Система управления устройствами Умного дома с использованием голосовых команд

Малых Дмитрий Андреевич Магистр 2 года,

Московский Государственный Технический Университет имени Н.Э.Баумана

Кириллова Юлия Сергеевна Магистр 2 года,

Московский Государственный Технический Университет имени Н.Э.Баумана

*В статье рассматривается метод распознавания речи на основе облачного сервиса, производится фильтрация исходных данных, производится исследование применимости данного метода в задаче распознавания голосовых команд Умного дома.*

***Ключевые слова****: Естественный интерфейс (natural user interface), распознавание речи (speech recognition, Умный дом (Smart House).*

Использование привычного для пользователя естественного способа взаимодействия человека и компьютера реализуется при помощи естественного интерфейса[1], что подразумевает использование человеком врожденных средств таких как: жесты, голос, прикосновения, взгляд. Наиболее естественным для человека средством взаимодействия является речь, поэтому именно речевой пользовательский интерфейс обладает рядом преимуществ: минимальным временем обучения пользователя использованию системы, простотой и высокой скоростью формирования запросов к системе. Такой речевой пользовательский интерфейс может быть использован в системах Умного дома для управления устройствами Умного дома.

Распознавание речи - технология, использующая естественный для человека речевой интерфейс при взаимодействии с компьютерными системами. Большинство современных методов, позволяющих осуществлять распознавание речи, требуют больших вычислительных мощностей. На сегодняшний день существует большое количество программного обеспечения, позволяющего осуществить операцию преобразования голоса в текст, но в большинстве случаев его использование просто невозможно из-за недостаточного количества памяти и вычислительных мощностей на современных автономных устройствах, которое могут быть использованы в системах Умного дома. Совершенно иной подход к решению данной задачи предлагают различные облачные сервисы распознавания речи.

Описание задачи

Задачей данной работы является выбор наиболее подходящего интернет сервиса для использования в системе управления устройствами умного дома посредством речевого интерфейса. А также тестирование работоспособности и точности распознавания речи.

В настоящее время существует множество сервисов позволяющих осуществлять задачу распознавания речи, рассмотрим наиболее популярные из них.

Данные сервисы должны обладать следующими качествами:

1. Поддержка русского языка;
2. Наличие программного интерфейса приложения(API);
3. Наличие бесплатной версии;

Также, так как данная система управления устройствами Умного дома является лишь подсистемой огромной системы Умного дома, использующей программную платформу Java[2], то наиболее предпочтительным будет выбор облачного сервиса имеющего готовый SDK[3], который можно использовать в программной платформе Java. В целом универсальность, эффективность, безопасность и совместимость с разными устройствами делают данную технологию идеальной для сетевых технологий. Платформа Java используется при разработке на персональных компьютерах, мобильных телефонах и других портативных устройствах, а также ТВ приставках, принтерах, веб камерах и других устройствах, в том числе датчиках и системах управления систем Умного дома.

Ниже представлена сравнительная таблица облачных сервисов распознавания речи.

Таблица 1

Сравнительная таблица облачных сервисов распознавания речи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название облачного сервиса распознавания речи | Поддержка русского языка | Максимальное количество запросов в сутки | Наличие бесплатной версии | Наличие SDK |
| Dragon Mobile SDK | Есть | 10000 | Есть | Есть(платформа Java) |
| Google Speech Recognition API | Есть | 500 | Есть | Есть(платформа Java) |
| Yandex Speech Kit | Есть | 10000 | Есть | Есть(платформа Java) |
| Microsoft Speech API | Есть | нет | Есть | Есть(платформа .NET) |

Как видно из таблицы (таблица 1) SDK облачного сервиса Microsoft Speech API ориентирован на платформу .NET и соответственно на использование в приложениях, написанных под операционную систему Windows. Остальные облачные сервисы имеют в наличии готовые SDK использующие платформу Java. Такие сервисы, как Dragon Mobile SDK и Yandex Speech Kit, имеют в наличии SDK, представленные на официальных сайтах разработчиков. Облачный сервис Google Speech Recognition API также имеет в наличии готовый SDK, но написанное сторонним разработчиком(JARVIS)[4]. Такой SDK имеет ряд недостатков, таких как отсутствие официальной поддержки, но также и ряд преимуществ, связанных с открытым исходным кодом, который может быть модифицирован под нужды разработчика.

Ранее были рассмотрены наиболее популярные системы распознавания речи. Dragon Mobile SDK обладает хорошей документацией, простым кодом для встраивания, поэтому выглядит очень привлекательным для использования. Тем не менее данный продукт имеет непростую систему лицензирования и чрезмерно строгие правила использования продукта. Поэтому целесообразность использования Dragon Mobile SDK находится под вопросом.

Исходя из вышесказанного, наиболее предпочтительным является использовать облачный сервис Google Speech Recognition API, который обеспечивает хорошую встраиваемость и быстродействие за счет больших вычислительных мощностей корпорации Google. Еще одним плюсом является наличие SDK для платформы Java с открытым исходным кодом. Google довольно активно и успешно развивает технологии распознавания речи, что также является неоспоримым преимуществом при использовании сервиса Google Speech Recognition API.

Исследование применимости Google Speech Recognition API

Одним из больших преимуществ использования Google Speech Recognition API является наличие фильтрации и подавления внешних шумов, на стороне облачного сервиса, которые существенно увеличивают точность распознавания. Для преобразования голосового сообщения в текст, посредствами данного сервиса, необходимо записать аудиофайл в формате моно(одноканальная запись), используя формат аудиофайлов FLAC[5] и частотой дискретизации 8кГц. После этого данный аудиофайл может быть отправлен на сервера Google для преобразования в текст. С помощью вышеуказанной библиотеки Jarvis, данная процедура записи и отправки аудиозаписи в облачный сервис может быть выполнена посредством платформы Java. Более того наиболее важной задачей данной библиотеки служит поиск начала и конца фразы в беспрерывном аудиопотоке. Стандартными средствами, включенными в библиотеку Jarvis, есть возможность реализовать простейший алгоритм поиска начала и конца фразы, то есть моменты, в которые необходимо начать запись аудиофайла и закончить запись для дальнейшей отправки в облачный сервис Google.

Алгоритм заключается в постоянном отслеживании уровня сигнала(громкости): в том случае, если уровень сигнала превысил некоторое граничное значение, то необходимо начать запись, в противном случае, если уровень громкости сигнала опустился некоторого значения, необходимо прекратить запись и отправить аудиозапись на обработку. Схематически данный алгоритм приведен ниже (рис. 1).

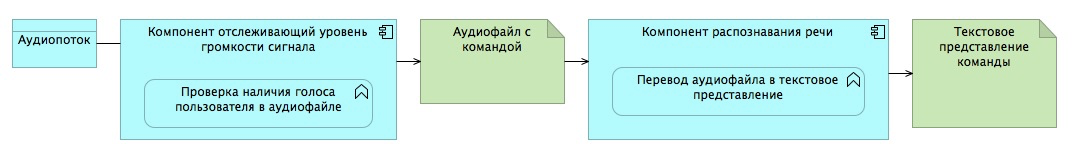


Рисунок 1 – алгоритм распознавания команды

Данный алгоритм обладает рядом недостатков, таких как ложно срабатывание при возникновении внешних шумов(музыка, работа устройств, автомобили) и голоса других людей, не являющихся пользователями системы, что может существенно ухудшить работу системы голосового управления устройствами Умного дома. Однако в качестве решения данной проблемы можно использовать полосовой фильтр, позволяющий фильтровать источники внешних шумов. Более того, благодаря открытости исходного кода используемой библиотеки, осуществить данную модификацию не составит труда. На рисунке представлен модифицированный алгоритм (рис. 2).

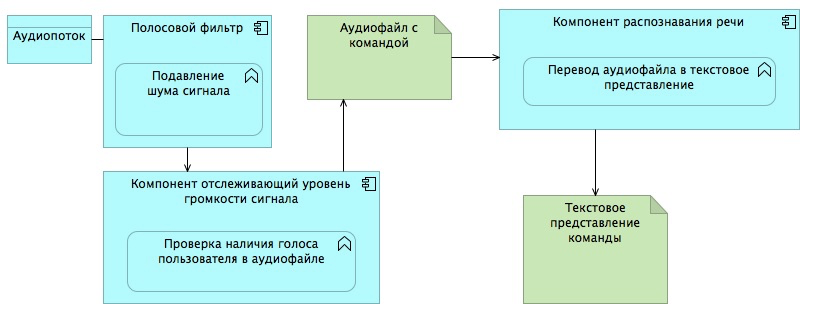


Рисунок 2 – модифицированный алгоритм распознавания команды

Для проведения исследования с помощью SDK Jarvis была создана тестовая конфигурация программного обеспечения позволяющая распознавать голосовые команды. Важной особенностью данной конфигурации является возможность тестирования работоспособности системы при помощи заранее записанных аудиофайлов, что будет использоваться при тестировании системы. На рисунке ниже представлена архитектура разработанной системы.

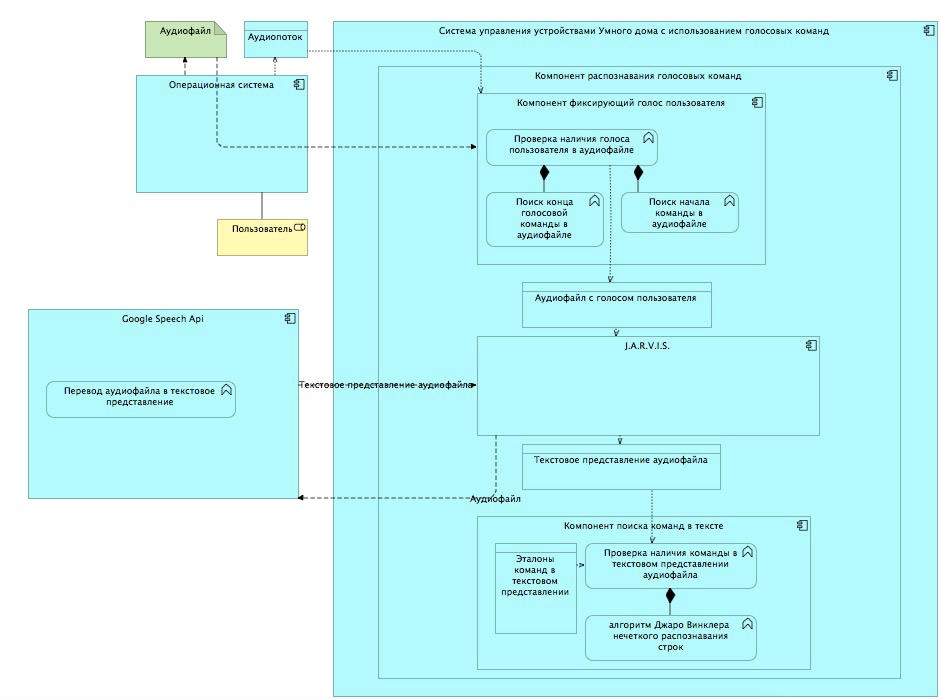


Рисунок 3 – архитектура разработанной системы

Исследование работоспособности системы

Для проведения исследования было записано 50 аудиофайлов с управляющей командой в разных условиях (дома, на улице, с фоновой музыкой, в людном месте, в метро) В каждом окружении было записано 10 аудиофайлов, в пяти разных условиях. Задача состоит в автоматическом определении команды человека для входных данных (задача классификации на 4 класса). Классы представляют собой несколько управляющих команд: начать обучение, название новой позы, фиксация позы, закончить обучение. В каждом аудиофайле содержится эталонная команда “начать обучение”, записанная следующим образом: первые и последние 5 секунд аудиофайла, есть звук окружения, остальная часть аудио файла, есть управляющая команда. Каждый аудиофайл был подан на вход тестовой конфигурации системы. Система должна была определить какая управляющая команда содержится в файле (команда “начать обучение”). Затем происходило сравнение таких параметров, как точность классификации в зависимости от условий, в которых были записаны аудиофайлы. Классификация производилась как с использованием модифицированного алгоритма, использующего полосовой фильтр, так и без него, после этого было произведено сравнение данных методов.

Тестирование системы было произведено при следующих исходных параметрах:

* Ноутбук с операционной системой Windows 10, процессором Core i5 (1,3 GHz) и 4 гигабайтами RAM, на Java VM выделено 512 МБ RAM;
* 50 аудиозаписей с командой Начать обучение в разных условиях

Результат работы программы представлен ниже.

В таблице 2 представлены результаты работы алгоритма, не использующего полосовой фильтр.

Таблица 2

Результат работы алгоритма распознавания команды без использования фильтрации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Количество команд распознанных в 10 аудиофайлах | | | | |
| Окружающая среда | начать обучение | название новой позы | фиксация позы | закончить обучение | точность распознавания |
| Дом | 9 | 0 | 0 | 0 | 90% |
| Улица | 9 | 0 | 1 | 0 | 90% |
| Фоновая музыка | 8 | 0 | 0 | 0 | 80% |
| Людное место | 7 | 0 | 2 | 0 | 70% |
| Метро | 2 | 0 | 0 | 0 | 20% |

Как видно из результатов таблицы(Таблица 2) точность распознавания эталонной команды(начать обучение), которая присутствовала во всех 50 аудиозаписях, падает при наличии внешних шумов. Наихудшая точность распознавания была продемонстрирована на аудиозаписях, записанных в метро. Такой результат можно объяснить наличием высокого уровня шума, который создает вагон метро. Данный шум обладает мощностью большей мощности полезного сигнала, поэтому определить начало и конец управляющей команды становится невозможно без применения фильтрации. Далее рассмотрим результаты работы алгоритма модифицированного полосовым фильтром (Таблица 3).

Таблица 3

Результат работы алгоритма распознавания команды с использованием фильтрации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Количество команд распознанных в 10 аудиофайлах | | | | |
| Окружающая среда | начать обучение | название новой позы | фиксация позы | закончить обучение | точность распознавания |
| Дом | 10 | 0 | 0 | 0 | 100% |
| Улица | 9 | 0 | 0 | 0 | 90% |
| Фоновая музыка | 9 | 0 | 0 | 0 | 90% |
| Людное место | 7 | 0 | 2 | 0 | 70% |
| Метро | 6 | 0 | 0 | 1 | 60% |

Как видно из результатов таблицы, расположенной выше, точность распознавания управляющей команды при всех условиях существенно улучшилась. В некоторых случаях система безошибочно распознала все управляющие команды(дома). Точность распознавания управляющих команд на аудиозаписях, записанных в метро, увеличилась до 60%. В целом, результат работы алгоритма, использующего фильтрацию, превосходит результаты работы алгоритма без фильтрации (рис. 4).

Рисунок 4 – сравнение результатов работы алгоритмов

Выводы

При анализе результатов эксперимента зависимости условий окружающей среды и точности распознавания голосовых команд, было выяснено, что в некоторых случаях точность распознавания существенно падает. Это можно объяснить тем, что исходные данные могут содержать шум, который мешает работе алгоритма. При фильтрации исходных данных влияние шума удалось существенно уменьшить.

Таким образом, использование данного метода допустимо в задаче распознавания речи.

В дальнейшем планируется модифицировать данный метод распознавания речи добавлением фонетических алгоритмов, которые позволят улучшить точность распознавания схожих по звучанию фраз.

Литература

* + - 1. Wikipedia – Natural user interface [Электронный ресурс]. Режим доступа: http://en.wikipedia.org/wiki/Natural\_user\_interface (22.03.2017).
      2. Wikipedia – Java (программная платформа) [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Java\_(программная\_платформа) (22.04.2017)
      3. Wikipedia – SDK [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/SDK (14.04.2017)
      4. Github – JARVIS [Электронный ресурс]. Режим доступа: https://github.com/lkuza2/java-speech-api (23.04.2017)
      5. Wikipedia – FLAC [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/FLAC (22.04.2017)