МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационных систем

Довженко Михаил Игоревич

Институт информационных технологий и управления в технических системах

курс 4 группа ИC/б-41(о)

09.03.02 Информационные системы (уровень бакалавриата)

ОТЧЁТ

о прохождении производственной практики

Отметка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практики

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ Шишкевич В.Е.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2016 г.

АННОТАЦИЯ

СОДЕРЖАНИЕ

1. Введение
2. Аналитический обзор современного состояния вопроса и обоснование актуальности темы
3. Основная часть, содержащая концепции и решения, сведения о методах, алгоритмах, программах (в целом и по подзадачам); используемые технологии, средства и инструменты разработки; результаты по подзадачам, результаты тестирования разработанных подсистем.
4. При разработке программного обеспечения в основную часть проекта обязательно должны входить следующие подразделы:
5. -описание математических методов, обоснование вводимых ограничений и допущений, используемых при решении задачи;
6. -описание алгоритмов и возможного взаимодействия программ с другими программами;
7. -описание и обоснование выбора метода организации входных и выходных данных;
8. -обоснование выбора языка программирования;
9. -описание программ и их применения;
10. -руководство для программиста, программиста и оператора;
11. -описание языка программирования;
12. -характеристика требуемого для реализации комплекса технических средств.
13. Допускается оформлять программную документацию в виде отдельных документов, включенных в состав пояснительной записки:

описание программы, текст программы, описание применения, руководство программиста, руководство системного программиста, описание языка, руководство оператора.

* 1. Аналитический обзор

В ходе прохождения практики был выполнен обзор систем с голосовым управлением. В ходе анализа было выяснено, что в мире существует достаточное количество таких систем. Наиболее известными и продвинутыми голосовыми помощниками на текущий момент являются iPhone's Siri, Android's Google Now и Windows 10's Cortana.  
Чем же отличается разрабатываемая система от остальных?

* Кроссплатформенность. Прежде всего, она не привязана к конкретным аппаратным устройствам или операционным системам, а может функционировать как на мобильных устройствах в виде веб-приложения, так и на устройствах под управлением Windows, Linux или OS X в виде desktop-приложения.
* Офлайн. Благодаря стеку используемых технологий система не требует доступа к интернету для выполнения ваших команд.
* Самообучаемость. То есть ассистент буквально умеет обучаться новым командам от своего владельца. Если система не поняла чего-то, то она всегда спросит вас «Что с этим делать?», а вы сможете направить ее, сказав какое из действий, приведенных в подсказке, выполнить.
* Система плагинов. Система предоставляет простое, удобное и вместе с тем, позволяющее использовать все возможности и преимущества фреймворка AngularJS, API для создания собственных голосовых функций (плагинов). Реализована система управления плагинами: система предоставляет интерфейс для доступа к списку всех доступных плагинов и автоматизирует процесс установки / обновления / удаления их по клику без каких-либо дополнительных действий.

Вы когда-нибудь задумывались о том, как работают те же Siri и Cortana? Ваш голос превращается в текст, потом происходит какая-то магия, и программа выполняет ваш запрос. Или не выполняет, если не поняла, что вы имели в виду. Система предоставляет возможность заглянуть внутрь этой «магии» и создать свои собственные голосовые функции. Причём их возможности становятся действительно безграничными.

* Русский язык. На сегодняшний день все рассмотренные системы в первую очередь ориентированы на англоязычное население, а некоторые и вовсе поддерживают только его. В связи с этим новая функциональность появляется с существенной задержкой для русскоговорящих пользователей. Таким образом лучшей альтернативой может стать стороннее решение, а именно то, что изначально создавалось для русского языка.
* Мультиязычность. Несмотря на изначальную направленность разработанной системы на русскоговорящих пользователей, алгоритм функционирования системы реализован таким образом, что система будет так же хорошо выполнять собственные голосовые функции иноязычных пользователей.
  1. Анализ методов реализации веб-ориентированных систем с голосовым управлением.

В общем случае система с голосовым управлением может быть представлена в виде совокупности следующих подсистем:

* система распознавания речи
* система классификации текста
* система извлечения ключевых слов из текста
* база знаний
* система принятия решений
* агент
  + 1. Анализ систем распознавания речи

Дипломная работа автора не предполагает реализацию собственной системы распознавания речи, так как это является очень сложной, трудоемкой и ресурсозатратной задачей, которую сложно выполнить в рамках данной работы. В связи с этим анализ представляет собой обзор уже существующих систем, с целью выбрать наиболее оптимальную для разрабатываемого продукта с возможностью использовать ее на основе лицензионного соглашения.

Были рассмотрены наиболее известные речевые системы преобразования голоса в текст, которые имеют открытое API для интеграции с разрабатываемой систему.

Прежде всего, исходя из требований к разрабатываемой системе, следует отметить основные характеристики, по которым производился выбор оптимальной аудио-системы распознавания речи:

* Поддержка русского языка
* Мультиязычность
* Доступность на всех ОС
* Ориентация на короткие фразы (3-7 слов)
* Скорость и точность обработки
* Высокие нагрузки
* Надежность и отказоустойчивость
* Ограничения на количество обращений
* Сложность интеграции

На сегодняшний день существует достаточно большой выбор среди систем распознавания речи, но лишь немногие соответствуют требованиям, представленным выше. Рассмотрим их:

1. Yandex SpeechKit

Прежде всего, взаимодействие с Yandex SpeechKit ведется через HTTP API. Из этого следует, что технология является доступной для всех платформ и операционных систем, а затраты на развертывание и поддержку собственной инфраструктуры – минимальны. Доля верно распознанных русских слов колеблется в диапазоне от 82 до 95%. Распознавание может происходить в реальном времени одновременно с передачей звуковых данных. Задержка от окончания отправки данных до получения результата распознавания не превышает 1 секунды. Инфраструктура SpeechKit спроектирована с учетом высоких нагрузок, что обеспечивает доступность и безотказную работу системы даже при большом количестве одновременных обращений. Кроме того, технология от Yandex обеспечивает простоту интерфейсов, высокую доступность и надежность. Так же, стоит отметить, что данная технология является одной из лучших для распознавания русского и турецкого языка, однако остальные языки она не поддерживает.

1. Google Cloud Speech

Google Cloud Speech, так же как и Yandex SpeechKit, предоставляет разработчикам удобный для использования web API для конвертации аудио в текст с помощью мощных моделей нейронных сетей. Однако, в отличии от технологии Яндекса, Google Cloud Speech распознает более 80 языков. Данная технология обладает так же рядом других особенностей: обработка шума и потоковая передача данных, т.е. возвращение промежуточных результатов распознавания как только они становятся доступны еще во время разговора. Минусом данной технологии является ограничение на количество запросов: до 10 000 в сутки.

1. HTML5 Speech Recognition

HTML5 Speech Recognition является одной из малоизвестных составляющих спецификации HTML5, которую сейчас поддерживают только браузеры, основанные на Webkit. Таким образом, будучи технологией, встроенной в браузеры, она не накладывает каких-либо ограничений на количество запросов в сутки и даже не требует подключения к сети Интернет. Данная технология способна распознавать как непрерывную речь, так и одиночные фразы, а так же работает с большим количеством языков, включая русский. Соответственно ее минусом является то, что не все современные браузеры ее поддерживают.

Стоит отметить, что в HTML5 спецификации есть еще одна малоизвестная технология - Speech Synthesis, которая позволяет выполнить обратную операцию: синтезировать речь, используя большое разнообразие голосов.

Таким образом, рассмотрев наиболее подходящие, для поставленной задачи, можно сделать вывод, что по своей библиотеке данных наиболее точным следует считать продукт Google Cloud Speech. Он более всего подходит для задач распознавания на базе разрабатываемой системы. Однако необходимо заметить, что данный инструментарий имеет ограничение на максимальное количество запросов в сутки, что является критическим недостатком. В то же время у технологии от Яндекса таким недостатком является поддержка слишком малого количества языком. Следовательно, в таком случае наиболее оптимальной технологией является HTML5 Speech Recognition, которая изначально интегрирована в браузер, имеет высокую точность и скорость распознавания, поддерживает множество языком, а так же не накладывает каких-либо ограничений по количеству запросов.

Система классификации текста

После того как речь была распознана, необходимо произвести интеллектуальный анализ текст, а именно – классификацию. Классификация текста - отнесение текста к одной из определенных категорий на основании его содержания.

Так как не хватает времени на разработку нейронных сетей бла бла бла…

Рассмотрим различные алгоритмы классификации текста: kNN, Naive Bayes, логическая регрессия, n-грамм, SVM, дельта TF-IDF для SVM

Модель наивного Баера показала достаточную эффективность

* 1. Разработка алгоритма функционирования веб-ориентированной системы с голосовым управлением

Если вкратце, алгоритм функционирования разрабатываемой системы выглядит так :D



Рисунок 3 – Алгоритм функционирования системы с голосовым управлением

* 1. Обоснование выбора и характеристика комплекса технических средств для реализации ИС

Для разработки серверного веб-приложения была выбрана платформа Node.js - среда выполнения JavaScript, основанная на движке V8 и предназначенная для создания масштабируемых распределённых сетевых приложений, таких как веб-сервер. Так же был использован Express.js – минималистичный и гибкий Node.js фреймворк, который обеспечивает широкий набор функций для веб-приложений.

Для реализации desktop-версии приложения был использован Electron.

Разработка десктопных приложений является достаточно сложной задачей, так как требуется заботится о таких вещах как упаковка и установка приложения, управление обновлениями, а так же о дийзане приложения и портировании его под различные операционные системы. Поддержка различных платформ обычно предполагает использование целого набора специализированных инструментов.

Electron – это фреймворк от компании GitHub, который позволяет создавать нативные кроссплатформенные приложения с использованием исключительно веб-технологий, таких как: JavaScript, HTML и CSS. Он заботится о всех сложных нюансах разработки десктопных приложений, предоставляя унифицированный интерфейс для взаимодействия с различными операционными системами и позволяя сфокусироваться на сути самого приложения. В добавок к своему собственному набору API, Electron включает в себя весь набор Chromium API, все встроенные модули Node.js и полностью поддерживает модули третьих лиц. Electron использует веб-страницы для создания пользовательского интерфейса, что снимает присущие десктопным приложениям ограничения в дизайне. В данном случае мы не ограничены системным набором элементов интерфейса нашей платформы и можем создавать нужные нам UI элементы с помощью веб-технологий. А учитывая, что данный фреймворк использует один из самых продвинутых браузеров, мы можем использовать самые новые веб-технологии. Кроме того, выбор данной технологии обусловлен так же тем, что дальнейшее портирование десктопного приложения на веб-сервер Node.js займет минимум затрат по причине того, что Electon сам использует Node.js для работы с десктопной средой.

Для реализации клиентской части как десктопного так и серверного приложения был выбран фреймворк Angular.js. AngularJS – это производительный и гибкий open source фреймворк для создания веб приложений. AngularJS предназначен для построения single-page веб приложений и является набором JavaScript функций для организации кода на стороне клиента. В основе Angular JS лежит шаблон проектирования Model View Controller, что дает ряд преимуществ при разработке и тестировании приложения. Фреймворк адаптирует и расширяет традиционный HTML, чтобы обеспечить двустороннюю привязку данных для динамического контента, что позволяет автоматически синхронизировать модель и представление. В результате AngularJS уменьшает роль DOM-манипуляций и улучшает тестируемость.

ВЫВОДЫ

В ходе прохождения практики на предприятии ООО «ОМЕГАКОД» были получены ценные навыки front-end разработки, начата работа над полноценным проектом.

Так же была выполнена часть дипломного проекта, а именно был сделан обзор веб-ориентированных систем с голосовым управлением и проведен анализ методов реализации таких систем. Так же был написал раздел анализа условий труда.