

Информатика

УДК 004.415.2.031.43

Разработка и использование голосового помощника на примере программного приложения «Омикрон»

А.В. ДОЛЖЕНОК, Н.Б. ОСИПЕНКО

Статья посвящена описанию особенностей разработки голосового помощника и принципов его работы в виде программного прототипа «Омикрон» на языке программирования Python с использованием его библиотек и модулей. Команды «Омикрона» работают как при помощи Интернета, так и без него, в некоторых командах задействовано использование баз данных NoSQL и MSSQL Server. Разработанный программный продукт «Омикрон» позволяет выполнить ряд голосовых команд пользователя и управлять игрой. Приведены результаты верификации работы приложения.

Ключевые слова: голосовой помощник, распознавание речи, искусственный интеллект.

The article is devoted to the description of the development features of the voice assistant and the principles of its operation in the form of the Omicron software prototype in the Python programming language using its libraries and modules. Omicron commands work both with and without the Internet, some commands use NoSQL and MSSQL Server databases. The developed software product “Omicron” allows you to execute a number of user voice commands and control the game. The results of verification of the application operation are given.

Keywords: voice assistant, speech recognition, artificial intelligence.

Введение. Голосовые помощники – это боты, которые работают при помощи распознавания голоса и обработки естественного языка, чтобы отвечать на вопросы, вести разговоры, осуществлять запуск простых задач. С появлением такой опции выполнять многие запросы стало намного удобнее. Голосовые помощники построены на искусственном интеллекте и технологиях машинного обучения. Одна из первых попыток реализации голосового ввода в поисковых системах была реализована в 2002 г. Google, а первой полноценной интерактивной помощницей, поддерживающей осмысленный диалог с пользователем, стала система Siri, введенная в 2007 г. Далее появились Cortana (Microsoft), Alexa (Amazon) и AliGenie (Alibaba). В 2017–2019 гг. появились русскоязычные приложения: Алиса (Яндекс), Олег (банк «Тинькофф») и Маруся (Mail.Ru Group), не уступающие своим зарубежным «коллегам», особенно в распознавании русскоязычной речи.

Голосовые помощники, представленные на IT рынке, постоянно расширяют свои возможности, приобретают новые функции. При этом у каждого из них есть свои преимущества и недостатки [1]. Актуальным направлением развития помощников является их активная интеграция в «умный» дом: свои ассистенты присутствуют у видеокамер безопасности, термостатов и даже бытовой техники в рамках «интернета вещей». Голосовые помощники берут на себя рутинные задачи и сокращают расходы компании на их выполнение. Замена голосовыми ботами сотрудников контактных центров позволяет экономить ресурсы предприятий и компаний.

В настоящей статье предлагается описание прототипа голосового помощника «Омикрон», спроектированного в среде IBM Rational Rose на основе нотации языка UML и разработанного на языке программирования Python. Сегодня использование голосового помощника является одним из перспективных направлений IT рынка. Его используют не только крупные IT компании, а также различные колл-центры, банки. Использование голосового помощника в колл-центре в виде голосового бота уменьшает время ожидания обращающихся граждан, так как данный бот запрограммирован на ответы, вопросами для которых являются часто задаваемые обращения. Внедрение голосового помощника в телевизоры, компьютеры и телефоны способствует упрощению работы с данными устройствами.

Проектирование приложения. Диаграмма вариантов использования программного приложения голосового помощника «Омикрон» представлена на рисунке 1. На ней показана совокупность выполняемых системой действий, которые имеют значение для пользователя, и отвечают на вопрос «Что должна выполнять система?», не отвечая на вопрос «Как она должна выполнять это?». Команды «Омикрона» работают как при помощи Интернета, так и без него, в некоторых командах задействовано использование базы данных MS SQL Server для хранения истории пользователя и MongoDB – для дополнительных сведений, которые были запрошены у голосового помощника.

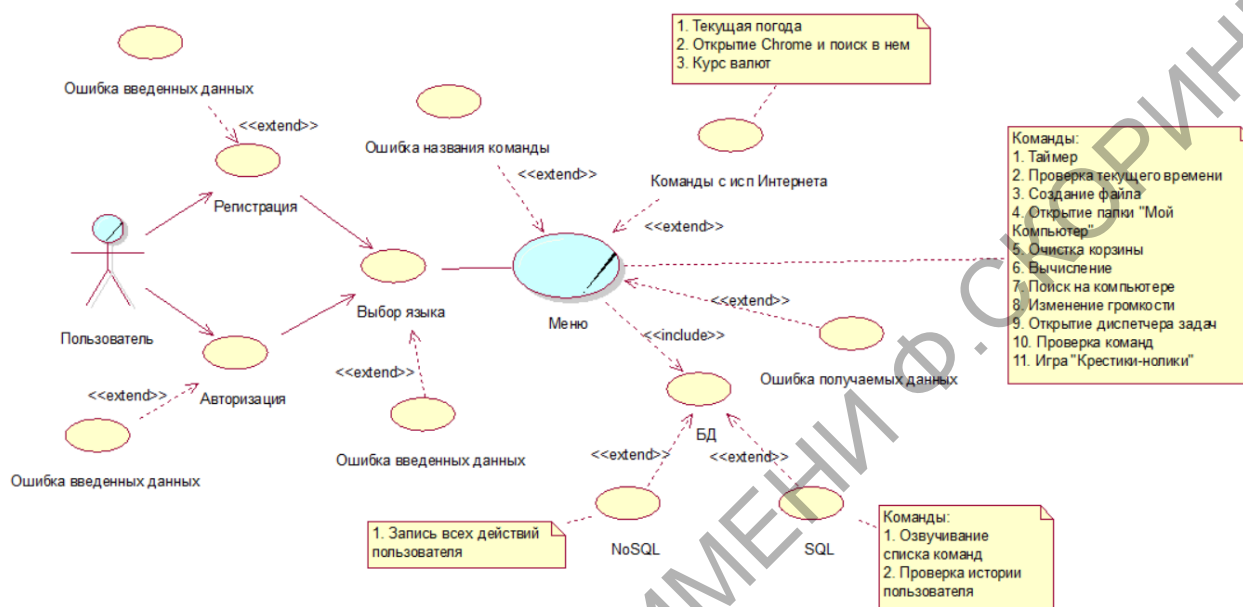


Рисунок 1 – Диаграмма вариантов использования прототипа голосового помощника «Омикрон»

На данный момент разработанный программный продукт «Омикрон» позволяет выполнить голосовые команды пользователя: включение таймера, озвучивание текущего времени и погоды, создание файла, открытие папки «Мой компьютер», очистка корзины, открытие и поиск в браузере Chrome, вычисление простых математических действий, поиск файлов, изменение языка раскладки и громкости, озвучивание курса валют, открытие диспетчера задач, описание всех команд, которые он умеет делать, просмотр истории и играть в «Крестики-Нолики». Для удобства использования голосового помощника в будущем для отслеживания истории обращений к приложению «Омикрон» была разработана авторизация пользователя с выбором языка, которая осуществляется при помощи графической библиотеки Tkinter.

Особенности реализации игрового приложения. При реализации программы на языке программирования Python создается новый проект в PyCharm, куда подключаются модули, библиотеки и пакеты (через import) os, time, pyttsx3, speech_recognition, docx, pyautogui, webbrowser, requests, pyodbc. Модуль os позволяет работать с основными задачами операционной системы, будь то платформа Windows или Linux. Модуль time предлагает множество способов представления времени в коде, таких как объекты, числа и строки; а также позволяет выполнять функции, отличные от представления времени, такие как ожидание во время выполнения кода и измерение производительности кода. Библиотека Pyttsx3 используется для преобразования текста в речь Python. В отличие от альтернативных библиотек он работает в автономном режиме и совместим с Python 2 и Python 3. Приложение вызывает «pyttsx3.init()», чтобы получить ссылку на Pyttsx3. Модуль speech_recognition – это система распознавания речи, в основном переводит произнесенные слова в текст. Преимущество использования системы распознавания речи в том, что она преодолевает грамотность. Модель распознавания речи может использоваться разной по грамотности аудиторией, сосредоточившись на устных аргументах. Библиотека docx – это библиотека для создания и обновления файлов Microsoft Word (.docx). Модуль PyAutoGUI предоставляет скриптам Python кон-

троль над мышью и клавиатурой для автоматизации взаимодействия с другими приложениями. Модуль `winshell` – это легкая оболочка для функций оболочки Windows. Библиотека `requests` – это простая HTTP-библиотека. Пакет `pyodbc` позволяет подключиться к базе данных SQL с помощью Python. Модуль `random` позволяет использовать метод `random.randint(a, b)`, который возвращает случайное целое число из промежутка $[a, b]$.

На основе упомянутых модулей, библиотек и пакетов Python созданы две основные функции прототипа голосового помощника:

1) функция с параметром (строка, которая должна озвучиваться), преобразующая текст в речь, используя специальную библиотеку преобразования;

2) функция без параметра для преобразования речи в текст.

На их основе можно свободно создавать любые функции для команд, которые будет использовать голосовой помощник.

Реляционная база данных MS SQL Server используется в прототипе голосового помощника «Омикрон» для записи однотипных и структурированных данных (для хранения истории пользователя), а нереляционная база данных Mongo DB – для хранения различных по типу и количеству получаемых данных (для хранения дополнительных сведений, которые были запрошены у голосового помощника).

С целью апробации возможности реализации игрового компонента в голосовом помощнике была выбрана известная детская игра «Крестики-нолики», она осуществлена с помощью языка программирования Python, благодаря которому был внедрен простой способ реализации искусственного интеллекта. На примере игры «Крестики-Нолики» показан принцип использования искусственного интеллекта при принятии решений голосовым помощником, который самостоятельно продумывает и осуществляет ходы.

Реализация ходов игрока в игре «Крестики-нолики». Игра «Крестики-нолики» запускается как одна из команд голосового помощника при помощи ключевого слова «игра» / «game», после чего голосовой помощник ожидает команду. После того, как пользователь скажет ключевые слова, в консоли запускается приветствие и рисуется помеченное цифрами из интервала $[1, 9]$ поле размером 3×3 . Голосовой помощник спрашивает у пользователя, на какой квадратик он хочет поставить свой X. Пользователь должен ответить цифру из диапазона от 1 до 9. Если же пользователь назовет другое число, прототип голосового помощника «Омикрон» сообщит ему об этом.

Данная игра предусматривает одного игрока. В роли второго игрока выступает компьютер. Ходы компьютера и пользователя сопровождаются голосом.

При создании поля формируется вектор `board[10]` с первоначальным значением в виде пустой строки. Функция `insertLetter()` принимает два параметра: `letter` и `pos`. Значение `pos` определяется с помощью вектора `board[10]`. «Омикрон» вставит символ «X» в заданную игроком позицию, которая будет храниться в `board[10]`. Функция `spaceIsFree(pos)` сообщит нам, свободно ли данное пространство (отсутствуют ли в некотором элементе игрового пространства «X» или «O»). У нее есть один параметр `pos`, который будет цифрой от 1 до 9. Функция `printBoard(board)` принимает поле в качестве параметра и выводит его. Функция `isWinner()` сообщит нам, выиграл ли данный символ на текущем поле. У функции два параметра: `bo` (`board`) и `le` (`letter`). Символ должен быть «X» или «O». Мы просто проверим каждую возможную выигрышную линию на доске и посмотрим, заполнена ли она данным символом. Функция `main()` вызывается, чтобы начать игру. Функция `isBoardFull()` принимает `board`, как параметр, и просто возвращает `True`, если поле заполнено, и `False`, если это не так. Функция `playerMove()` запрашивает у пользователя число, на какое место ставить «X», и проверяет его. Если число правильное, мы добавляем «X» в игровое поле, иначе – голосовой помощник будет ожидать, пока пользователь не скажет верное число.

Реализация ходов компьютера в игре «Крестики-нолики». Поведение компьютера описывает функция `compMove()` с использованием искусственного интеллекта. Она изучает все поле и определяет, какой ход лучше сделать на основании правил, по которым формируется алгоритм ходов компьютера: если есть выигрышный ход, то берется он; если у пользователя есть выигрышный следующий ход, то компьютер занимает позицию, чтобы пользова-

тель не победил; компьютер занимает позицию в уголках, если свободно больше одного; компьютер занимает центральную позицию; компьютер занимает боковую позицию, если доступно больше одной; если ход невозможен, игра заканчивается с результатом ничья. Функция `selectRandom(li)` определит случайным образом, куда следует сделать ход, учитывая список возможных позиций. Функция `tic_tac_toe_eng()` запускает весь процесс игры и от пользователя зависит, начнется эта игра или нет.

Результаты апробации. Приложение прошло верификацию на тестовых данных. Верификация прототипа голосового помощника «Омикрон» с помощью метода `matplotlib.pyplot` модуля `matplotlib` показала, что при правильном произношении команды он работает точно, иначе просит повторить команду. На рисунке 2 приведен анализ работы команды голосового помощника «Открыть Chrome» этим методом. Для того чтобы открыть Chrome, необходимо сказать «Омикрону»: «Открой Хром», на данный запрос «Омикрон» ответит: «Я в процессе открытия браузера». После чего откроется браузер и «Омикрон» спросит: «Что ты хочешь найти?». Получив ответ от пользователя, «Омикрон» найдет ответ на интересующий пользователя вопрос. На графике видно, что прототип голосового помощника «Омикрон» имел некоторые сложности при выполнении запроса на английском языке, что с большой вероятностью зависит от акцента пользователя; при использовании русского языка прототип распознает речь без ошибок.

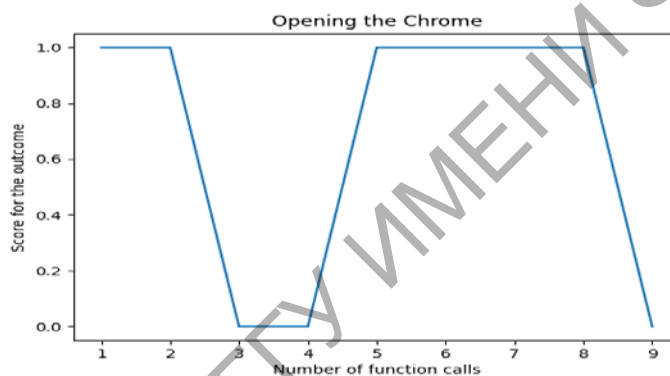


Рисунок 2 – Результат выполнения работы команды «Открыть Chrome»

Была проведена апробация игры «Крестики-нолики» с компьютером, который формирует ответные ходы на основе разработанного алгоритма. На рисунке 3 приведено семь ходов: четыре хода пользователя, обозначенных символом «X», и три хода компьютера, помеченных символом «O». В заключительной позиции игры выводится текст о том, кто одержал победу.

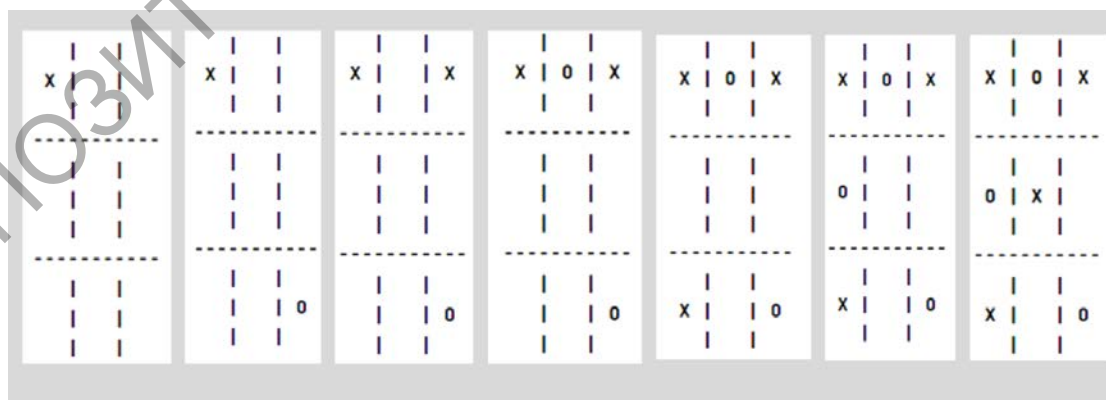


Рисунок 3 – Игра «Крестики-Нолики» с «Омикрон»

Заключение. В работе рассмотрена задача построения прототипа голосового помощника, который помогает пользователю управлять компьютером при помощи речи, отвечать на некоторые его вопросы, управлять игрой с использованием искусственного интеллекта.

Реализация игрового компонента показывает принцип работы голосового помощника, самостоятельно продумывающего и осуществляющего ходы. Разработанный готовый рабочий прототип доступен в сети Интернет по адресу https://github.com/ADolzhenok/Voice_assistant.

Интерактивные ассистенты служат не только для получения информации и общения. Они становятся всё более популярным инструментом для коммуникации с потребителями в бизнесе. Многие компании внедряют роботов в свои каналы взаимодействия с клиентами, например, в телефонную линию и чаты. Дальнейшее развитие экосистем умных девайсов приведет к появлению большого прибыльного рынка голосовых приложений, что даст толчок развитию новых бизнес-направлений для стартапов, команд программистов, интернет-компаний. Так, сеть ресторанов сможет выпустить приложение по созданию рецептов на основе базы данных миллионов ранее приготовленных блюд, которое будет руководить действиями повара-любителя прямо во время готовки. А продавцу кофе приложение посоветует, как правильно приготовить напиток.

Сегодня наметился устойчивый тренд на увеличение количества сегментов, в которых используются голосовые технологии, идея о полном отказе от общения с оператором утопична; живое общение в некоторых областях останется незаменимым, автоматизировать получится типовые запросы, а более сложные будут передаваться на обработку сотрудникам. Всё указывает на то, что в ближайшее время мы станем более тесно общаться с голосовыми помощниками в самых разных сферах. Их словарные запасы будут расти, а вместе с тем и распознаваемость запросов. Наиболее перспективным представляется использование голосовых помощников в различных рекомендательных системах, например, в медицинской диагностике или при выполнении простых медицинских операций. Внедрение голосового ассистента может значительно повысить эффективность, в частности, таких авторских рекомендательных систем, как подбор профессии [2], выбор фильма для просмотра [3]. Концепция таких систем описана в [4]. Не вызывает сомнений, что возможности голосового помощника постепенно распространятся с решения простых пользовательских требований, на решение более сложных и технически затратных задач, для реализации которых потребуются длительное обучение искусственного интеллекта.

Таким образом, продемонстрированные в статье возможности использования искусственного интеллекта позволяют мотивировать начинающих разработчиков к более глубокому овладению инструментом машинного обучения с целью разработки приложений в наиболее актуальных областях жизнедеятельности общества.

Литература

1. Больше денег – больше слов : как голосовые помощники завоевывают рынок [Электронный ресурс]. – Режим доступа : <https://rocketdata.ru/blog/voice-search-market#popup:cornerform>. – Дата доступа : 10.03.2022.
2. Осипенко, А. Н. Web-приложение для выбора сферы деятельности / А. Н. Осипенко, Н. Б. Осипенко, Ю. А. Слепенко // Проблемы физики, математики и техники. – 2018. – № 3 (36). – С. 87–94.
3. Мироненко, В. М. Автоматизация подбора персонифицированных предложений на примере задачи поиска фильма / Н. Б. Осипенко, В. М. Мироненко // Известия Гомельского гос. ун-та им. Ф. Скорины. – 2020. – № 6 (123). – С. 105–110.
4. Осипенко, Н. Б. Концепция рекомендательной системы «Подбор решений, отвечающих предназначению человека» / Н. Б. Осипенко, А. Н. Осипенко // Известия Гомельского гос. ун-та им. Ф. Скорины. – 2019. – № 6 (117). – С. 108–112.