МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по научно-исследовательской работе в осеннем семестре 2022 года

Тема: Разработка сервиса электронного протоколирования совещаний и защит

Студент гр. 7303 _____ Петров С.А.

Руководитель _____ Шевская Н.В.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
ПОСТАНОВКА ЗАДАЧИ 4
РЕЗУЛЬТАТЫ РАБОТЫ В ОСЕННЕМ СЕМЕСТРЕ 5
ОПИСАНИЕ ПРЕДПОЛАГАЕМОГО МЕТОДА РЕШЕНИЯ 11
ПЛАН РАБОТЫ НА ВЕСЕННИЙ СЕМЕСТР 13
СПИСОК ЛИТЕРАТУРЫ 14

ВВЕДЕНИЕ

Зачастую в академической и деловой сферах проводятся разнообразные встречи и звонки, как в очном, так и электронном виде (с помощью средств электронной телефонии, таких как Zoom [1], Google Meet [2], и т. д.). Данные встречи, звонки и собрания необходимо протоколировать как для статистики, так и для последующей обработки для заполнения разнообразных документов и форм. В настоящее время для подобного заполнения сотрудникам приходится вручную прослушивать аудиозаписи встреч для получения необходимой информации (например, для заполнения протоколов защит бакалавров). Следовательно, есть необходимость в автоматическом протоколировании аудиозаписей звонков/собраний/встреч с транскрипцией [3], статистикой и фиксацией определенных моментов.

ПОСТАНОВКА ЗАДАЧИ

Целью данного исследования является создание программы для автоматического протоколирования аудио собраний/встреч/звонков.

Предполагаемый функционал:

- Транскрипция аудио;
- Составление статистики по участникам (сколько задали вопросов, как участвовали в дискуссии, как быстро отвечали и т. д.);
- фиксация вопросов и задач;

Для достижения цели были поставлены следующие задачи:

1. Провести обзор технологий и библиотеку для транскрипции речи по аудиоданным.
2. Разработать метод определения вопросов в распознанном тексте.
3. Спроектировать и разработать программу для автоматического заполнения документов.
4. Реализовать и проанализировать разработанное решение

Объектом исследования является автоматическое протоколирование.

Предметом исследования является обработка аудиоданных собраний.

Практическая ценность работы: разработанное решение позволит сотрудникам вузов или компаний автоматизировать протоколирование рабочих совещаний (например, очных защит дипломов студентов и аспирантов).

РЕЗУЛЬТАТЫ РАБОТЫ В ОСЕННЕМ СЕМЕСТРЕ

Результаты на момент окончания предыдущего семестра.

В предыдущем (весеннем) семестре 2022 года было разработано веб-приложение (прототип) на языке программирования Python 3.10 [4] и библиотеки для распознавания речи Vosk [5], для расшифровки потокового звука с микрофона и определения вопросов (на русском языке).

Функции:

- запись аудиоданных спикеров и подготовка пула спикеров
- онлайн-обработка потока микрофона с транскрипцией и обнаружением говорящего
- экспорт записанных аудиосессий (метаданные + обнаруженные вопросы)

Ссылка на репозиторий - https://github.com/moevm/zoom-transcriber.

План на осенний семестр.

- Работа над устранением сильной зависимости качества распознавания от качества звука (микрофона), необходимость дополнительной программная обработка на клиентской и/или серверной стороне;
- Работы над устранением зависимости качества распознавания от быстроты и качества речи;
- Распознавание английских терминов в русской речи;
- Отображение плохо распознаваемых слов в клиентской части приложения.

Результаты осеннего семестра.

Работа велась в том же репозитории, в котором велась работа в весеннем семестре 2022 года.

Устранение зависимости качества распознавания от качества входного звука.

В рамках работы по данному направлению было решено добавить средство для очистки входного звука от шума на серверной стороне клиентского приложения. Таким образом, если раньше звуковой отрезок подавался напрямую в библиотеку для распознавания речи, то теперь он будет пропускаться через программную очистку шума.

В качестве программного решения для очистки было принято решение взять Python-библиотеку noisereduce [6] [7]. Данная библиотека вычисляет спектрограмму сигнала (и шумового сигнала, если он подан на вход) и оценивает шумовой порог для каждой полосы частот этого сигнала/шума. Этот порог используется для вычисления маски, которая блокирует шум ниже порога изменения частоты [6].

После внедрения в решение данной библиотеки была проведена серия из 4-х экспериментов.

Каждый эксперимент включал в себя следующие шаги:

1. Выбор исходного отрывка из записи аудио-конференции;
2. Ручная разметка (распознавание) отрывка для получения текстового эталона;
3. Подача звукового отрывка на вход разработанному решению в различных режимах для получения результирующих текстов с распознанной речью
4. Сравнение сходства распознанных отрывков с эталоном;
5. Получение текстовой разницы распознанных отрывков и текста-эталона с помощью сторонних решений.

Для сравнения сходства текстов были выбраны 4 метрики:

- метрика Python-библиотеки для обработки естественных языков Spacy [8] (какой конкретно метод используется в библиотеке, не указано в документации);
- метрика сходства Жаккара – отношение длины пересечения множеств к длине объединения множеств. В качестве множеств в данном случае выступают отдельные слова. В применении к текстам была применена в 2-х видах: с первичной лемматизацией слов (jaccard_raw), и без лемматизации (jaccard_process). Выражается в общем виде следующей формулой:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

;

- метрика сходства на основе евклидового расстояния между двумя векторами – выражается формулой $\frac{1}{e^{\sqrt{\Sigma(x_i - y_i)^2}}}$. Для применения в данной задаче тексты были векторизованы с помощью библиотеки Spacy на основании метода Word2Vec [9]. Таким образом чем ближе расстояние между векторами, тем более 2 текста похожи друг на друга;
- метрика сходства по косинусу – вычисляется по формуле косинуса угла между двумя векторами, подходит в качестве метрики так как угол между двумя параллельным/сходными векторами равен 0. Тексты были векторизованы так же, как и для метрики на основе еквлидового расстояния, с помощью метода Word2Vec;

Эксперименты проводились на записях защит июня 2022 года, предоставленных кафедрой МО ЭВМ.

Результаты проведенных экспериментов не дали возможность выявить значительную разницу между распознаванием речи без очистки от шума и с включенной очисткой от шума, из-за минимального наличия шумов на

представленных записях. Однако, проведенные эксперименты позволили выявить наиболее эффективную конфигурацию библиотеки для очистки шумов (неправильные конфигурации ухудшали качество распознавания при отсутствии шумов). При наличии шумов данная конфигурация должна обеспечить наилучший результат.

Таблица 1 - результаты метрик для одного из экспериментов

Эксперимент/Метрика	spacy	jaccard_raw	jaccard_process	euclidean	cosine
result_only_medfilt	0.9911609499029664	0.6947368421052632	0.7619047619047619	0.7163678114524413	0.9914369794842874
result_no_medfilt_no_sample_nftt4096_timemask64	0.9917372374288371	0.7526881720430108	0.7976190476190477	0.7295124280737039	0.9918639277942686
result_no_medfilt_with_sample_nftt4096_timemask64	0.9914739916524232	0.7263157894736842	0.7882352941176470	0.7252836393861609	0.9915608483163058
result_no_medfilt_no_sample_default	0.9912744131028703	0.7083333333333334	0.7738095238095238	0.7174831528613965	0.9912365720174088
result_no_medfilt_with_sample_default	0.9910903014787334	0.7083333333333334	0.7738095238095238	0.7149979422902976	0.9912101546229997
result_raw	0.9917372374288371	0.7526881720430108	0.7976190476190477	0.7295124280737039	0.9918639277942686

Из таблицы видно, что наивысшие коэффициенты сходства по метрикам наблюдаются у результатов без очистки от шума (results_raw) и с определенной конфигурацией (result_no_medfilt_no_sample_nftt4096_timemask64).

Также были получены текстовые сравнения отрывков по всем экспериментам.

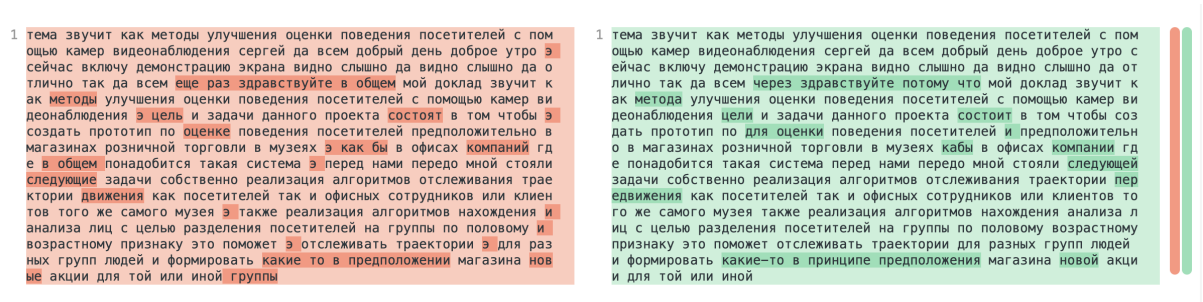

Рисунок 1 – пример текстового сравнения

Результаты по всем экспериментам и ipynb-ноутбук с реализацией метрик сходства и генерацией результатов были загружены на Google Drive и доступны по следующей ссылке - [результаты экспериментов по очистке шума].

Отображение плохо распознаваемых слов в клиентской части приложения.

Из диалога с научным руководителем было принято расширить функционал клиентской части путем добавления выделения слов, которые были “плохо” распознаны используемой библиотекой распознавания речи.

Библиотека в качестве результата выдает коэффициент уверенности для каждого распознанного слова. Данный коэффициент не использовался предыдущем семестре, слова просто конкатенировались в клиентской части без дополнительной смысловой нагрузки. В текущей работе было принято решение использовать данные коэффициенты – слова отправляются на клиент вместе с соответствующими коэффициентами. Клиент (html + javascript + css) рассматривает каждое слово в отдельности вместе с его коэффициентом, и, если коэффициент не достигает настроенного порога (используется значение 0.7), использует html-разметку чтобы выделить слово для пользователя.

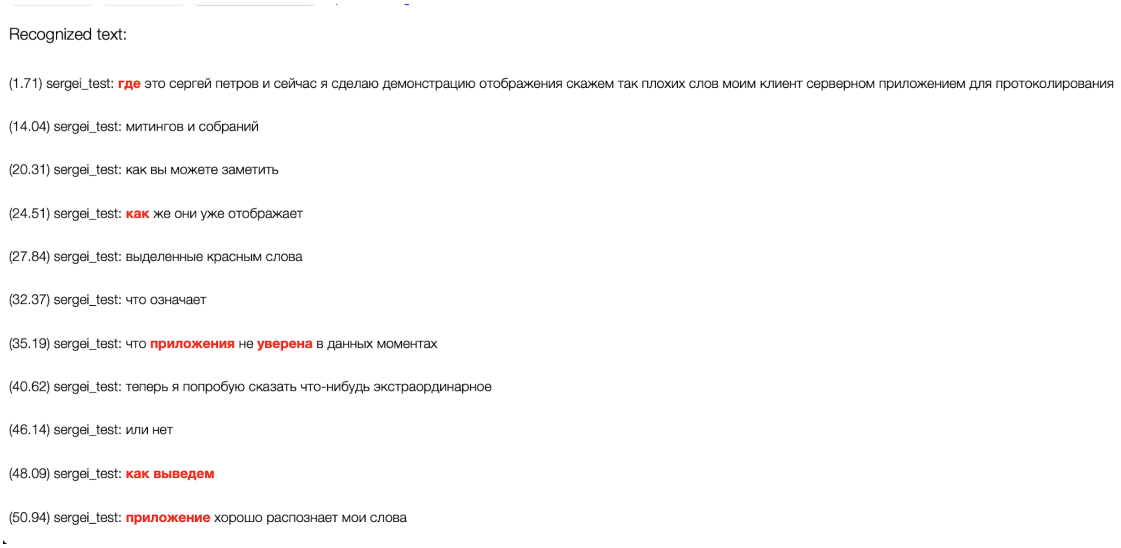

Рисунок 2 – отображение распознанной речи на клиенте с выделением слов с низким коэффициентом распознавания

Другие результаты.

Был расширен формат экспортируемых аудио-сессий, путем добавления колонки с “сырыми” данными распознанных слов (слово + коэффициент распознавания) в формате JSON, которые можно загрузить программными средствами и использовать для дальнейшего анализа.

ОПИСАНИЕ ПРЕДПОЛАГАЕМОГО МЕТОДА РЕШЕНИЯ

Предполагаемое решение представляет из себя клиент-серверное приложения. Сервер представляет из себя веб-приложение на языке Python, клиент – статические файлы HTML + JavaScript/CSS. Данная архитектура упрощает развертывание приложения, а также позволяет сосредоточиться на непосредственно разработке метода распознавания вместе траты времени на разработку клиентских приложений под разные платформы. Передача аудиоданных с клиента осуществляется при помощи языка JavaScript и WebSockets [10], поток аудиоданных с микрофона получается при помощи Web Audio API [11]. Серверная часть принимает поток аудиоданных и обрабатывает его: очищает от шума библиотекой noisereduce и посылает для распознавания другому развернутому веб-серверу, который использует библиотеку для распознавания речи Vosk для транскрибации полученных отрывков. Использование программного решения для распознавания речи в качестве отдельного компонента архитектуры позволяет легко изменить/заменить его в случае необходимости, без изменения кода клиента и основного приложения.

Транскрибированные фразы в виде текста вместе с метаданными (временные метки, метка вопроса, коэффициент точности распознанной фразы) сохраняются в NoSQL-базу MongoDB [12]. База была выбрана из-за отсутствия необходимости хранить нормализованные данные.

Архитектура приложения позволяет легко изменять и расширять правила для определения вопросов, а также дает возможность внедрения дополнительной обработки аудиоданных и распознанного текста.

Приложение позволяет осуществлять экспорт записанных и обработанных аудиоданных. При экспорте пользователь получает zip-архив, содержащий 2 csv-файла:

- records.csv – содержит распознанные фразы спикеров, с временными метками, показателем точности, меткой вопроса, данными по отдельным словам в формате JSON (упакованном в строку);

- metadata.csv – метаданные обработанного аудио/конференции (название, временные метки, распознанные спикеры, и т.п.)

Выгруженный на Google Drive пример экспортированных результатов обработки можно посмотреть здесь.

ПЛАН РАБОТЫ НА ВЕСЕННИЙ СЕМЕСТР

Направления для улучшения/дальнейшего исследования:

- исследование подходов для улучшения распознавания вопросов спикеров, расширение/улучшение набора правил для распознавания вопросов;
- добавление возможности для редактирования получившейся транскрипции/расшифровки совещания/конференции, предложение альтернатив для распознанных слов с низким коэффициентом распознавания;
- оптимизация разработанного решения, в частности взаимодействия между основным сервером и сервером распознавания речи;

СПИСОК ЛИТЕРАТУРЫ

1. Официальный сайт Zoom: [сайт]. URL: https://zoom.us
2. Обзор возможностей Google Meet: [сайт]. URL: https://apps.google.com/intl/ru/intl/ru_ALL/meet/how-it-works/
3. Кибрик А. А., Подлесская В. И. К созданию корпусов устной русской речи: принципы транскрибирования //Научно-техническая информация (серия 2). – 2003. – Т. 6. – С. 5-11.
4. Официальный сайт языка программирования Python: [сайт]. URL: https://www.python.org/
5. Официальная документация библиотеки для распознавания речи Vosk: [сайт]. URL: https://alphacephei.com/vosk/
6. Репозиторий библиотеки noisereduce: [сайт]. URL: https://github.com/timsainb/noisereduce
7. Sainburg T., Thielk M., Gentner T. Q. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires //PLoS computational biology. – 2020. – Т. 16. – №. 10. – С. e1008228.
8. Документация библиотеки Spacy для обработки естественных языков: [сайт]. URL: https://spacy.io/
9. Goldberg Y., Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method //arXiv preprint arXiv:1402.3722. – 2014.
10. Руководство по технологии WebSockets: [сайт]. URL: https://developer.mozilla.org/ru/docs/Web/API/WebSockets_API
11. Документация Web Audio API: [сайт]. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
12. Официальная документация MongoDB: [сайт]. URL: https://docs.mongodb.com/manual.