ПОЛЯНИЧКО К. С.

ОБЗОР И ТЕСТИРОВАНИЕ СИСТЕМ РАСПОЗНАВАНИЯ РЕЧИ

Аннотация. В данной статье описываются и сравниваются характеристики, таких наиболее распространённых систем по распознаванию речи, как CMU Sphinx, Google Speech, HTK, Julius, iAtros, RWTH ASR. Сравнение происходит по следующим характеристикам: поддерживаемые языки распознавания, поддерживаемые операционные системы, наличие документации, открытость кода, наличие или отсутствие офлайн режима, показатели WER, WRR, SF. Также описан процесс и результаты тестирования наилучших по различным показателям систем.

Ключевые слова: распознавание речи, CMU Sphinx, Google Speech, HTK, Julius, iAtros, RWTH ASR, точность распознавания, скорость обработки речи.

POLYANICHKO K. S.

OVERVIEW AND TESTING OF SPEECH RECOGNITION SYSTEMS

Abstract. This article describes and compares the characteristics of such most common speech recognition systems as CMU Sphinx, Google Speech, HTK, Julius, iAtros, RWTH ASR. The comparison is based on the following important characteristics: supported recognition languages, supported operating systems, availability of documentation, open source code, presence or absence of offline mode, WER, WRR, SF indicators. It also describes the process and results of testing the best systems for various indicators.

Keywords: speech recognition, CMU Sphinx, Google Speech, HTK, Julius, iAtros, RWTH ASR, recognition accuracy, speech processing speed.

Системы распознавания речи чаще всего применяются для воссоздания более удобного взаимодействия человека с техникой, например, для голосового управления. На сегодняшний день распознавание речевых сигналов обладает очень обширным спектром применения, как пример, голосовые помощники, которые могут использоваться в современных телефонах, автомобилях и т.д. [1]. В связи с актуальностью и широким распространением систем распознавания, была поставлена задача определить наиболее эффективную программу по распознаванию речи из выбранных с целью дальнейшего применения в робототехнической лабораторной платформе с техническим зрением, предназначенной для обучения робототехнике, а также участия в робототехнических соревнованиях.

Основываясь на некоторые данные из работы Беленко М. В. и Балакшина П. В. «Сравнительный анализ систем распознавания речи с открытым кодом» были рассмотрены существующие системы распознавания речи, такие как CMU Sphinx, HTK, iAtros, Julius, Kaldi и RWTH ASR, Google. Выбор именно этих систем обусловлен частотой упоминания в научно-исследовательских журналах и популярности среди индивидуальных разработчиков программного обеспечения [2]. В отличии от выше указанной работы рассматривались программы не только с открытым кодом, а также подробно были описаны этапы тестирования программ и результаты испытаний.

При анализе систем были рассмотрены следующие показатели: поддерживаемые языки распознавания, поддерживаемые операционные системы, наличие документации, открытость кода, наличие или отсутствие офлайн режима, показатели WER, WRR, SF.

WER – это точность распознавания, которая является показателем качества и определяется как процент неправильно распознанных слов (Word Error Rate), а показатель WRR (Word Recognition Rate) наоборот отражает процент правильно распознанных слов. Второй важный критерий распознавания речи, который был рассмотрен – скорость обработки речи, выраженный в показатели скорости SF (Speed Factor) [3].

В таблице 1 представлены описанные характеристики различных систем по распознаванию речи [2].

Таблица 1

Обзор систем по распознаванию речи

Система	**Поддерживаемые языки распознавания**	**Поддерживаемые ОС**	**Наличие документации**	**Открытость кода**	**Офлайн режим**	**Показатели: WER(%), WRR(%), SF**
CMU Sphinx (pocketsphinx)	Множество языков, в том числе экзотические	Linux, Mac OS, Windows, Android	Подробная онлайн документация	+	+	21.4/22.7, 78.6/77.3, 0.5/1
HTK	Английский	Linux, Solaris, HPUX, IRIX, Mac OS, FreeBSD, Windows	HTK Book – исчерпывающая информация	+	-	19,8, 80,2, 1.4

Продолжение таблицы 1

Система	**Поддерживаемые языки распознавания**	**Поддерживаемые ОС**	**Наличие документации**	**Открытость кода**	**Офлайн режим**	**Показатели: WER(%), WRR(%), SF**
Julius	Японский, Английский	Linux, Windows, FreeBSD, Mac OS	Julius Book – аналогично HTK Book	+	-	16.1, 83.9, 2.1
iAtros	Английский, Испанский	Linux	Отсутствие документации	+	-	16.1, 83.9, 2.1
RWTH ASR	Английский	Linux, Mac OS	Неподробная документация	+	-	15.5, 84.5, 3.8
Google	Множество языков	Linux, Mac OS, Windows, Android	Онлайн документация	-	-	4.9, 95.1, 0.5

Из перечисленных вариантов для тестирования было решено выбрать системы SMU Sphinx и Google. Данное решение было сделано в связи с тем, что CMU Sphinx единственная из перечисленных может работать в офлайн режиме, а также, не смотря на не очень высокую точность распознавания, обладает одной из лучших скоростей распознавания из всех указанных, большим количеством поддерживаемых языков, распространяется под лицензией BSD, которая разрешает встраивание в коммерческие проекты, а также возможностью внесения изменений в открытый код. Выбор второй системы был обусловлен наилучшими показателями WER и WRR. И одним из важнейших критериев была поддерживаемая ОС, это ОС Windows.

Далее было проведено тестирование систем, которое осуществлялось с применением языка python и выбранных ранее в обзоре систем: PocketSphinx API и Google Speech API и библиотек SpeechRecognition и Pyaudio.

Первый тест по распознаванию текста из аудио файла был сделан с использованием Google Speech API и библиотек SpeechRecognition.

Сам аудио файл содержал следующий текст: «Who’s the low to the left shoulder take the winding path reach the lake no closely the size of the gas tank wipe degrees off is dirty face men to call

before you go out the Redwood valley strain and hung limp the stray cat gave birth to kittens the young girl gave no clear response the meal was cooked before the bell rang what Joy there is a living». Данный аудио файл взят с интернет-ресурса и текст, записанный в нем произносится диктором, для которого английский язык является родным, запись сделана в хорошем качестве без посторонних шумов.

Настройка и тестирование первой системы осуществлялась по следующему алгоритму:

1. Установка библиотеки SpeechRecognition в командной строке компьютера.
2. Вызов python в командной строке компьютера.
3. Импорт только, что загруженной библиотеки.
4. Создание экземпляра класса Recognizer.
5. Создание объекта класса AudioFile модуля speech_recognition.
6. Преобразование аудиофайла в объект AudioData методом record() класса Recognizer. Передача объекта AudioFile методу record().
7. Проверка типа переменной.
8. Передача объекта audio_content методу recognize_google() объекта класса Recognizer(), и аудиофайл будет преобразован в текст. Полный текст программы представлен на рисунке 1.

main.py

```
1
2   import speech_recognition as speech_recog
3   recog = speech_recog.Recognizer()
4   sample_audio = speech_recog.AudioFile('D:/001.wav')
5   with sample_audio as audio_file:
6   audio_content = recog.record(audio_file)
7   type(audio_content)
8   recog.recognize_google(audio_content)
```

Рис. 1. Программа по настройке и запуску распознавания речи из аудио файла с помощью Google Speech API.

Пример корректной работы программы с распознаванием из аудио файла с применением Google Speech API с полученным текстом представлен на рисунке 2.

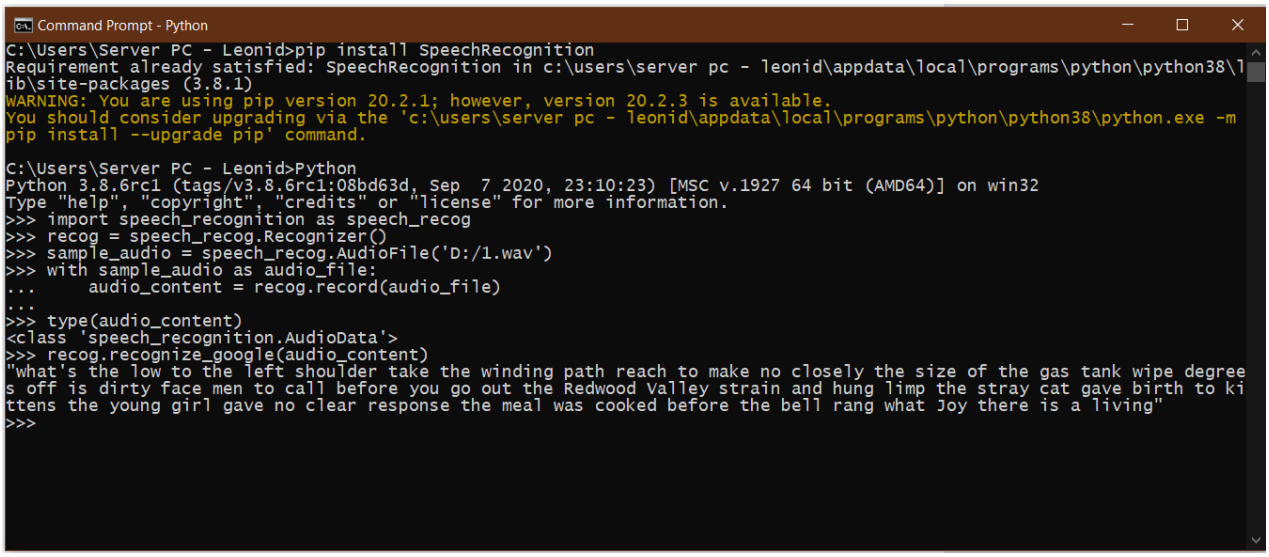

Рис. 2. Пример корректной работы программы с распознаванием из аудио файла с применением Google Speech API.

Второй тест по распознаванию текста из аудио файла был сделан с использованием PocketSphinx API и библиотек SpeechRecognition. Настройка и тестирование первой системы осуществлялась по прежнему алгоритму из приложения Б, только необходимо заменить метод

recognize_google() на метод recognize_sphinx(). PocketSphinx API может работать без подключения к сети интернет.

Пример корректной работы программы с полученным текстом представлен на рисунке 3.

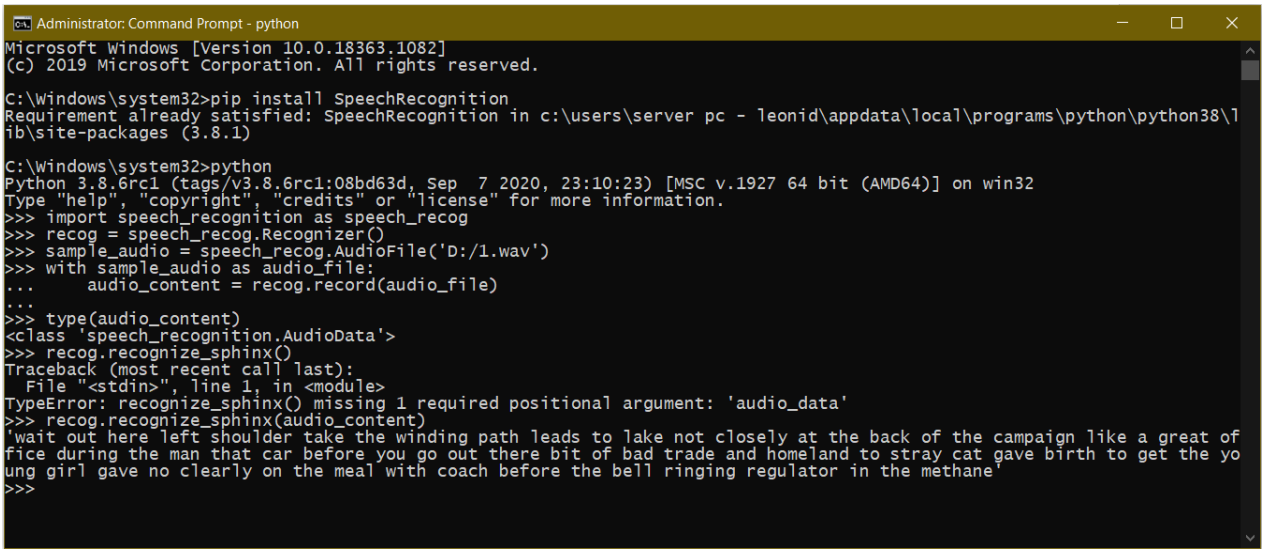

Рис. 3. Пример корректной работы программы с распознаванием из аудио файла с применением PocketSphinx API.

Результаты проверки программ показали следующие результаты:

1.Система по распознаванию Google из записанных в аудио файле 71 слова правильно распознала 68 слов.

2. Система по распознаванию PocketSphinx из записанных в аудио файле 71 слова правильно распознала 28 слов.

Для третьего теста, то есть распознавания речи с живого микрофона с использованием Google Speech API и PocketSphinx API, а также библиотеки Pyaudio была установлена VisualStudio – это линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств [4]. Распознавание речи с живого микрофона и перевод ее в текст осуществлялся последующему алгоритму программы, показанном на рисунке 4.

main.py

```
import speech_recognition as speech_recog
import pyaudio
import pocketsphinx

recog = speech_recog.Recognizer()
mic = speech_recog.Microphone()
with mic as audio_file:
    print("Speak Please")

    recog.adjust_for_ambient_noise(audio_file)
    audio = recog.listen(audio_file)

    print("Converting Speech to Text...")
    print("You said with google: " + recog.recognize_google(audio))
    print("You said with sphinx: " + recog.recognize_sphinx(audio))
```

Рис. 4. Программа по настройке и запуску распознавания речи с микрофона с помощью Google и PocketSphinx.

Программа задействует сразу две библиотеки Google Speech API и PocketSphinx API и выводит результаты в двух разных строках. Пример корректной работы программы с полученным текстом, распознанным с микрофона представлен на рисунке 5, где произносилось: «What’s going on». Необходимо отметить, что текст произносился в условиях отсутствия посторонних шумов и с использованием довольно качественного конденсаторного микрофона.

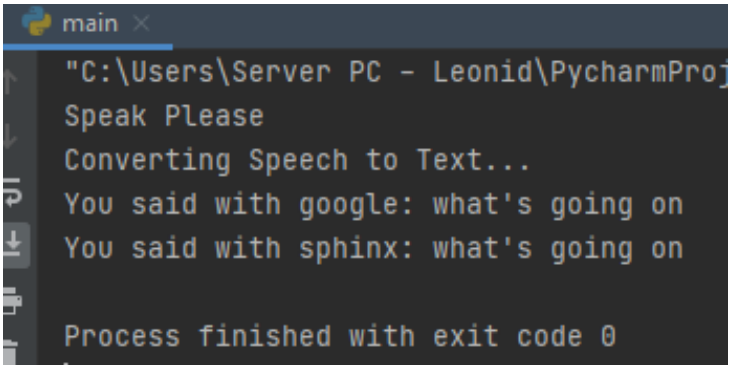

Рис. 5. Пример работы программы с распознаванием короткой фразы с микрофона.

Следующий пример корректной работы программы с полученным текстом, распознанным с микрофона представлен на рисунке 6, где произносилось: «Who's the low to the left shoulder take the winding path reach the lake no closely the size of the gas». Также следует отметить, что текст произносился в условиях отсутствия посторонних шумов, с использованием довольно качественного конденсаторного микрофона и со скоростью примерно 45 слов в минуту и не носителем английского языка.

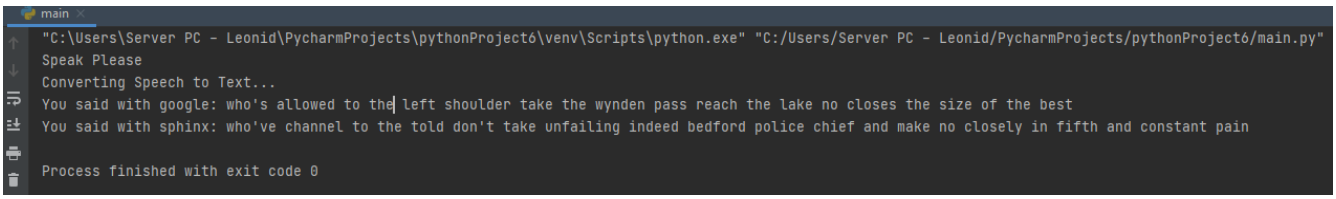

Рис. 6. Пример работы программы с распознаванием длинной фразы с микрофона.

По результату третьего теста было выявлено, что Google практически всегда правильно распознает короткие фразы, в то время, как PocketSphinx делает это с переменным успехом, для того, чтобы он корректно распознал фразу требуется очень четкого и правильного произношения. В случае с длинными фразами Google из произнесенных 23 слов корректно распознал 16 слов, а PocketSphinx всего 6 слов. Также следует отметить, что на качество распознавания влияет множество факторов и в связи с этим достаточно сложно выявить определенную зависимость.

Делая выводы, Google Speech API по качеству распознавания показал намного лучшие результаты, что и подтверждается параметрами, указанными в таблице, но в некоторых случаях и проектах, где может быть важна автономность и возможность применения в коммерческих

проектах, также может быть использован PocketSphinx API, который к тому же имеет возможность усовершенствования за счёт открытости кода.

СПИСОК ЛИТЕРАТУРЫ

1. Бабаринов С. Л., Будникова М. А. О распознавании речи // Научные ведомости Белгородского государственного университета. Серия История. Политология. Экономика. Информатика. – 2014. – № 21(192), выпуск 32/1. - С. 182-185.
2. Беленко М. В., Балакшин П. В. Сравнительный анализ систем распознавания речи с открытым кодом // Международный научно-исследовательский журнал. – 2017. – № 4(58), Часть 4. – С. 13–18 [Электронный ресурс]. – Режим доступа: https://researchjournal.org/technical/sravnitelnyj-analiz-sistem-raspoznavaniyarechi-s-otkrytym-kodom/ (дата обращения: 14.01.2021).
3. Карпов А. А., Кипяткова И. С. Методология оценивания работы систем автоматического распознавания речи // Известия высших учебных заведений. Приборостроение. – 2012. – Т. 55, №. 11. – С. 38-43.
4. Visual Studio [Электронный ресурс] // Официальная страница Visual Studio компании Microsoft. – Режим доступа: https://visualstudio.microsoft.com/ (дата обращения: 14.01.2021).