

РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ПОД ПЛАТФОРМУ ANDROID СИНХРОННОГО ПЕРЕВОДА РЕЧИ В АНИМАЦИЮ ЯЗЫКА ЖЕСТОВ

М. С. Пестрак, В. А. Ульяницкий

*Белорусский государственный университет, г. Минск;
fpm.pestrak@bsu.by, fpm.ulyanick@bsu.by;
науч. рук. – А. С. Гусейнова, ст. преп.*

Речь - важнейший инструмент коммуникации между людьми. Но далеко не всем людям доступно это средство общения. В связи с этим, было решено реализовать приложение, осуществляющее перевод голосовых команд в язык жестов. Цель мобильного приложения для синхронного перевода – усовершенствовать средства коммуникации для людей с особенностями слуха и речи. Важной частью работы является модуль распознавания речи, так как он должен быть достаточно быстрым и точным. Было проведено изучение существующих библиотек, позволяющих распознавать голосовые команды, а также сравнение алгоритмов, которые в них используются.

В результате проведенной работы был создан тестовый прототип приложения, реализован модуль распознавания речи, усовершенствована акустическая модель.

Ключевые слова: распознавание речи; жестовый перевод; социальная среда; android; CMU Sphinx.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ РАСПОЗНАВАНИЯ РЕЧИ

Модуль распознавания речи используется для перевода устной речи пользователя в текст, который далее передаётся в клиентское приложение. Подавляющее большинство подобных систем работают онлайн: речь передаётся на удалённый сервер, где и происходит распознавание. Такой вариант не подходит для данного приложения, поэтому был сделан обзор среди систем, имеющих возможность работы офлайн. Также, для большего удобства интеграции в приложение, выбор производился среди систем, распространяемых с open-source лицензиями.

Таким образом, были рассмотрены следующие системы: CMU Sphinx, НТК, iAtros, Julius, Kaldi и RWTH ASR. Выбор системы зависел от таких критериев, как точность и скорость распознавания, удобство использования в программном коде и внутренняя структура. По точности системы оценивались исходя из наиболее популярных метрик: коэффициент распознавания слов (WRR), коэффициент ошибок в словах (WER). Скорость работы системы оценивалась с помощью коэффициента Speed Factor (SF) — показателя отношения времени распознавания к длительности распознаваемого сигнала, иногда называемого также Real Time Factor [1].

Таблица 1

Сравнение систем распознавания речи

Система	WER, %	WRR, %	SF
HTK	19,8	80,2	1,4
CMU Sphinx (pocketsphinx/sphinx4)	21.4/22.7	78.6/77.3	0,5/1
Kaldi	6,5	93,5	0,6
Julius	23,1	76,9	1,3
iAtros	16,1	83,9	2,1
RWTH ASR	15,5	84,5	3,8

Исходя из вышеизложенного, система CMU Sphinx была выбрана для использования в проекте. CMU Sphinx включает в себя два декодера - pocketsphinx, реализованный на C, и sphinx4, реализованный на Java. Это позволяет использовать эту систему на многих платформах, включая те, на которых установлена операционная система Android, что значительно облегчает ее интеграцию в проекты, написанные на Java. Модульность реализации системы CMU Sphinx значительно упрощает работу с ней, позволяя легко вносить в неё изменения и исправлять ошибки.

ПРИНЦИПЫ РАБОТЫ АКУСТИЧЕСКОЙ МОДЕЛИ

В процессе разработки приложения выявилась необходимость увеличения шумоустойчивости модуля распознавания речи. Рассмотрим более подробно, как реализовано распознавание речи в системе CMU Sphinx [2]. CMU Sphinx использует для извлечения частотных признаков мел-частотные кепстральные коэффициенты (mel-frequency cepstral coefficients), точнее их модификацию – нормированные по мощности частотные кепстральные коэффициенты (power-normalized cepstral coefficients), которые достаточно шумоустойчивы. Синтез речи можно разделить на две стадии: сначала акустическая волна генерируется в дыхательной системе (лёгкие, бронхи, трахея), а затем преобразуется в речевом тракте. Таким образом, процесс можно представить как последовательную генерацию основного тона и его фильтрацию. При этом основной тон является дикторозависимым, а коэффициенты фильтра зависят от параметров голосового тракта, при этом именно эти параметры определяют произносимые фонемы и, соответственно, важны для распознавания. Итоговый сигнал можно представить в виде формулы (1).

$$F(\omega) = S(\omega)H(\omega) \quad (1)$$

Для выделения характеристик фильтра, не пересекающихся с характеристиками основного фона, выполняются преобразования.

Итоговая функция – кепстр – функция обратного преобразования Фурье от логарифма спектра мощности сигнала представлена формулой (2). Для вычисления коэффициентов сигнал разбивается на отрывки-фреймы длиной 23 мс.

$$C_s(q) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \ln|S^2(\omega)| e^{i\omega q} d\omega \quad (2)$$

На следующем этапе – акустическом моделировании – используются скрытые марковские модели. При этом вычисленные на предыдущем шаге коэффициенты сопоставляются с фонемами языка. Скрытые марковские модели описывают двухэтапный стохастический процесс. Первый этап – последовательность фонем рассматривают как марковскую цепь. На следующем этапе для каждого фрейма рассматривается наблюдаемый результат, которым в нашем случае является последовательность коэффициентов, вычисленных для фреймов. Для вычисления наиболее вероятного значения скрытой последовательности в CMU Sphinx используется алгоритм Витерби – алгоритм динамического программирования для поиска наиболее вероятной скрытой цепи Маркова в скрытой марковской модели; этот список состояний называется путём Витерби.

На этапе языкового моделирования необходимо на основе фонем определить произнесённые слова. Задачей языковой модели является определение вероятности произнесения слов и различных словосочетаний [4].

В нашей работе было принято решение использовать грамматику речи Java ввиду того, что ценность для распознавания представляют только фразы, для которых существует представление в виде анимации, остальные всё равно будут проигнорированы, а быстроедействие, наоборот, весьма значимо.

РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

В качестве платформы, которая использовалась для разработки приложения, выбор был сделан в пользу Android, так как он является одним из самых распространенных и доступных. Android является более простым для начинающих разработчиков, и позволяет использовать Java в качестве языка программирования. В связи с этим выбор технологий стал следующим: клиентское приложение реализовано на Android с использованием SQLite для создания базы данных для хранения целевого словаря, для распознавания речи была выбрана библиотека CMU Sphinx, как самая доступная для новичков, также она позволяет использовать Java в качестве языка программирования.

База данных приложения состоит из набора команд, который был составлен инструктором автошколы и анимированных изображений.

Приложение будет работать offline, так как это позволит увеличить скорость получения результата. Для работы пользователю необходимо будет загрузить целевые словари, которые будут храниться локально на его устройстве, в качестве плагинов.

Для хранения словарей необходима база данных – набор таблиц, связанных определёнными отношениями. В качестве базы данных для проекта было решено использовать реляционную базу данных SQLite [3].

Для хранения словаря, доступного пользователям, понадобится следующая таблица, которая имеет поля: ID слова или выражения – сгенерированный искусственный номер лексической единицы, непосредственно сама лексическая единица и путь к анимации, которая хранится в локальных ресурсах приложения.

В физическом смысле приложение после сборки представляет собой файл с расширением *.apk для установки основной части приложения на устройство пользователя.

Приложение позволяет пользователю осуществлять перевод голосовых сообщений в жестовую интерпретацию, дополнительно выводится текстовый перевод. Пользователь может просматривать весь целевой словарь.

Условно можно разделить части интерфейса на следующие части:

- интерфейс главного экрана;
- интерфейс словаря.

Интерфейс главного экрана - интерфейс, предоставленный пользователю для основной работы по осуществлению перевода. Служит для настройки приложения, осуществления голосового ввода, по нажатию кнопки с изображением микрофона и просмотра жестового перевода. Такой интерфейс включает в себя меню с тремя кнопками для навигации между фрагментами окна: перевод, настройки и словарь.

Библиографические ссылки

1. *Беленко, М. В.*, Сравнительный анализ систем распознавания речи с открытым исходным кодом / *М. В. Беленко, П. В. Балакишин* // Международный научно-исследовательский журнал [Электронный ресурс]. – 2017. - Режим доступа: <https://research-journal.org/technical/sravnitelnyj-analiz-sistem-raspoznavaniya-rechi-s-otkryтым-kodom/> - Дата доступа: 02.10.2019
2. CMU Sphinx [Электронный ресурс] / Carnegie Mellon University. – 2019. – Режим доступа: <https://cmusphinx.github.io/>. – Дата доступа: 25.01.2020.
3. Android – Программирование для профессионалов / *Б. Харди* [и др.]; под общ. ред. *Б. Харди*. — 2-е изд. — СПб.: Питер, 2016. – 640 с.
4. *Гриф, М. Г.*, Система перевода текста с учетом синтаксической составляющей русского жестового языка / *М. Г. Гриф, Ю. С. Мануева* // CYBERLENINKA [Электронный ресурс]. – 2018. - Режим доступа: <https://cyberleninka.ru/article/n/sistema-perevoda-teksta-s-uchetom-sintaksicheskoy-sostavlyayuschey-russkogo-zhestovogo-yazyka> - Дата доступа: 28.09.2019