunko@omgtu.ru

Yegor O. Shikin, Master degree student of the Chair of Applied Mathematics and Fundamental Informatics, Information Technology and Computer Systems Faculty, Omsk State Technical University (11 Mira Ave, Omsk 644050, Russian Federation), **ORCID:** <https://orcid.org/0009-0000-2882-339X>

All authors have read and approved the final manuscript.