УДК 004.021 doi: 10.15827/0236-235X.148.576-584 2024. Т. 37. № 4. С. 576–584

Разработка программного комплекса многоканального распознавания и коррекции речевых сообщений на основе алгоритмов машинного обучения в структуре импортозамещения

Ф.Н. Абу-Абед [1]✉, С.А. Жиронкин [1]

[1] Тверской государственный технический университет, г. Тверь, 170026, Россия
[2] Национальный исследовательский Томский политехнический университет, г. Томск, 634050, Россия

Ссылка для цитирования
Абу-Абед Ф.Н., Жиронкин С.А. Разработка программного комплекса многоканального распознавания и коррекции речевых сообщений на основе алгоритмов машинного обучения в структуре импортозамещения // Программные продукты и системы. 2024. Т. 37. № 4. С. 576–584. doi: 10.15827/0236-235X.148.576-584
Информация о статье
Группа специальностей ВАК: 2.3.1
Поступила в редакцию: 09.07.2024 После доработки: 14.08.2024 Принята к публикации: 19.08.2024

Аннотация. Предметом данного исследования является разработка программного комплекса многоканального распознавания и коррекции речевых сообщений на основе машинного обучения. Комплекс призван решать задачу повышения эффективности факторов производства как составляющую моделирования структуры импортозамещения в российской экономике. Актуальность работы заключается в отсутствии программных аналогов и разработок на отечественном рынке программных продуктов в условиях усиления технологических санкционных ограничений. Целью исследования является повышение эффективности внутренних коммуникаций российских компаний. В качестве методов исследования применяются системный анализ, машинное обучение, информационно-телекоммуникационное системное проектирование и объектно-ориентированное программирование. В настоящей статье приведены архитектура и основные компоненты программного комплекса, такие как бот для взаимодействия с пользователями, кэширование данных, долговременное хранение информации и сервис для распознавания и коррекции речевых сообщений с использованием методов машинного обучения. В разработанном авторами приложении решаются задачи развертывания и контейнеризации речевых сообщений с реализацией сервиса распознавания и расшифровки речевых сообщений. Предложенная система основана на кэшировании данных, распределяет нагрузку между независимыми сервисными компонентами с поддержкой контейнеризации, она адаптирована к масштабированию и работе на различных платформах и облачных средах. Интерфейс приложения дает пользователю возможность произвести необходимые настройки с целью автоматического распознавания, диаризации, коррекции и обобщения речевых сообщений. Научная новизна заключается в получении результатов, способствующих оптимизации внутренних коммуникаций с помощью алгоритмов машинного обучения для повышения точности и адаптивности корпоративных систем связи. Данные системы позволяют эффективно решать важную задачу моделирования структуры импортозамещения в условиях усиления внешних шоков и технологических ограничений.

Ключевые слова: структура, распознавание речевых сообщений, машинное обучение, программное средство, пользовательский интерфейс, команды и процедуры, база данных, импортозамещение
Благодарности. Исследование выполнено за счет гранта РНФ № 23-28-01423, https://rscf.ru/project/23-28-01423/

Введение. Решение задачи моделирования структуры импортозамещения в российской экономике связано с определением потенциала рынка отечественных программных продуктов, в наибольшей степени пострадавшего от санкционных ограничений и технологических шоков. В целом острота проблемы импортозамещения технологий и компонентов систем искусственного интеллекта, машинного зрения и обучения, бортовых систем [1] связана с их диффузией в структуре российской экономики как сквозных технологий, глубоко модернизирующих целые отрасли обрабатывающего и добывающего комплекса. Моделирование отраслевой и секторальной структуры импортозамещения должно учитывать перспективы повышения эффективности факторов производства (производительность труда, фондоотдача, рентабельность), которые используются в целевой функции структуры импортозамещения как основные переменные [2]. Возможности их радикального повышения за счет разработки и внедрения отечественных технологий Индустрии 4.0, таких как машинное обучение, распознавание и коррекция речи, отличаются особой актуальностью [3].

Существующие подходы к учету технологического компонента в повышении эффективности факторов производства включают в себя:

– общее положительное влияние технологий Индустрии 4.0 на производительность труда в обрабатывающем секторе, в частности,

за счет сокращения времени вспомогательных операций [4, 5];

– рост производительности труда от внедрения в промышленности широкого ряда АСУ, основанных на искусственном интеллекте [6];

– прирост производительности и рентабельности за счет внедрения систем машинного видения и обучения [7];

– падение производительности и рентабельности предприятий ряда обрабатывающих отраслей из-за санкционных ограничений и разрыва связей с ведущими мировыми производителями программных продуктов.

Учитывая потери российской экономики от санкций и ограничений в сфере передовых производственных технологий Индустрии 4.0 [8], поиск подходов к опережающей разработке программных компонентов, соответствующих уровню сквозных (отраслевых) технологий, играет важную роль в моделировании структуры импортозамещения. В частности, разработка отечественного программного комплекса для распознавания речевых сообщений внутри цифровых коммуникационных платформ, в том числе промышленных, имеет стратегическое значение, поскольку способствует долгосрочному снижению зависимости от зарубежных поставщиков принципиально новых передовых производственных технологий, в которых доля отечественных составляет порядка 1 % от используемых сегодня в России [9].

Для анализа используемых в России программных комплексов распознавания речевых сообщений были рассмотрены популярные зарубежные системы Microsoft Azure Speech, Amazon Transcribe, Google Cloud Speech-to-Text, а также отечественные Сбер Салют и Яндекс Speech API. Выявлено, что при высокой точности распознавания, которая является преимуществом зарубежных систем, отмечены такие недостатки, как высокие затраты, внешний сервис, невозможность улучшать модели, существенные затруднения доступа к сервисам в условиях санкций, непредсказуемость введения новых ограничений функционала при ужесточении санкционного противостояния. К преимуществам отечественных систем следует отнести сам сервис и высокую точность на общих тестах, а к недостаткам – низкое качество распознавания в домене, невозможность радикального улучшения основного функционала, затруднения отраслевой кастомизации, неудовлетворительный внешний сервис.

Для разработки комплекса многоканального распознавания и коррекции речевых сообщений на основе алгоритмов машинного обучения был выбран язык Python как один из самых популярных в мире, а значит, есть большое сообщество разработчиков, которое поддерживает инструменты разработки, создает значительную базу специализированных знаний, в которой можно найти решение для любой возникающей проблемы [10]. Практическим следствием этого является наличие значительного количества библиотек и фреймворков, упрощающих разработку новых продуктов. Большинство библиотек языка написано на C и C++, что способствует достаточно высокой производительности итогового продукта [11]. Наряду с этим язык Python поддерживает достаточное число инструментов для улучшения качества разработки программных продуктов, таких как Jupyter Notebook с интерактивной средой для выполнения и визуализации исследований, а также Hydra с возможностью логирования всех экспериментов и их гибкого конфигурирования. Это имеет важное значение для диффузии технологий машинного обучения в отраслевой структуре промышленности, при создании коллаборативных роботов, использование которых вместе с людьми, а не вместо них, является основой будущей Индустрии 5.0 (ее экспансия ожидается во второй половине 21 в. [12]), направленной на радикальное повышение производительности труда за счет синергии человеко-интеллектуально-машинных систем.

Основа архитектуры системы

Архитектура сервиса многоканального распознавания и коррекции речевых сообщений состоит из следующих ключевых компонентов:

– модуль распознавания – основан на моделях машинного обучения и включает в себя компоненты для обнаружения голоса, диаризации (разделения речи разных говорящих), устранения шумов, а также средства для распознавания речи;

– Discord-бот – основной модуль, реализованный на языке Python с использованием библиотеки discord.py; предназначен для прямого взаимодействия с пользователями Discord и для управления процессом записи речевых сообщений;

– система Redis – используется для временного кэширования данных о голосовых сообщениях, таких как их идентификатор, идентификатор канала Discord и идентификатор сообщения с информацией о сообщении;

– PostgreSQL – реляционная БД для долговременного хранения аудиофайлов и информации о записанных голосовых сообщениях;

– сервер gRPC – обеспечивает API для взаимодействия с записанными голосовыми сообщениями, предоставляя возможность получения метаданных и скачивания аудиофайлов.

Рассматриваемая в данной статье система распознавания и коррекции речевых сообщений спроектирована с учетом требований масштабируемости и высокой производительности при работе с большим количеством голосовых сообщений и пользователей. Этого возможно добиться при реализации следующих решений.

1. Использование СУБД Redis для кэширования данных о голосовых сообщениях, что обеспечивает быстрый доступ к информации и снижает нагрузку на БД PostgreSQL.

2. Разделение ответственности между компонентами системы (Discord-бот, gRPC-сервер, сервис распознавания речи), что позволяет масштабировать их независимо друг от друга пропорционально нагрузке.

3. Контейнеризация с помощью инструмента развертывания приложений Docker Compose, что упрощает масштабирование системы на различных платформах и облачных средах. Это особо ценно для диффузии разработки в разных отраслях экономики.

Общий вид архитектуры предлагаемой системы распознавания и коррекции речевых сообщений представлен на рисунке 1.

Взаимодействие пользователя с системой для записи речевых сообщений происходит следующим образом.

Шаг 1. Пользователь переходит в один из голосовых каналов Discord и ожидает присоединения остальных участников.

Шаг 2. После этого пользователь вводит в любом текстовом канале команду /record.

Шаг 3. Discord-бот подключается к голосовому каналу, в котором находится пользователь, и начинает запись речевых сообщений всех присутствующих участников.

Записанные аудиофайлы сохраняются в Redis и PostgreSQL. API, реализованный на основе gRPC, обеспечивает доступ к аудиофайлам и позволяет интегрироваться с другими системами.

Представим листинг процесса взаимодействия пользователя при записи сообщений:

```
@bot.command()
async def record(ctx): logging.info
   (”Attempting␣to␣record␣call”) voice
   = ctx.author.voice
   if not voice:
```

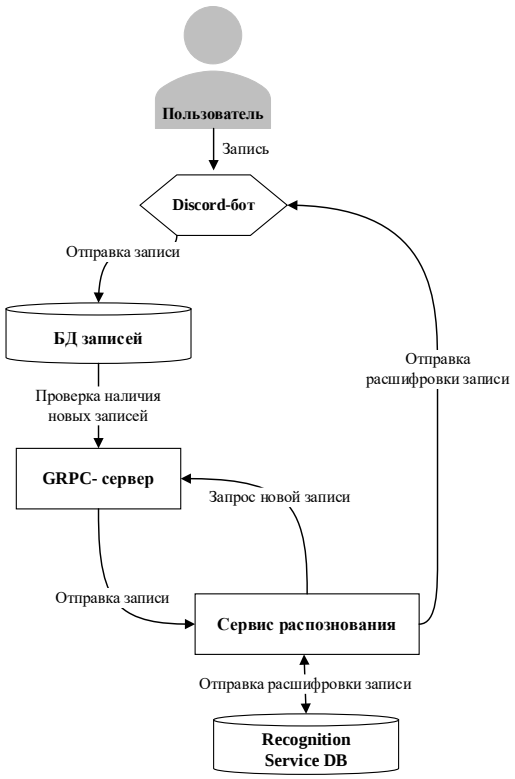

Рис. 1. Архитектура системы распознавания и коррекции речевых сообщений

Fig. 1. Architecture of the speech message recognition and correction system

```
   await ctx.
   respond(JOIN_TO_CHANNEL_TO_
   RECORD)
   return
vc = await voice.channel.connect() #
    Connect to the voice channel the
    author is in.
call_id = uuid.uuid4().hex
start_recording(vc, ctx.channel,
call_id) embed_data = create_new_
record_embed_msg(vc, voice,call_id)
```

Далее Discord-ботом отправляется сообщение в gRPC-сервер с основной информацией о голосовом сообщении (идентификатор, название канала) и кнопкой для завершения записи. После завершения записи, аудиофайлы с голосовыми сообщениями сохраняются в БД PostgreSQL, а информация о сообщении кэшируется в Redis.

Реализация сервиса распознавания и расшифровки речевых сообщений

Для распознавания речи были использованы следующие технологии:

– модель для обнаружения голосовой активности в аудиофайле Silero VAD (Россия);

– программный инструмент с открытым кодом для диаризации речи Pyannote (Франция);

– программный продукт для устранения шумов из аудиозаписей Denoiser (Россия);

– модель для распознавания речи с открытым кодом GigaAM CTC (Россия);

– модель для разделения речевых сообщений Sepformer (США);

– модель для исправления базовых ошибок Yandex Speller (Россия);

– авторская модель для расстановки пунктуации.

В представленном списке большая часть используемых моделей российского производства, а иностранные имеют открытый код.

Что касается валидации, для каждого компонента были выбраны наиболее подходящие технологии и алгоритмы. Для диаризации использовались Audacity и Pyannote, так как они показывали высокие результаты в разделении речи разных говорящих в условиях многоканальных записей (рис. 2).

Для проверки точности моделей были использованы следующие валидационные данные (http://www.swsys.ru/uploaded/image/2024-4/1.jpg):

– порядка 5 часов вручную размеченных аудиоданных;

– оптимизация аудиоданных под задачи обнаружения и для диаризации;

– 200 000 сообщений с мессенджера Discord;

– 30 000 задач из приложения Jira.

Для обнаружения голоса изначально рассматривались две модели машинного обучения: Silero VAD и MarbleNet VAD; по данным авторских тестов, Silero VAD оказался предпочтительнее по основным метрикам – по охвату и по точности. Точность сообщает, насколько положительные предсказания совпадают с фактическими данными. Охват показывает, насколько хорошо модель находит все положительные предсказания. Авторами была проведена оптимизация порогового значения при помощи использованных валидационных данных, что значительно повысило эффективность предлагаемой системы распознавания и коррекции речевых сообщений.

Использование Silero VAD позволило сократить вычисления на 22,26 %, при этом уровень ошибок составил всего 0,6 %. Данная модель машинного обучения, используя метод скользящего окна, анализирует аудиофайлы и с помощью задачи бинарной классификации [13] определяет, присутствует ли человеческий голос в текущем сегменте (около 5-30 мс). Хотя использование модели Silero VAD не является обязательным в предложенной авторами системе распознавания речи, она может быть очень полезна в определенных сценариях. Например, когда аудио уже разделено на отдельные файлы (как это часто бывает с записью, полученной с цифровых платформ), поскольку логично предположить, что люди перебивают друг друга редко (лишь в 2,1 % случаев в рассматриваемом неделимом наборе данных (датасете)). Таким образом, для обработки аудиозаписи длиной 60 минут, содержащей речь пяти участников, необходимо обработать 300 минут аудио. Это значительно превышает объем полезного сигнала, который составляет максимум 60 минут, что позволяет не превышать максимальную длительность обработки объекта полезного сигнала.

Результат работы модели Silero VAD представлен на рисунке 3.

Для обнаружения человеческого голоса были рассмотрены только модели MarbleNet и Silero VAD, другие известные методы Audiotok и WebRTC уже подробно исследованы (https://github.com/wiseman/py-webrtcvad/issues/68).

Для распознавания речи выбор предпочтительной модели производился из трех перспективных: Whisper v3, GigaAM CTC и RNN-T.

Удобство модели Whisper в том, что нет необходимости распознавать пунктуацию. Кроме того, можно распознавать сторонние звуки (например, звуковой фон), однако из-за этого возникают неоднозначные ситуации, когда мо-

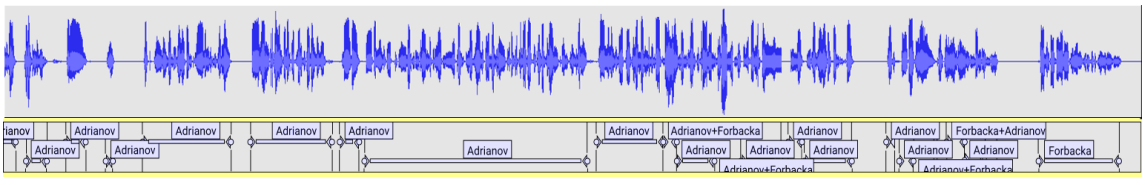

Рис. 2. Пример разметки в Audacity

Fig. 2. Audacity markup example

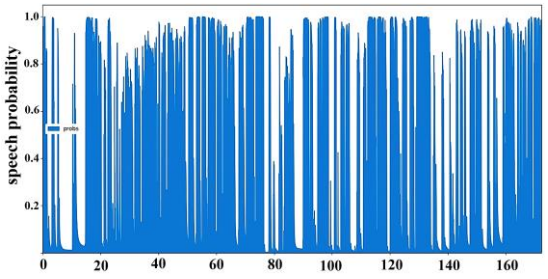

Рис. 3. Работа SileroVAD

Fig. 3. SileroVAD operation

дель распознает не существующие в реальности музыку, шуршание и тому подобное во многом из-за обучения на субтитрах видео. К тому же фактическое качество распознавания русского языка при помощи Whisper оказалось низким. Поэтому выбор происходил между моделями CTC и RNN-T, и на основе полученных данных авторы отдали предпочтение CTC.

После этого была обучена N-граммная модель для рескоринга (процесса валидации гипотез CTC-модели с помощью внешней модели). То есть после предсказаний CTC-модели ее обрабатывают языковой моделью, которая определяет, насколько данный текст правдоподобен. Основной метрикой здесь является WER (*Word Error Rate*) – сумма перестановок, удалений и вставок слов по сравнению с эталоном. С помощью языковой модели WER удалось сократить примерно до 10 %.

Для оценки качества классификации моделей обнаружения голоса зачастую используют метрики Precision (точность) и Recall (полнота). Метрика Precision показывает, насколько точно модель предсказывает положительные случаи. Она измеряет долю истинно положительных предсказаний (TruePositive) среди всех предсказанных положительных случаев (TruePositive и FalsePositive).

Метрика Recall оценивает, насколько полно модель обнаружения голоса охватывает все положительные случаи. Она измеряет долю истинно положительных предсказаний (TruePositive) среди всех фактических положительных случаев (TruePositive и FalseNegative).

После выбора модели необходимо определить оптимальный порог для классификации. Так как данная модель важна в контексте оптимизации прогноза, важно найти компромисс между прогнозами FalseNegative (не определяем фразы, когда они есть) и TrueNegative (места в аудио, которые можно не обрабатывать). Итоги определения оптимального прогноза отражены в таблице.

Результаты при разном пороговом значении классификации

Results at different classification thresholds

Порог	TruePositive, %	FalsePositive, %	TrueNegative, %	FalseNegative, %
0,05	55,77	40,22	3,67	0,35
0,1	55,77	40,22	3,67	0,35
0,15	55,77	37,05	6,84	0,35
0,2	55,77	36,85	7,03	0,35
0,25	55,77	34,74	9,15	0,35
0,3	55,50	34,74	9,15	0,61
0,35	55,50	31,50	12,38	0,61
0,4	55,50	29,35	14,53	0,61
0,45	55,50	27,21	16,68	0,61
0,5	**55,50**	**21,62**	**22,26**	**0,61**
0,55	55,24	18,19	25,70	0,87
0,6	54,03	15,67	28,22	2,09
0,65	51,12	10,68	33,21	4,99
0,7	47,68	8,56	35,32	8,43
0,75	47,42	7,24	36,64	8,69
0,8	46,04	3,60	40,29	10,07
0,85	44,98	3,34	40,55	11,13
0,9	40,40	3,07	40,81	15,71
0,95	35,68	2,81	41,08	20,44

Из табличных данных видно, что наиболее подходящим значением является порог 0,5, который позволяет терять лишь 0,61 % фреймов при сокращении вычислений на 22,26 %.

Диаризация и устранение шумов

Для обработки аудиофайлов, не разделенных по спикерам, важно выявить моменты, когда говорят несколько участников, и разделить их аудиодорожки [14]. Это позволяет обрабатывать их отдельно и параллельно, значительно ускоряя процесс. Для выполнения задачи была использована библиотека pyannote, которая специализируется на диаризации.

Для валидации модели был адаптирован датасет, использованный ранее, содержащий метки моментов речи нескольких человек. Это позволило разделить аудиофайл на фрагменты, каждый из которых содержит речь одного человека, и затем объединить их, имитируя одновременную речь нескольких участников. Процесс синтеза данных для данной задачи представлен на рисунке 4.

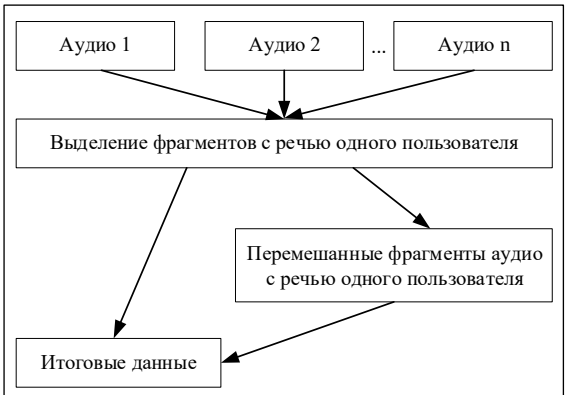

Рис. 4. Синтез данных для валидации модели распознавания и коррекции речевых сообщений

Fig. 4. Data synthesis for validating the speech message recognition and correction model

Расстановка пунктуации

Ввиду отсутствия подходящих решений для расстановки пунктуации была обучена авторская модель. Контекст обработки был расширен в два раза по сравнению с базовой моделью, что позволило обрабатывать до 1 500 слов за один прогон (модель была адаптирована к работе с полилогами). Модель обучалась с помощью разбиения Discord-сообщений на небольшие диалоги и удаления после этого всей пунктуации (рис. 5). Затем пользователей нумеровали, и это добавлялось в контекст; потом производилось обучение на три эпохи в течение 6 часов на одной видеокарте с 24 Гб видеопамяти (NVidia GeForce RTX 3090 Ti).

Обучающая выборка состояла из 200 000 внутренних сообщений компании и предварительной очистки (удаление эмодзи, ссылок, фрагментов кода) с помощью регулярных выражений.

Фрагмент кода для загрузки и сохранения данных представлен в листинге:

```
load_dotenv()
async def load_messages(guild, excluded_channels: list[str]): messages = []
   for channel in guild.text_channels:
    if channel.id not in excluded_channels: logging.info(”Reading␣channel:␣%s”, channel) try:
     async for message in channel.history(limit=None): msg_data = DiscordMessage(channel.id,
       message.content, message.created_at.timestamp(), message.author.id)
       messages.append(msg_data)
    except discord.Forbidden: logging.error(f”Do␣not␣have␣permissions␣to␣read␣
{channel.name}”)
    except discord.HTTPException: logging.error(f”Failed␣to␣read␣history␣for␣
{channel.name},␣skipping...”)
    else:
      logging.info(”Skipping␣channel:␣%s”, channel)
return messages
```

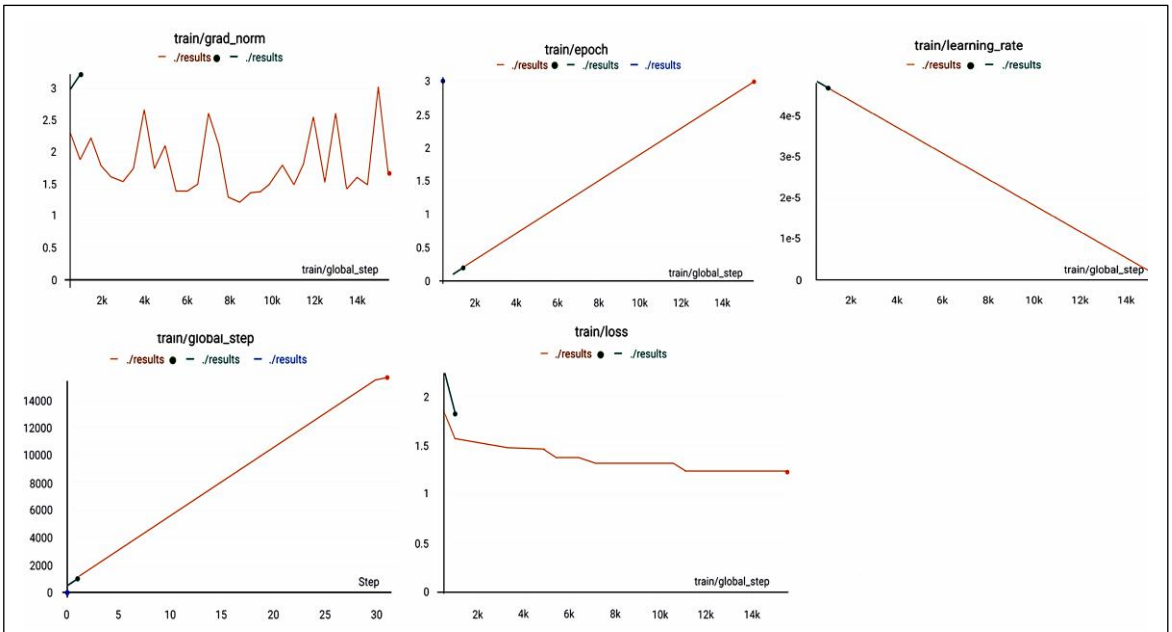

Рис. 5. Динамика обучения пунктуационной модели

Fig. 5. Punctuation model training dynamics

Коррекция речевых сообщений и разделение их на беседы

В результате обоснования моделей для коррекции речевых сообщений было установлено, что для большинства и охват, и точность были ниже 50 % (см. таблицу) даже на общих данных, внутри домена данные отличались еще меньшей точностью. Поэтому было принято решение использовать Yandex Speller, обогащенный данными из диалогов пользователей внутри беседы, что позволило уменьшить количество ложноположительных результатов практически до нуля. Наиболее эффективные решения получены при комбинации Yandex Speller и словарей из Discord и Jira. Иначе говоря, при отсутствии в словаре искомого слова (гипотеза отвергалась), данная модель предлагала заменить его на слово, которое присутствует в словаре

В целях разделения сообщений на беседы данные после загрузки и очистки были предобработаны для выполнения задач с диалоговыми данными. Добавлены токены для обозначения пользователей < _ >, для определения конца сообщения < // >, а также для разделения контекстных и генерируемых данных < / >.

Выводы

Основной целью работы явилось создание программного комплекса, который повышает эффективность внутрикорпоративных коммуникаций и поддерживает политику импортозамещения. Для достижения этой цели были поставлены задачи, включающие исследование существующих методов распознавания и коррекции речи, разработку ПО, интеграцию с цифровыми коммуникационными платформами и создание средств для обобщения речевых сообщений. В ходе анализа существующих программных продуктов были выделены функциональные и нефункциональные требования к программному комплексу. Функциональные требования включают точное распознавание речи, коррекцию орфографических и грамматических ошибок, автоматическую расстановку пунктуации, обнаружение голоса и удаление шума. В состав нефункциональных требований входят высокая производительность, масштабируемость, надежность, совместимость с различными платформами и удобство эксплуатации.

Разработанная система записи речевых сообщений включает ключевые компоненты, такие как Discord-бот для взаимодействия с пользователями, система Redis для кэширования данных, БД PostgreSQL для долговременного хранения информации, а также сервис для распознавания и коррекции речевых сообщений, реализованный с использованием современных методов машинного обучения. В ходе экспериментов и тестирования использованы различные модели для обнаружения голоса, диаризации и удаления шума, что позволило достичь высокой точности и производительности системы.

Заключение

В данной работе продемонстрировано, что современные системы распознавания речи являются результатом многолетних исследований и развития технологий. Использование методов глубокого обучения и облачных технологий значительно расширило возможности применения распознавания речи в реальных приложениях, делая эту технологию неотъемлемой частью современных коммуникационных систем и умных устройств. Разработанная система успешно интегрирована в цифровые коммуникационные платформы, обеспечивает высокую точность и надежность распознавания и коррекции речевых сообщений. Во время работы были выполнены все поставленные задачи. Дальнейшее развитие систем распознавания речи будет связано с улучшением моделей машинного обучения, с повышением их адаптивности к различным условиям и с интеграцией их с другими технологиями искусственного интеллекта.

Список литературы

1. Горелиц Н.К., Гукова А.С., Краснощеков Д.В. Анализ российского программного обеспечения для поддержки жизненного цикла разработки бортовых систем в условиях политики импортозамещения // Тр. ИСП РАН. 2020. Т. 32. № 2. С. 175–190.

2. Абу-Абед Ф.Н., Жиронкин С.А. Моделирование структуры импортозамещения на базе модели системы оптимального распределения // Программные продукты и системы. 2023. Т. 36. № 4. С. 644–653. doi: 10.15827/0236-235X.144.644-653.

3. Абу-Абед Ф.Н. Киберфизические системы и человек в контексте интеллектуального производства Индустрии 4.0 // Экономика и управление инновациями. 2022. № 3. С. 78–87. doi: 10.26730/2587-5574-2022-3-78-87.

4. Тарасов И.В. Технологии Индустрии 4.0: влияние на повышение производительности промышленных компаний // Стратегические решения и риск-менеджмент. 2018. № 2. С. 62–69. doi: 10.17747/2078-8886-2018-2-62-69.

5. Гасанов М.А., Гасанов Э.А., Ашванян С.К., Жаворонок А.В., Жиронкин С.А. Цифровой структурный сдвиг: подход к анализу в современной экономике // Экономика и управление инновациями. 2024. № 2. С. 23–34. doi: 10.26730/2587-5574-2024-2-23-34.

6. Макаров М.Ю. Влияние искусственного интеллекта на производительность труда // Экономика и управление. 2020. Т. 26. № 5. С. 479–486.

7. Miranda S.A.D., Aguilar R.R. Machine learning models in health prevention and promotion and labor productivity: A co-word analysis. Iberoamerican J. of Sci. Measurement and Communication, 2024, vol. 4, no. 1, pp. 1–16. doi: 10.47909/ijsmc.85.

8. Таран Е.А., Слесаренко Е.В., Жиронкин В.С. К вопросу о формировании модели структуры импортозамещения в российской экономике и ее ограничениях в условиях внешних шоков // Экономика и управление инновациями. 2024. № 2. С. 12–22. doi: 10.26730/2587-5574-2024-2-12-22.

9. Таран Е.А., Жиронкин С.А. Структура импортозамещения в российской экономике в условиях внешних шоков. Томск: STT, 2023. 144 с.

10. Магеррамов И.М., Акперов Г.И. Решение задач интернет-маркетинга средствами Python // Вестн. кибернетики. 2022. № 1. С. 29–37.

11. Subero A. The C programming language. In: Programming PIC Microcontrollers with XC8. Berkeley, CA, APress, 2024, pp. 15–33.

12. Абу-Абед Ф.Н. Применение технологий интеллектуального управления и бизнес-проектирования Индустрии 5.0 в Майнинге 5.0 // Экономика и управление инновациями. 2022. № 3. С. 50–59. doi: 10.26730/2587-5574-2022-3-50-59.

13. Lu L., Guo M., Renals S. Knowledge distillation for small-footprint highway networks. Proc. ICASSP, 2017, pp. 4820–4824. doi: 10.1109/ICASSP.2017.7953072.

14. Watanabe S., Barker J.P., Vincent E., Mandel M. CHiME-6 challenge: Tackling multi-speaker speech recognition for unsegmented recordings. Proc. Int. Workshop CHiME, 2020, pp. 1–7. doi: 10.21437/CHiME.2020-1.

Software & Systems doi: 10.15827/0236-235X.148.576-584 2024, 37(4), pp. 576–584

Software complex for multi-channel recognition and correction of speech messages based on machine learning algorithms in terms of import substitution

Fares N. Abu-Abed [1]✉, Sergey A. Zhironkin [2]

[1] Tver State Technical University, Tver, 170026, Russian Federation
[2] National Research Tomsk Polytechnic University,
Tomsk, 634050, Russian Federation

For citation
Abu-Abed, F.N., Zhironkin, S.A. (2024) 'Software complex for multi-channel recognition and correction of speech messages based on machine learning algorithms in terms of import substitution', *Software & Systems*, 37(4), pp. 576–584 (in Russ.). doi: 10.15827/0236-235X.148.576-584
Article info
Received: 09.07.2024 After revision: 14.08.2024 Accepted: 19.08.2024

Abstract. The research focuses on developing a software complex for multichannel recognition and correction of speech messages based on machine learning. The complex solves the problem of increasing the efficiency of production factors as a part of modeling an import substitution structure in the Russian economy. This work is relevant due to the lack of software analogs among domestic software products under conditions of increasing technological sanction restrictions. The study aims to improve the efficiency of internal communications of Russian companies. The authors use system analysis, machine learning, information and telecommunication system design and object-oriented programming as research methods. This paper presents the architecture and the main components of the software complex. For example, a bot for interaction with users, data caching, long-term storage of information and a service for recognizing and correcting speech messages using machine learning methods. The application implements a speech message recognition and decryption service to solve the problems of speech message deployment and containerization. The proposed system is based on data caching. It distributes the load among independent service components with containerization support. It is adapted to scale and work on different platforms and cloud environments. The application interface enables the user to make necessary adjustments in order to automatically recognize, diarize, correct and summarize speech messages. The scientific novelty consists in obtaining results that contribute to the optimization of internal communications using machine-learning algorithms to improve the

accuracy and adaptability of corporate communication systems. These systems allow solving the important problem of modeling the structure for import substitution under the conditions of increasing external shocks and technological constraints.

Keywords: structure, speech message recognition, machine learning, software tool, user interface, commands and procedures, database, import substitution

Acknowledgements. The study was supported by the Russian Science Foundation grant no. 23-28-01423, https://rscf.ru/en/project/23-28-01423/

References

1. Gorelits, N.K., Gukova, A.S., Krasnoschekov, D.V. (2020) 'Analysis of Russian software supporting onboard systems development lifecycle in context of import substitution policy', *Proc. of ISP RAS*, 32(2), pp. 175–190 (in Russ.).

2. Abu-Abed, F.N., Zhironkin, S.A. (2023) 'Russian import-substitution structure based on an optimal distribution system model', *Software & Systems*, 36(4), pp. 644–653 (in Russ.). doi: 10.15827/0236-235X.144.644-653.

3. Abu-Abed, F.N. (2022) 'Cyber-physical systems and human in the context of intelligent production of Industry 4.0', *Economics and Innovation Management*, (3), pp. 78–87 (in Russ.). doi: 10.26730/2587-5574-2022-3-78-87.

4. Tarasov, I.V. (2018) 'Industry 4.0: Technologies and their impact on productivity of industrial companies', *Strategic Decisions and Risk Management*, (2), pp. 62–69 (in Russ.). doi: 10.17747/2078-8886-2018-2-62-69.

5. Gasanov, M.A., Gasanov, E.A., Ashvanyan, S.K., Zhavoronok, A.V., Zhironkin, S.A. (2024) 'Digital structural shift: An approach to analysis in modern economy', *Economics and Innovation Management*, (2), pp. 23–34 (in Russ.). doi: 10.26730/2587-5574-2024-2-23-34.

6. Makarov, M.Yu. (2020) 'The influence of artificial intelligence on labor productivity', *Economics and Management*, 26(5), pp. 479–486 (in Russ.).

7. Miranda, S.A.D., Aguilar, R.R. (2024) 'Machine learning models in health prevention and promotion and labor productivity: A co-word analysis', *Iberoamerican J. of Sci. Measurement and Communication*, 4(1), pp. 1–16. doi: 10.47909/ijsmc.85.

8. Taran, E.A., Slesarenko, E.V., Zhironkin, V.S. (2024) 'On the issue of the formation of a model of the structure of import substitution in the Russian economy and its limitations under conditions of external shocks', *Economics and Innovation Management*, (2), pp. 12–22 (in Russ.). doi: 10.26730/2587-5574-2024-2-12-22.

9. Taran, E.A., Zhironkin, S.A. (2023) *The Structure of Import Substitution in the Russian Economy in Conditions of External Shocks*. Tomsk, 144 p. (in Russ.).

10. Magerramov, I.M., Akperov, G.I. (2022) 'Solving internet marketing problems via Python', *Bull. of Cybernetics*, (1), pp. 29–37 (in Russ.).

11. Subero, A. (2024) 'The C programming language', in: *Programming PIC Microcontrollers with XC8*, Berkeley, CA: APress, pp. 36–45.

12. Abu-Abed, F.N. (2022) 'Application of intelligent management technologies and business design of Industry 5.0 in Mining 5.0', *Economics and Innovation Management*, (3), pp. 50–59 (in Russ.). doi: 10.26730/2587-5574-2022-3-50-59.

13. Lu, L., Guo, M., Renals, S. (2017) 'Knowledge distillation for small-footprint highway networks', *Proc. ICASSP*, pp. 4820–4824. doi: 10.1109/ICASSP.2017.7953072.

14. Watanabe, S., Barker, J.P., Vincent, E., Mandel, M. (2020) 'CHiME-6 challenge: Tackling multi-speaker speech recognition for unsegmented recordings', *Proc. Int. Workshop CHiME*, pp. 1–7. doi: 10.21437/CHiME.2020-1.

Авторы
Абу-Абед Фарес Надимович [1], к.т.н, доцент, декан, aafares@mail.ru
Жиронкин Сергей Александрович [2], д.э.н., профессор, zhironkin@tpu.ru

[1] Тверской государственный технический университет, г. Тверь, 170026, Россия
[2] Национальный исследовательский Томский политехнический университет, г. Томск, 634050, Россия

Authors
Fares N. Abu-Abed [1], Cand. of Sci. (Engineering), Associate Professor, Dean, aafares@mail.ru
Sergey A. Zhironkin [2], Dr.Sci. (Economics), Professor,zhironkin@tpu.ru

[1] Tver State Technical University, Tver, 170026, Russian Federation
[2] National Research Tomsk Polytechnic University, Tomsk, 634050, Russian Federation