III Международная конференция учащихся
НАУЧНО-ТВОРЧЕСКИЙ ФОРУМ

Секция: программирование

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ГОЛОСОВОГО ПОМОЩНИКА «КСЮША»

Автор: Казанов Михаил Романович,
Россия, г. Мурманск, МБОУ г. Мурманска ММЛ, 10 класс

Научные руководители:
Никанорова Елена Анатольевна,
заместитель директора по УВР, МБОУ г. Мурманска ММЛ,
Явдошенко Юлия Ивановна, учитель биологии, МБОУ г. Мурманска ММЛ,

Мурманск
2022

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ГОЛОСОВОГО ПОМОЩНИКА «КСЮША»

Казанов Михаил Романович

Россия, Мурманская область, г. Мурманск МБОУ г. Мурманска «Мурманский международный лицей», 10 класс

Аннотация. Статья посвящена актуальной на сегодняшний день проблеме в любом общеобразовательном учреждении, в том числе в Мурманском международном лицее, отсутствию оперативной помощи для учителей и учеников лицея по настройке и использованию компьютерного оборудования. В работе осуществлен теоретический анализ, используемых в мировой практике голосовых помощников, языков программирования и их возможности. На основе анализа осуществлена разработка программного обеспечения голосового помощника ‘Ксюша’. Проведена апробация, разработанного программного обеспечения, на основе анализа которого были установлены некоторые недочеты в реализациипроекта.

Ключевые слова: голосовой помощник, языки программирования, сервер образовательногоучреждения, программирование, распознавание лиц.

Введение

История создания голосовых помощников начинается с 1930-х годов, когда ученые начали предпринимать попытки распознавать человеческую речь силами технологий. В то время, чтобы создать голосового помощника мешали два фактора:

- существование омонимов — слов с одинаковым звучанием, но с разным значением;
- постоянный шумовой фон, из которого система должна выбирать речь пользователя.

Сейчас для решения этих проблем разработчики используют машинное обучение. Оно учит нейронные сети самостоятельно анализировать контекст и эффективно определять основной источник звука. Надо отметить, что пришли разработчики к этому не сразу — потребовалось как минимум 80 лет подготовительных работ: 1939 год. Советский физик Лев Мясников создал аппарат, способный распознавать человеческую речь — несколько гласных и согласных звуков.

1952 год. Сотрудники лаборатории Bell разработали механизм, который распознавал продиктованные по телефону числа от 1 до 9.

1962 год. Компания IBM представила собственную технологию распознавания речи — Shoebox. Машина распознавала 16 английских слов, 10 цифр и 6

арифметических команд. 1980 год. Инженеры научились применять методы «Скрытой модели Маркова».

Со временем это позволило голосовым системам лучше распознавать речь. Они обрабатывают слово, учитывая несколько предыдущих и предсказывая, что может с ними сочетаться. Скрытая модель Маркова описывает генерацию случайных событий в зависимости от текущегосостояния объекта.

Как мы видим, история создания голосовых помощников развивалась достаточно стремительно, однако в настоящий момент не существует голосовых помощников для работников образовательных организаций. В условиях ведения воспитательной работы в школах, часто бывает необходимо произвести настройку оборудования в актовом зале для концерта или для обеспечения конференций, что сложно, а порой и невозможно, если отсутствует инженер.

Исходя из вышесказанного целью нашей работы является: разработать эффективную модель голосового помощника актуального для общеобразовательных организаций (на примереМБОУ г. Мурманска ММЛ).

В соответствии с целью, нами были поставлены следующие **задачи:**

1. Изучить основы языков программирования: Python, Java, JavaScript, Xcode, C++/C+.
2. Исследовать потребности всех участников образовательных отношений пофункциональной и содержательной нагрузке голосового помощника.
3. Смоделировать содержательные задачи, которые должен понимать и делать голосовойпомощник.
4. Создать «универсальную» модель голосового помощника для общеобразовательногоучреждения.
5. Провести апробацию проекта сервера для голосового помощника «Ксюша».

В соответствии с целью и задачами исследования были определены объект и предмет исследования.

Объект исследования: процесс программирования голосового помощника с точки зрения создания нейронных сетей.

Предмет исследования: создание сервера для голосового помощника.

Гипотеза исследования: в общеобразовательных учреждениях существует потребностьоперативной помощи учителям и ученикам для настройки компьютерного оборудования для решения задач образовательного и воспитательного характера, которая может быть удовлетворена созданием универсального голосового помощника.

Методы исследования.

В нашем исследовании нами были использованы как теоретические, так и эмпирическиеметоды. Методы анализа и синтеза мы использовали для решения задач изучения информации и выделения качественных характеристик программного обеспечения. Метод анкетирования был использован для изучения потребностей по содержательным характеристикам программного обеспечения со стороны учащихся, их родителей и педагогического состава лицея. Методы моделирования и программирования были использованы для решения практических задач по созданию программного обеспечения и сервера голосового помощника. Художественный метод мы использовали для записи голоса и его модуляции.

Основная часть

В настоящее время существует много «бета-помощников», которые начали существовать, но проекты были заморожены по разным причинам. «Зачем вообще нужен голосовой помощник ?» - многие дети, учителя и т.д. задаются данным вопросом, и не каждыйможет дать объяснение.

Голосовые помощники нужны для ускорение каких-то либо действий таких как: «Поиска информации в интернете, открытие файлов, документов и, безусловно, обучения.

Как устроены современные голосовые помощники?

Голосовой помощник – это, активируемое голосом программное обеспечение на смартфоне или другом устройстве, которое может предоставлять пользователю информацию ивыполнять определенные типы задач.

Чтобы воспользоваться ГП нужен микрофон и компьютер со стабильным интернет- соединением.

Нынешние голосовые помощники устроены очень легким для общества способом, многие известные компании создали для своих проектов отдельные сервера, куда добавляют время от времени новые функции и обновления системы.

Рассмотрим голосового ассистента «Алиса» от компании «Яндекс», специалистами был создан отдельный серверный блок, к которому подключаются все девайсы (гаджеты) и ПК. Это позволяет избежать перегрузки серверов компании «Яндекс». С точки зрения маркетингаспециалистами было принято очень правильное решение. Однако с точки зрения эффективности работы гаджетов и простоты их подключения, по нашему мнению, было допущено неоправданное усложнение процесса, выраженное:

А) функции голосового ассистента одобряет компания «Яндекс», они создали

отдельное приложение под названием «Яндекс. Диалоги», в котором программисты пишут код и в дальнейшем проверяют его **виртуально,** тем самым сторонний программист не может загрузить свою функцию на свою умную станцию или иные гаджеты. Для личного пользования приходиться пользоваться тем, что есть, а именно сервисом на ПК;

Б) при каждой продаже умной колонки компания получает выгоду и гарантию, что их устройством будут пользоваться так, как они настроили, у пользователя нет права выбора добавить какие-то функции, а какие-то убрать;

В) приобретая умную станцию, покупатель активирует ее и в течение 20–40 минут станция подключается через интернет по зашифрованному каналу к серверу, который с ней делится информацией. Подобные сервера имеют высокий уровень шифрования, так как на них хранятся не только личные данные граждан, но и иные секретные документы. Таким образом, сторонние программисты не имеют возможности вносить, изменяя в программу и никогда их иметь не будут.

Проанализировав опыт компаний, выпускающих программное обеспечивание для голосовых помощников, мы определили, что голосовой помощник «Ксюша» (далее ГП) должен обладать способностью где-то обучаться сам с помощью нейронных сетей, а также иметь возможность корректировки программного обеспечивание (дописывание кодов) для облегчения освоение новой информации голосовым помощником.

Идея проекта такова: для обеспечения актового зала гарантированной технической поддержкой и гарантированным обучением человека, находящегося на месте звукорежиссёра, мы разрабатываем голосового помощника, который сможет давать консультации по настройке оборудования, запускать обучающее видео для настройки отдельных систем в соответствии с введенным запросом. Надо отметить, что система предполагает первоначальное тестирование пользователя, для допуска его в систему управления актового зала.

Так как мероприятия, проводимые в актовом зале, являются разноплановыми, разнонаправленными, то для них требуется различная аппаратура, имеющая различные настройки. Для отслеживания мероприятий голосовым помощником и участников предполагается создание отдельного приложения на платформе Android. С помощью данного приложения педагог-организатор будет вносить определённые данные по типу: название мероприятия, время, дата, предполагаемый визит гостей, время репетиций. При внесении данных ученики, педагоги, связанные с проведением мероприятий, будут уведомлены о предстоящем мероприятии заранее. Данное

приложение предполагает синхронизации с АИС.

Второй блок функционального применения голосового помощника — это помощь в работе педагога. Он сможет помочь учителям в освоении нового программного обеспечения на местах (в учебных кабинетах). Для реализации этого блока задач нами будет создана библиотека проблемных вопросов с заданным заранее алгоритмом их решения. Если, возникшая проблема не входит в базу данных, то ГП при понимании фразы и после сравнения поступивших данных с данными находящимся на сервере переведет поступивший вопрос в веб- браузер, который поможет подобрать алгоритм.

Ресурсы для реализации проекта

1. Для создания проекта нам было необходимо создать отдельный сервер с хранилищем, которое будет активно 24/7 и для него же сделать бесперебойный блок питания, которыйсможет обеспечить питание серверу в течение 4-5 часов в случае аварийной ситуации.

2. Изучить языки программирования и написать исходный код. **3.** Провести тестирование навыков и просмотр функционирования ГП. **4.** Осуществить доработку проекта.

5. Провести второе тестирование проекта на иных устройствах помимо компьютераразработчика.

6. Создание приложения для мобильных устройств, Windows, Mac Os, Android.

7. Загрузка исходного кода в приложение.

8. Загрузка интерфейса в приложение.

9. Обеспечивание компьютеров учебного заведения микрофонами, колонками, веб-камерами.

10. Проведение испытания.

Оборудование для реализации проекта:

1. ЦОМ (серверное хранилище) модель: AQUARIUS ARRAY NS4300S

2.Ноутбук Apple MacBook Pro 13 2011 года 3.Жесткие диск для хранения информации проекта

СРОКИ РЕАЛИЗАЦИИ ПРОЕКТА: проект был разработан и апробирован на базе актового зала ММЛ и учебных мероприятиях. Начат с января 2021, первая апробация - сентябрь 2021. Предполагается доработка проекта, в части дополнительного функционала, направленного наоблегчение работы учителей

При создании проекта нами было использовано штатное оборудование ММЛ. Для

повышения эффективности работы ГП необходимо обновить техническую базу.

Для реализации проекта с решением практических задач, необходимых для современного образования, потребуется (Документация приведена в дополнительных материалах).

Рыночная стоимость данного оборудования составляет: 1.425.086

Оборудование в кабинетах:

1.ПК, оснащенные периферической

оргтехникой.2.Компьютерные классы

3. Интерактивные доски

При наличии данного оборудования, наш ГП может быть доработан:

1. Онлайн конференции и онлайн совещания могут проходить без инженера;

Репетиции будут проводиться с педагогом-организатором, но без звукорежиссёра, т.е. с помощью приложения для ГП, педагог будет вносить музыку и когда надо её включать не подходя к оборудованию.

2. Педагоги смогут самостоятельно определять формат как учебных, так и внеклассныхмероприятий и настраивать технику под эти задачи.
3. В актовом зале станет возможно создание многофункционального пространства, оснащенного современным техническим оборудованием, для развития творческого и интеллектуального потенциала учащихся через занятия творчеством и использованиековоркинг зон в процессе обучения.

Разработка голосового ассистента ‘Ксюша’

В результате анализа теории по языкам программирования таким как: Python, Java, JavaScript, Xcode(Swift), C++/C+ нами было проведено сравнение, на основе которого мы определили язык программирования нашего проекта. Сравнительная характеристика приведена в Приложении 1.

После определения и изучения языка программирования Python, последовала попытка написания кода для голосового помощника. С самого просто с чего начинают программисты это загрузка некоторых библиотек, связанных с языком программирования Python.[3] После загрузки библиотек последовал этап написание простейшего кода «**print "Hello, World!».** Компьютеру была поставлена задача прочитать фразу: «Hello, world!», что означат «Привет, мир!». С чем компьютер справился очень быстро и легко.

После некоторых комбинаций с кодом. [5] Было принято решение перейти на создание приложения для составления расписания мероприятий, учебных занятий. Приложение составлялось в Xcode [6].

После создания приложения вернулся к созданию кода голосового помощника, он

распознает голос человека с помощью встроенного или внешнего микрофона, для проверки распознания речи был создан отдельный код, в котором голосовой помощник все время повторял фразу «Скажите что-нибудь» и после выводил фразы с фразами которые произнес человек.[7]

После проверки распознавания речи приступил к написанию кода для кодоткрытия приложений, файлов, папок на рабочем столе. После того как код был написан, было проведена проверка на разных типах устройств. Т.к. Написание проходило на ноутбуки с ОС Mac Os, открытие документов, файлов и т.д. На компьютерес ОС Windows, потерпело не удачу. Были написаны два варианта открытия программ. Написание происходило для компьютеров учителей, у которых определенные программы такие как Microsoft Office и т.д.

После проверки работоспособности работа на этом этапе разработки голосового помощника для учителей была завершена.

Для написания голосового помощника для актового зала, нами были определены характеристики звука, которые необходимо распознавать голосовому помощнику: примерная громкость звука на мероприятиях таких как:

1) Первое сентября, линейка, мероприятия, связанные с лицеем.
2) Дискотека и т.д.
3) Филармония и т.д.

У ГП для актового зала будет создана в приложении отдельная инструкция по выставке звукового оборудования на мероприятия, приведенные выше.

У ГП будет определённая характеристика, рекомендуемая для сохранения работоспособности оборудования.

ГП будет иметь возможность открывать программы, которые не обходимы для проверки работоспособности компьютера, муз. Приложения, веб-браузеры и. т.д.

Также ГП будет иметь доступ к приложению, в котором будет педагог делать расписание мероприятий, и при каждом мероприятии ГП будет помогать настраивать оборудование.

После всех определенных характеристик ГП, мы приступили к разработке кода [8] для реализации задумки.

После создания задумки были проведены тесты в том числе тесты на мероприятиях.

В лицее 1 сентября было проведено мероприятие «1 сентября», в котором ГП играл важную роль, как ассистент, заменяющий работу человека.

ГП прошел все проверки для реализации в реальность, и справился почти на все 100%, были некоторые недочеты, которые были исправлены в ходе испытаний.

Заключение

Предложенная нами модель голосового помощника носит универсальный характер и имеет минимальный набор технических требований для введения в образовательный процесс. Создавая данную модель, мы изучили языки программирования Python, Java, JavaScript, Xcode, C++/C+, в основу взят язык программирования Python. При анализировании процесса создания был проведен опрос сотрудников, учащихся лицея, и были сделаны определенные выводы. Были смоделированы содержательные задачи, которое голосовой помощник понимал и мог помочь сотруднику решить данную проблему. Была создана модель голосового помощника и проведены испытания на мероприятиях. Апробация прошла успешна, были исправлены следующие недочеты:

-неправильность вывода фраз

-неправильное объяснение настройки оборудования.

Как мы видим, ввод в функционирование учебного заведения голосового помощника и создание приложения с предложенным функционалом, значительно облегчит жизнь ученикам и учителям, а людям, находящимся за оборудованием актового зала будет легче обучиться использованию техники. Таким образом, наша гипотеза подтверждена полностью. Цель достигнута.

Список литературы

1.Клаус Шваб. Четвертая промышленная революция. – Издательство «Эксмо». 2016. 2.Каталог Python [Электронный ресурс]. – Режим доступа https://gb.ru/posts/knigi-dlya-izucheniya-python

3.Каталог Python [Электронный ресурс]. – Режим доступа:https://techrocks.ru/2019/11/25/5-best-books-on-cpp/

4. Каталог Java [Электронный ресурс]. – Режим доступа: https://mogilev.itstep.by/for-students/7-luchshih-knig-po-java/

5. Каталог JavaScript/ [Электронный ресурс]- Режим доступа: https://htmlacademy.ru/blog/educatiZon/collections/javascript-books

6. Язык Swift/ [Электронный ресурс]- Режим доступа: https://techrocks.ru/2019/07/26/12-books-on-swift-programming-language/

7. Саймон Хайкин. Нейронные сети. Полный курс. 2-е издание / Саймон Хайкин. – М.:Диалектика, 2019.

8. Искусственный интеллект: современный подход (AIMA-2). 2-е издание/ СтюартРассел, Питер Норвиг – М.: Williams, 2019.

9. Разработка голосового интеллектуального ассистента// Елена Дроздова,АлександрКоваленко, Виктор Шавкунов - Republic of Moldova; Lambert Academic Publishing, 2020.

10. Голосовые помощники: чему они научились за последние годы/ [Электронныйресурс]- Режим доступа: https://blog.dti.team/voice-assistants-1/

Приложение 1

СРАВНЕНИЕ	Java	C++	Python	Java Script	Xcode (Swift)
Плюсы	**Объектно-ориентированное программирование.** **Java** — язык высокого уровня с простым синтаксисом и плавной кривой обучения. Стандарт для корпоративных вычислительных систем. Безопасность. Независимость от платформы («Написать один раз и использовать везде»).	Довольно неплохая совместимость с Си. Высокая производительность (скорость работы программ, но не их написание). Доступность литературы, документации (книги Страуструпа, Герберта Шилдта,	**Низкая скорость** выполнения программ, по сравнению с другими языками. **Копирование кода.** При копировании кода с другого ресурса, в некоторых случаях, он может копироваться без сохранения отступов. **Конвертация программы на python в exe.** Программы на python имеют расширение *ру. Для использования без интерпретатора, например, на Windows,	Быстрый для конечного пользователя: сценарий Java написан для клиентской стороны, для поддержки и веб-сервера не требуется поддержка. Он также не нуждается в компиляции на стороне клиента, что дает ему определенные преимущества скорости. **Простота:** JavaScript относительно прост в освоении и реализации. **Универсальность:** JavaScript отлично работает с другими языками и может	Приложение для написание мобильных приложений.

		справочники и документация). Универсальность	его нужно конвертировать в файл с расширением *exe (для этого можно использовать приложение py2exe).	использоваться в самых разных приложениях.	
Минусы	Платное коммерческое использование. Низкая производительность. Низкая производительность. Многословный и сложный код.	Низкоуровневость, которую ещё называют “тяжелое наследие Си”. При программировании могут встречаться свойства, характерные для низкоуровневого программирования, даже там, где они не нужны. Не подходит для создания корпоративных приложений (для их разработки		Безопасность: JavaScript явно добавлен к веб-страницам и клиентским браузерам, он может использовать систему пользователя, поэтому вредоносный код может быть запущен на клиентской машине. Поддержка браузера: JavaScript иногда интерпретируется по-разному разными браузерами. Все больше и больше конкурентов: JavaScript - это очень старый язык	Работает только на Apple продукции

		предпочитают Java или C#).			

Приложение 2

5.
```
from vosk import Model, KaldiRecognizer  # оффлайн-распознавание от Vosk
import speech_recognition  # распознавание пользовательской речи (Speech-To-Text)
import wave  # создание и чтение аудиофайлов формата wav
import json  # работа с json-файлами и json-строками
import os  # работа с файловой системой

def record_and_recognize_audio(*args: tuple):
    """
    Запись и распознавание аудио
    """
    with microphone:
        recognized_data = ""

        # регулирование уровня окружающего шума
        recognizer.adjust_for_ambient_noise(microphone, duration=2)

        try:
            print("Listening...")
            audio = recognizer.listen(microphone, 5, 5)

            with open("microphone-results.wav", "wb") as file:
                file.write(audio.get_wav_data())

        except speech_recognition.WaitTimeoutError:
            print("Can you check if your microphone is on, please?")
            return

        # использование online-распознавания через Google
        try:
            print("Started recognition...")
            recognized_data = recognizer.recognize_google(audio, language="ru").lower()

        except speech_recognition.UnknownValueError:
            pass

        # в случае проблем с доступом в Интернет происходит попытка
        # использовать offline-распознавание через Vosk
        except speech_recognition.RequestError:
            print("Trying to use offline recognition...")
            recognized_data = use_offline_recognition()

        return recognized_data
```

```
def use_offline_recognition():
    """
    Переключение на оффлайн-распознавание речи
    :return: распознанная фраза
    """
    recognized_data = ""
    try:
        # проверка наличия модели на нужном языке в каталоге приложения
        if not os.path.exists("models/vosk-model-small-ru-0.4"):
            print("Please download the model from:\n"
                  "https://alphacephei.com/vosk/models and unpack as 'model' in the current folder.")
            exit(1)

        # анализ записанного в микрофон аудио (чтобы избежать повторов фразы)
        wave_audio_file = wave.open("microphone-results.wav", "rb")
        model = Model("models/vosk-model-small-ru-0.4")
        offline_recognizer = KaldiRecognizer(model, wave_audio_file.getframerate())

        data = wave_audio_file.readframes(wave_audio_file.getnframes())
        if len(data) > 0:
            if offline_recognizer.AcceptWaveform(data):
                recognized_data = offline_recognizer.Result()

                # получение данных распознанного текста из JSON-строки
                # (чтобы можно было выдать по ней ответ)
                recognized_data = json.loads(recognized_data)
                recognized_data = recognized_data["text"]
```

```
    except:
        print("Sorry, speech service is unavailable. Try again later")
    return recognized_data

if __name__ == "__main__":
    # инициализация инструментов распознавания и ввода речи
    recognizer = speech_recognition.Recognizer()
    microphone = speech_recognition.Microphone()
    while True:
        # старт записи речи с последующим выводом распознанной речи
        # и удалением записанного в микрофон аудио
        voice_input = record_and_recognize_audio()
        os.remove("microphone-results.wav")
        print(voice_input)
```

5.https://habr.com/ru/post/529590/

(код)6.

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func showMessage(sender: UIButton) {
        let alertController = UIAlertController(title: "Welcome to My First App", message: "Hello World", preferredStyle: UIAlertController.Style.alert)
        alertController.addAction(UIAlertAction(title: "OK", style: UIAlertAction.Style.default, handler: nil))
        present(alertController, animated: true, completion: nil)
    }
}
```

(Стандартное значение кода в Xcode, если вставить в вторичное приложение, результата не будет, https://rational-lab.com/news/start-v-ios-programmirovanii)

7. {

```
  "Can you check if your microphone is on, please?": {
    "ru": "Пожалуйста, проверь, что микрофон включен",
    "en": "Can you check if your microphone is on, please?"
  },
```

```
 "What did you say again?": {
   "ru": "Пожалуйста, повтори",
   "en": "What did you say again?"
  },
}
```

https://habr.com/ru/post/529590/, команды которые произносит ГП если не понимает о чем говорит человек.

8. # подготовка корпуса для распознавания запросов пользователя с некоторой вероятностью

```
# (поиск похожих)

vectorizer = TfidfVectorizer(analyzer="char", ngram_range=(2, 3))

classifier_probability = LogisticRegression()
classifier = LinearSVC()

prepare_corpus()


while True:
   # старт записи речи с последующим выводом распознанной речи
   # и удалением записанного в микрофон аудио
   voice_input = record_and_recognize_audio()


   if os.path.exists("microphone-results.wav"):
       os.remove("microphone-results.wav")
   print(colored(voice_input, "blue"))


   # отделение команд от дополнительной информации (аргументов)
   if voice_input:
       voice_input_parts = voice_input.split(" ")
```

```
    # если было сказано одно слово - выполняем команду сразу
    # без дополнительных аргументов
    if len(voice_input_parts) == 1:
        intent = get_intent(voice_input)
        if intent:
            config["intents"][intent]["responses"]()
        else:
            config["failure_phrases"]()

    # в случае длинной фразы - выполняется поиск ключевой фразы
    # и аргументов через каждое слово,
    # пока не будет найдено совпадение
    if len(voice_input_parts) > 1:
        for guess in range(len(voice_input_parts)):
            intent = get_intent((" ".join(voice_input_parts[0:guess])).strip())
            if intent:
                command_options = [voice_input_parts[guess:len(voice_input_parts)]]
                config["intents"][intent]["responses"](*command_options)
                break
            if not intent and guess == len(voice_input_parts)-1:
                config[«failure_phrases"]()
```

https://habr.com/ru/post/529590/, пример распознавания запросов с н