

А. А. Нецманов^{1✉}, Р. В. Гришин¹

Разработка информационной системы для автоматизированной суммаризации материалов лекций с использованием искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: me@netsmanov.ru

Аннотация. В данной статье описана разработка системы, которая автоматизирует создание конспектов лекций на базе загруженных аудиозаписей. Используя Deepgram API, система распознает текст из предоставленной аудиозаписи, сокращает его до необходимого размера, на основе выбранного уровня детализации с помощью Gigachat и сохраняет полученный результат в базе данных в виде документов формата PDF/DOCX. Пользователь имеет возможность загружать аудиозаписи, просматривать полученные файлы в отдельной вкладке системы, а также фильтровать их по некоторым параметрам. Нельзя не упомянуть, что система оформлена при помощи Flask и имеет интуитивно понятный веб-интерфейс, а также возможность создания личного кабинета для каждого отдельного пользователя. На основе проведенной работы был проведен ряд тестов, по итогам которых было выяснено, что она помогает экономить время и делает лекции более доступными.

Ключевые слова: автоматическое конспектирование, распознавание речи, суммаризация текста, аудиозаписи лекций, веб-приложение, искусственный интеллект

A. A. Netsmanov^{1✉}, R. V. Grishin¹

Development of an information system for automated summarization of lecture materials using artificial intelligence

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: me@netsmanov.ru

Abstract. This article describes the development of a system that automates the creation of lecture notes based on uploaded audio recordings. Using the Deepgram API, the system decomposes the text from the submitted audio recording, reduces it to the required size, based on the selected level of detail using Gigachat, and stores the result in a database in the form of PDF/DOCX documents. The user has the opportunity to download audio recordings, view the received files in a separate tab of the system, and filter them by certain parameters. It should be mentioned that the system is designed using Flask and has an intuitive web interface, as well as the ability to create a personal account for each individual user. Based on the work carried out, a number of tests were conducted, according to the results of which it was found that it Help to save time and makes lectures more accessible.

Keywords: automatic summarization, speech recognition, text summarization, lecture audio, web application, artificial intelligence

Введение

Одной из проблем современного образования является то, что студенты не всегда успевают записать и уловить всю информацию, так как объемы того, что

нужно пройти слишком велики. Из этого следуют, что автоматизация этого процесса нужна, чтобы доступ к пройденным материалам был быстрым и удобным. Решением стала система, способная преобразовать и хранить аудиозаписи лекций в текстовом формате.

Безусловно, попытки создания таких систем предпринимались и ранее, но они зачастую не имеют базу данных для хранения полученной информации и имеют лишь минимальную настройку детализации получаемых лекций. Также проблемой этих сервисов является достаточно высокая цена, которую сможет осилить не каждый студент.

Эта работа была направлена на создание системы, которая будет иметь доступный интерфейс, в котором сможет разобраться каждый, а также технологии распознавания речи, возможность генерации текста, гибкую настройку и хранение полученных данных.

Методы и материалы

Для создания такой системы предварительно были выделены пять ключевых модулей, из которых она будет строиться:

1. Загрузка аудиофайлов.
2. Точное распознавание речи.
3. Суммаризация текста с возможностью дополнения.
4. Генерация итогового документа в популярных форматах.
5. База данных, в которой будет храниться итоговый результат.

Самым сложным этапом стал выбор подходящего API для конвертации аудиозаписей в текст. После просмотра множества вариантов идеальным выбором стал Deerpgram API, поскольку он имеет поддержку русского языка, обеспечивает точную и быструю транскрибацию текста, а также предоставляется бесплатно, в отличие от многих аналогов, в том числе и отечественных, таких как, например, YandexSpeechKit и SaluteSpeech [1–3]. Также основополагающим фактором стало то, что она имеет гибкую настройку и обрабатывает данные в реальном времени [4].

После транскрибации идет этап суммаризации данных полученных из Deerpgram. Для этого был выбран отечественный сервис Gigachat [5–7], потому что он находится в свободном доступе, получает регулярные обновления и постоянно развивается. Это будет постоянно увеличивать качество суммаризации текста, что будет только плюсом в дальнейшем развитии сервиса.

Gigachat позволяет создавать точные конспекты лекций с выбранным пользователем уровнем детализации, а также при необходимости дополняет полученную информацию, исходя из запросов, которые он получает.

Для генерации итогового документа был использован язык разметки Markdown, поскольку он дает возможность отформатировать текст и сделать его понятным [8, 9].

Финальным и немаловажным решением стал выбор библиотеки для создания базы данных, ведь она должна быть удобна в использовании. Среди них есть

два фаворита, которые постоянно используются программистами – это Django и SQLAlchemy. Проведя сравнение было выявлено, что SQLAlchemy больше подходит для создания и дальнейшего развития системы, ведь он дает возможность оперировать более сложными запросами, а также обеспечивает хорошую гибкость реализации уровня доступа к данным [10–12].

Чтобы пользователь мог взаимодействовать с системой необходимо было выбрать в каком формате будет реализована система. Так как в современном мире мы привыкли взаимодействовать с сайтами, ведь к ним можно получить доступ как с настольных ПК, так и с телефона был выбран формат Веб-приложения. Для его реализации был использован фреймворк Flask. Он дает разработчику полную свободу в проектировании архитектуры приложения, поскольку позволяет подключать только необходимые модули [13]. Так как Flask является микрофреймворком он имеет следующие плюсы: простота, гибкость и расширяемость [14–18].

Результаты

В результате проведенной работы была налажена система взаимодействия всех программных компонентов. Веб-интерфейс был организован при помощи микрофреймворка Flask и готовых шаблонов Bootstrap [19, 20], что позволило создать интуитивно понятный интерфейс. На главной странице (рис. 1) следующие элементы управления:

1. Кнопка «Начать», которая переносит пользователя на страницу авторизации.
2. Кнопка «Вход», которая осуществляет те же действия, что и кнопка «Начать»
3. Кнопка «Регистрация», которая переносит пользователя в раздел регистрации.

На страницах регистрации (рис. 2) и авторизации мы можем увидеть поля для ввода Логина и Пароля, а также кнопку подтверждения этих действий.

После прохождения регистрации и дальнейшей авторизации в веб-приложении нас переносит на основную страницу (рис. 3), на который уже виден основной функционал. Загрузка файлов организована как через файловый менеджер, так и через функцию Drag&Drop, которая позволяет просто перетащить аудио-файл в поле предназначенное для этого. Также на этой странице мы можем увидеть Drop-down list, через который необходимо выбирать язык загруженной лекции, Checkbox, предназначенный для добавления смарт дополнений в лекцию, и Slider для выбора уровня детализации.

О из самых важных страниц веб-интерфейса является раздел «Мои Лекции» (рис. 4), в котором отображены все суммаризованные материалы. В этом разделе представлена возможность скачивания и фильтрации лекций по различным параметрам.

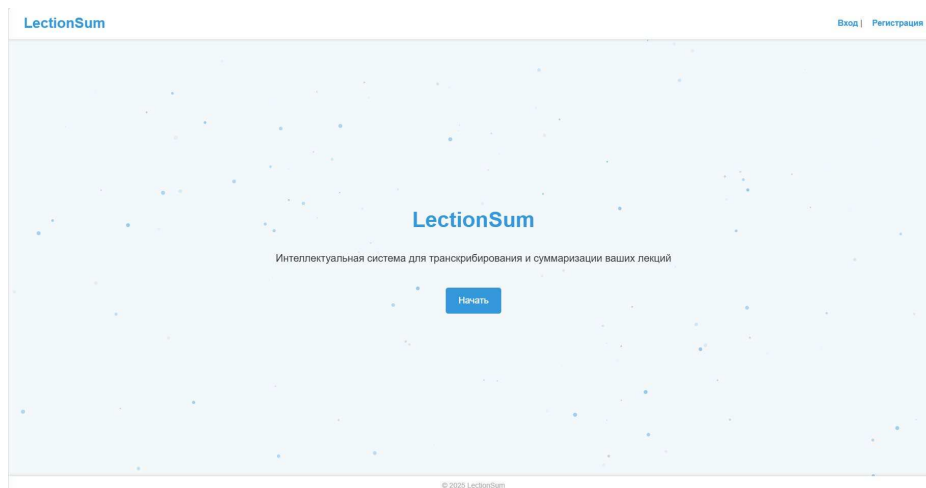


Рис. 1. Главная страница

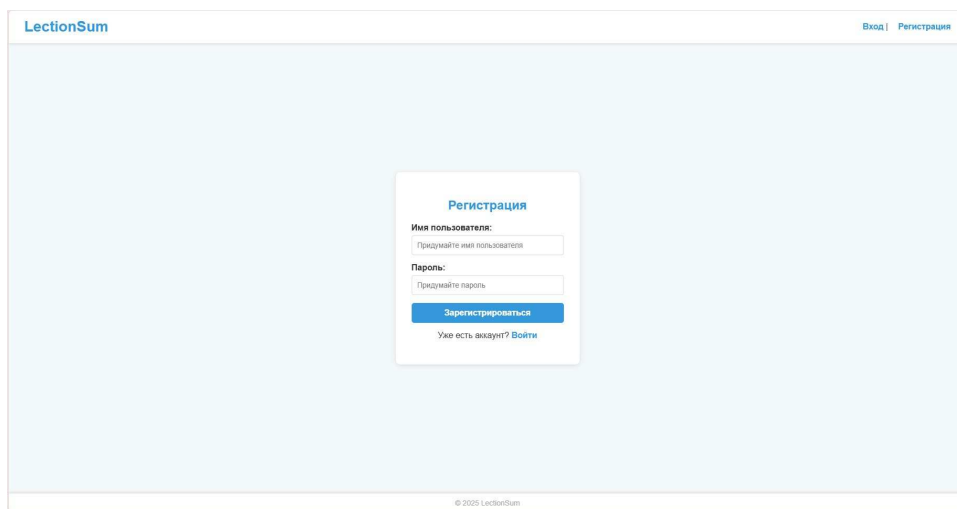


Рис. 2. Страница регистрации

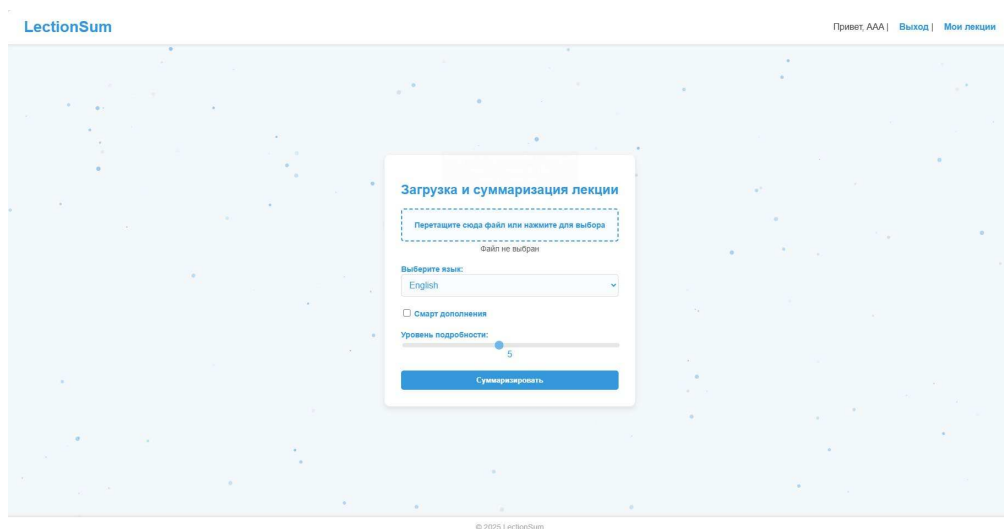


Рис. 3. Страница загрузки файла

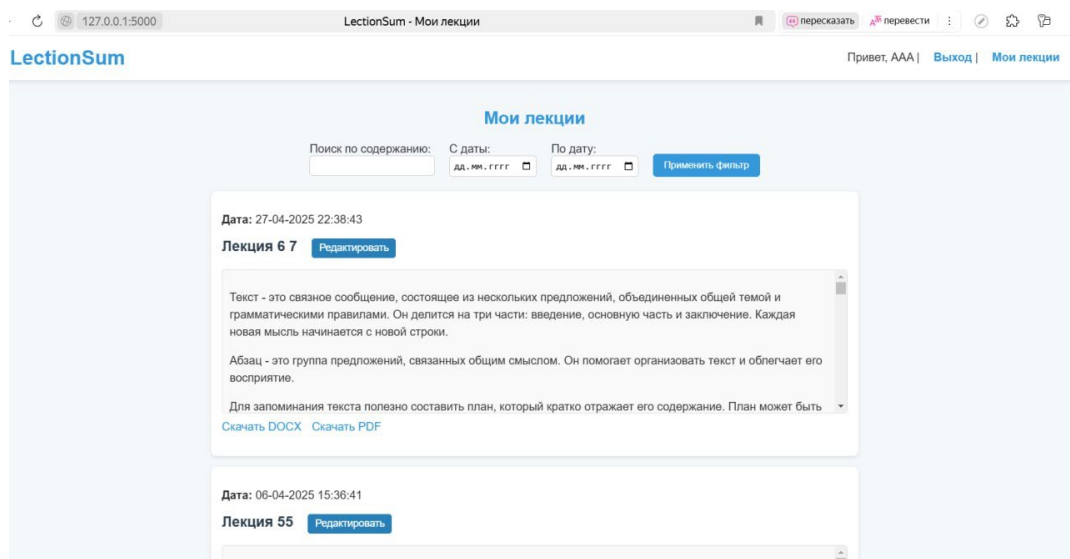


Рис. 4. Страница «Мои лекции»

Обсуждение

По результатам проведенной работы можно с точностью сказать, что интеграция инструментов с искусственным интеллектом в ежедневные рутинные задачи может автоматизировать их и сделать их удобнее и доступнее. К сожалению, сделать на 100 % идеальную систему нет технической возможности, поскольку она напрямую зависит от качества записанных материалов. Даже обычный шум может снизить качество полученных лекций, что повлечет за собой пропуск некоторой информации. Эти недостатки могут быть устранены за счет улучшения качества записывающего устройства или же внедрения очистки аудио. Кроме того, в дальнейшем возможна реализация мобильной версии интерфейса, что позволит еще больше увеличить универсальность созданной информационной системы.

Заключение

Созданная система выполняет автоматическое распознавание речи, суммаризацию текста и генерацию итоговых конспектов. Она может использоваться в вузах и других образовательных учреждениях для ускорения подготовки учебных материалов, а также для упрощения учебного процесса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Transkriptor – Audio to Text APIs. – URL: <https://transkriptor.com/ru/audio-to-text-apis/> (дата обращения: 20.05.2025).
2. Deepgram – Поддержка языков – URL: <https://speechify.com/ru/blog/deepgram-languages/> (дата обращения: 20.05.2025).
3. Connecte AI – Audio to Text: Full Guide – URL: <https://connecte.ai/blog/audio-to-text> (дата обращения: 20.05.2025).
4. AIPure – Обзор Deepgram – URL: <https://aipure.ai/ru/products/deepgram> (дата обращения: 20.05.2025).

5. GigaChat Pro – обзор возможностей – URL: <https://habr.com/ru/companies/sberdevices/articles/780334/> (дата обращения: 20.05.2025).
6. Сравнительное тестирование YandexGPT и GigaChat – URL: <https://3dnews.ru/1107094/sravnitelnoe-testirovanie-yandexgpt-i-gigachat> (дата обращения: 20.05.2025).
7. Российские нейросети: обзор GigaChat и YandexGPT – URL: <https://kingservers.com/blog/rossiyskie-neiroseti-ai-obzor-yandex-gpt-gigachat/> (дата обращения: 20.05.2025).
8. Как использовать Python для работы с Markdown – URL: <https://sky.pro/media/kak-ispolzovat-python-dlya-raboty-s-markdown/> (дата обращения: 20.05.2025).
9. Сравнение Markdown-библиотек для Python – URL: <https://python.libhunt.com/python-markdown-alternatives> (дата обращения: 20.05.2025).
10. Сравнение Django и SQLAlchemy – URL: <https://andreyex.ru/bazy-dannyx/sravnenie-django-i-sqlalchemy/> (дата обращения: 20.05.2025).
11. Что такое библиотека SQLAlchemy для Python – URL: <https://gitverse.ru/blog/articles/data/193-chto-takoe-biblioteka-sqlalchemy-dlya-python-ee-preimushchestva-i-kak-nachat-s-nej-rabotat> (дата обращения: 20.05.2025).
12. Stack Overflow – SQLAlchemy vs Django ORM – URL: <https://stackoverflow.com/questions/3422858/sqlalchemy-vs-django-orm> (дата обращения: 20.05.2025).
13. Создание приложения на Python и Flask – URL: <https://selectel.ru/blog/tutorials/create-app-python-and-flask/> (дата обращения: 20.05.2025).
14. Что такое Flask и как быстро написать простой сайт – URL: <https://ru.hexlet.io/blog/posts/chto-takoe-flask-i-kak-na-nem-bystro-napisat-prostoy-sayt> (дата обращения: 20.05.2025).
15. Сравнение ORM Django и SQLAlchemy – URL: <https://otus.ru/nest/post/280/> (дата обращения: 20.05.2025).
16. Flask vs Django: какой веб-фреймворк выбрать – URL: <https://worksolutions.ru/useful/flask-vs-django-kakoj-web-frejmwork-vybrat/> (дата обращения: 20.05.2025).
17. Django или Flask: что выбрать? – URL: <https://habr.com/ru/articles/867872/> (дата обращения: 20.05.2025).
18. Документация Flask на русском языке – URL: <https://flask-russian-docs.readthedocs.io/> (дата обращения: 20.05.2025).
19. Сравнение CSS-фреймворков: Tailwind, Bootstrap и Materialize – URL: <https://cpbruno.ru/news/programming/sravnenie-css-freymvorkov-tailwind-bootstrap-i-materialize/> (дата обращения: 20.05.2025).
20. Сравнение фреймворков для верстки: Bootstrap, Foundation и другие – URL: <https://ya.zerocoder.ru/pgt-sravnivaem-frejmworki-dlya-verstki-bootstrap-foundation-i-drugie/> (дата обращения: 20.05.2025).

© А. А. Нецманов, Р. В. Гришин, 2025