

 tutam 17 марта 2018 в 22:34

# Распознавание речи на python с помощью pocketsphinx или как я пытался сделать голосового ассистента

Python, Звук

Tutorial

Это tutorial по использованию библиотеки pocketsphinx на Python. Надеюсь он поможет вам побыстрее разобраться с этой библиотекой и не наступать на мои грабли.

Началось все с того, что захотел я сделать себе голосового ассистента на python. Изначально для распознавания решено было использовать библиотеку speech\_recognition. Как оказалось, я не один такой. Для распознавания я использовал Google Speech Recognition, так как он единственный не требовал никаких ключей, паролей и т.д. Для синтеза речи был взят gTTS. В общем получился почти клон этого ассистента за чего я не мог успокоиться.

Правда, успокоиться я не мог не только из-за этого: ответа приходилось ждать долго (запись заканчивалась не сразу, отправка речи на сервер для распознавания и текста для синтеза занимала немало времени), речь не всегда распознавалась правильно, дальше полуметра от микрофона приходилось кричать, говорить нужно было четко, синтезированная гуглом речь звучала ужасно, не было активационной фразы, то есть звуки постоянно записывались и передавались на сервер.

Первым усовершенствованием был синтез речи при помощи yandex speechkit cloud:

```
URL = 'https://tts.voicetech.yandex.net/generate?text='+text+'&format=wav&lang=ru-RU&speaker=ermil&key='+key+'&speed=1&emcn=good'
response=requests.get(URL)
if response.status_code==200:
    with open(speech_file_name,'wb') as file:
        file.write(response.content)
```

Затем настала очередь распознавания. Меня сразу заинтересовала надпись "CMU Sphinx (works offline)" на странице библиотеки. Я не буду рассказывать об основных понятиях pocketsphinx, т.к. до меня это сделал @chubakur(за что ему большое спасибо) в этом посте.

## Установка Pocketsphinx

Сразу скажу, так просто pocketsphinx установить не получится(по крайней мере у меня не получилось), поэтому ~~pip install pocketsphinx~~ не работает, упадет с ошибкой, будет ругаться на wheel. Установка через pip будет работать только если у вас стоит swig. В противном случае чтобы установить pocketsphinx нужно перейти [вот сюда](#) и скачать установщик(msi). **Обратите внимание: установщик есть только для версии 3.5!**

## Распознавание речи при помощи pocketsphinx

Pocketsphinx может распознавать речь как с микрофона, так и из файла. Также он может искать горячие фразы(у меня не очень получилось почему-то код, который должен выполняться когда находится горячее слово выполняется несколько раз, хотя произносил его я только один раз). Облачные решения pocketsphinx отличается тем, что работает оффлайн и может работать по ограниченному словарю, вследствие чего повышается точность. Если интересно, на [странице библиотеки](#) есть примеры. Обратите внимание на пункт "Default config".

## Русская языковая и акустическая модель

Изначально pocketsphinx идет с английской языковой и акустической моделями и словарем. Скачать русские можно по этой ссылке. Архив распаковать. Затем надо папку <your\_folder>/zero\_ru\_cont\_8k\_v3/zero\_ru.cd\_cont\_4000 переместить в папку C:/Users/tutam/AppData/Local/Programs/Python/Python35-32/Lib/site-packages/pocketsphinx/model, где <your\_folder> это папка в которую распаковали архив. Перемещенная папка — это акустическая модель. Такую же процедуру надо проделать с файлами ru.lm и ru.dic и <your\_folder>/zero\_ru\_cont\_8k\_v3/. Файл ru.lm это языковая модель, а ru.dic это словарь. Если вы все сделали правильно, то следующий код должен работать.

```
import os
from pocketsphinx import LiveSpeech, get_model_path
```

```
model_path = get_model_path()

speech = LiveSpeech(
    verbose=False,
    sampling_rate=16000,
    buffer_size=2048,
    no_search=False,
    full_utt=False,
    hmm=os.path.join(model_path, 'zero_ru.cd_cont_4000'),
    lm=os.path.join(model_path, 'ru.lm'),
    dic=os.path.join(model_path, 'ru.dic')
)

print("Say something!")

for phrase in speech:
    print(phrase)
```

Предварительно проверьте чтобы микрофон был подключен и работал. Если долго не появляется надпись `Say something!` — это нормально. Большую часть этого времени занимает создание экземпляра `LiveSpeech`, который создается так долго потому, что русская языковая модель весит более 500(!) мб. У меня экземпляр `LiveSpeech` создается около 2 минут.

Этот код должен распознавать почти любые произнесенные вами фразы. Согласитесь, точность отвратительная. Но это можно исправить. Увеличить скорость создания `LiveSpeech` тоже можно.

## JSGF

Вместо языковой модели можно заставить `rocketsphinx` работать по упрощенной грамматике. Для этого используется `jsgf` файл. Его использование ускоряет создание экземпляра `LiveSpeech`. О том как создавать файлы грамматики написано [здесь](#). Если языковая модель и `jsgf` файл будет игнорироваться, поэтому если вы хотите использовать собственный файл грамматики, то нужно писать так:

```
speech = LiveSpeech(
    verbose=False,
    sampling_rate=16000,
    buffer_size=2048,
    no_search=False,
    full_utt=False,
    hmm=os.path.join(model_path, 'zero_ru.cd_cont_4000'),
    lm=False,
    jsgf=os.path.join(model_path, 'grammar.jsgf'),
    dic=os.path.join(model_path, 'ru.dic')
)
```

Естественно файл с грамматикой надо создать в папке `C:/Users/tutam/AppData/Local/Programs/Python/Python35-32/Lib/site-packages/pocketsphinx/model`. И еще: при использовании `jsgf` придется четче говорить и разделять слова.

## Создаем свой словарь

Словарь — это набор слов и их транскрипций, чем он меньше, тем выше точность распознавания. Для создания словаря с русскими словами нужно воспользоваться проектом `ru4sphinx`. Качаем, распаковываем. Затем открываем блокнот и пишем слова, которые должны быть в словаре, каждое с новой строки, затем сохраняем файл как `my_dictionary.txt` в папке `text2dict`, в кодировке **UTF-8**. Затем открываем консоль и пишем: `C:\Users\tutam\Downloads\ru4sphinx-master\ru4sphinx-master\text2dict> perl dict2transcript.pl my_dictionary.txt my_dictionary_out.txt`. Открываем `my_dictionary_out.txt`, копируем содержимое. Открываем блокнот, вставляем скопированный текст и сохраняем файл как `my_dict.dic` (вместо "текстовый файл" выберите "все файлы"), в кодировке **UTF-8**.

```
speech = LiveSpeech(
    verbose=False,
    sampling_rate=16000,
    buffer_size=2048,
    no_search=False,
    full_utt=False,
    hmm=os.path.join(model_path, 'zero_ru.cd_cont_4000'),
    lm=os.path.join(model_path, 'ru.lm'),
    dic=os.path.join(model_path, 'my_dict.dic')
)
```

Некоторые транскрипции может быть нужно подправить.

## Использование pocketsphinx через speech\_recognition

Использовать pocketsphinx через speech\_recognition имеет смысл только если вы распознаете английскую речь. В speech\_recognition нельзя указать пустую языковую модель и использовать jsf, а следовательно для распознавания каждого фрагмента придется ждать 2 минуты. Проверенно.

### Итог

Угробив несколько вечеров я понял, что потратил время впустую. В словаре из двух слов(да и нет) сфинкс умудряется ошибаться, причем Отъедает 30-40% celeron'a, а с языковой моделью еще и жирный кусок памяти. А Яндекс почти любую речь распознает безошибочно, при этом есть память и процессор. Так что думайте сами, стоит ли за это браться вообще.

**P.S.:** это мой первый пост, так что жду советы по оформлению и содержанию статьи.

Только зарегистрированные пользователи могут участвовать в опросе. Войдите, пожалуйста.

Какое решение для распознавания речи вам нравится больше?

- ☐ sphinx
- ☐ Yandex Speechkit Cloud
- ☐ Google Cloud Speech API
- ☐ Свой вариант

Проголосовали 67 пользователей. Воздержались 53 пользователя.

**Теги:** python 3, cmu sphinx, pocketsphinx, распознавание речи, работа со звуком, speech\_recognition

↑ +13 ↓ 127 👁 18,1k 💬 13



11,0

Карма

-0,1

Рейтинг

1

Подписчики

@tutam

Пользователь

Поделиться публикацией

### ПОХОЖИЕ ПУБЛИКАЦИИ

29 марта 2017 в 01:50

Новый рекорд в распознавании речи: уровень ошибок алгоритма снижен до 5,5%

↑ +13 👁 8,6k 📖 33 💬 4

23 сентября 2015 в 13:39

Распознавание речи с помощью CMU Sphinx

↑ +18 👁 48,1k 📖 196 💬 2

25 февраля 2013 в 20:12

## Pocketsphinx. Распознавание речи и голосовое управление в Linux

↑ +54

👁 98,1k

📖 481

💬 37

## ВАКАНСИИ

Мой к



Python-разработчик

от 140000 до 1800

Онлайн-кинотеатр ivi · Москва



Ищем perl/python разработчика

от 20000 до 600

karasiki · Казань · Возможна удаленная работа



Программист Python

от 10000 до 200

Хантфлоу · Махачкала · Возможна удаленная работа

áxmit

Python developer

от 60000 до 1200

áxmit · Петрозаводск · Возможна удаленная работа



Python Team leader

от 30

GetPure Inc · Возможна удаленная работа

Все вакансии

## Комментарии 13



mbait

18 марта 2018 в 01:42



Сразу скажу, так просто rocketsphinx установить не получится(по крайней мере у меня не получилось), поэтому `pip install pocketsphinx` не сработает, упадет с ошибкой, будет ругаться на wheel.

`pip install pocketsphinx` отлично работает. Скорее всего, у вас не установлен `swig`, который нужен для сборки.

Также он может искать горячие фразы(у меня не очень получилось, почему-то код, который должен выполняться когда находится горячее слово выполняется несколько раз, хотя произносил его я только один).

Потому что надо документацию читать.

В словаре из двух слов(да и нет) сфинкс умудряется ошибаться, причем часто.

Потому что для данной акустической модели нужно было выставить частоту дискретизации 8000, а не 16000.

Отъедает 30-40% celeron'a, а с языковой моделью еще и жирный кусок памяти. А Яндекс почти любую речь распознает безошибочно, при том не е память и процессор.

Сравнение тёплого с мягким. Если ПО для распознавания Яндекса запустить на вашей машине, то она, вероятно, вообще заглохнет. Хотите попробовать — Kaldi



tutam

18 марта 2018 в 22:31



`pip install pocketsphinx` отлично работает. Скорее всего, у вас не установлен `swig`, который нужен для сборки.

Насчет `swig` я знал, решил не заморачиваться с его установкой.

Потому что надо документацию читать.

Можете ткнуть носом, а то я никак понять не могу в чем дело?

Потому что для данной акустической модели нужно было выставить частоту дискретизации 8000, а не 16000.

Спасибо. Точность увеличилась, правда говорить приходится громче.

Сравнение тёплого с мягким. Если ПО для распознавания Яндексa запустить на вашей машине, то она, вероятно, вообще загнётся. Хотите попробовать? — Kaldi

Загнется, не сомневаюсь. Но Яндекс распознает на сервере, так что в итоге по потреблению ресурсов он эффективнее(если нет проблем с интернетом). И при своем огромном словаре он распознает эти несколько команд точнее чем sphinx, у которого словарь только из команд и состоит.

 Delneg 18 марта 2018 в 08:04   


Попробуйте [github.com/kalliope-project/kalliope](https://github.com/kalliope-project/kalliope)

 kedobear  18 марта 2018 в 09:54   

Вот вам до кучи:

Rocketsphinx. Распознавание речи и голосовое управление в Linux

У меня с двумя словами 5 лет назад работало хорошо.

 DollaR84 18 марта 2018 в 22:06   

Я тоже написал себе голосового помощника на rocketsphinx.

Через pip устанавливается прекрасно.

Так как я не задавался целью распознавания произвольной речи, поэтому сразу отбросил вариант использования языковой модели.

Использовал исключительно набор грамматических правил JSFG и составил свой ограниченный словарь.

Данная схема позволила получить в итоге:


быструю загрузку;

очень высокую точность распознавания, по-моему. Ошибки бывают, но не очень часто.

И это абсолютно без подстройки к голосу.

Ну и самое главное конечно оффлайн.

Так что считаю sphinx наиболее удачным решением именно для помощника, выполняющего определенные команды, а онлайнные sdk яндекса и гу хороши для распознавания произвольной речи, но это другой круг задач.


 tutam 18 марта 2018 в 22:49    

Через pip устанавливается прекрасно.

Через pip устанавливается прекрасно потому, что у вас стоит swig.

В остальном я все сделал также как вы(без языковой модели, со своим словарем и грамматикой), только точности не получил.

И еще: поделитесь секретом, как вы сделали активационную фразу?

 DollaR84 19 марта 2018 в 00:22    

Ну насчет swig я не знал, может быть. Да я и сейчас не знаю установлен он или нет, никогда не возникал такой вопрос.

И не буду говорить, что я его полностью закончил или все идеально. Я постоянно продолжаю что-то добавлять или изменять, много бывает ебашек, но постепенно допиливаю потихоньку.

А насчет активационной фразы, так у меня тоже возникли проблемы с ее использованием, я читал документацию про keyphrase, но мне на тот момент не удалось настроить ее порог срабатывания, и я ее оставил до лучших времен, вот думаю скоро снова заняться может быть этой проблемой.


А на данном этапе пока как временное решение я поступил немного по другому.

У меня все команды разделены по разделам, плюс основной раздел, плюс активационная фраза.

Все разделы разделены по разным грамматическим правилам, в том числе фраза для активации представлена также отдельным правилом.

При инициализации я их всех загружаю функцией set\_jsfg\_file(), с указанными именами, а затем в цикле крутится распознавание, сначала активационного правила. Когда фраза найдена, происходит смена set\_search() на основное правило, которое уже ждет распознавание по разделу и так далее. При окончании или отмене, возвращается правило с активационной фразой.

Я конечно согласен, что это костыль тот еще, поэтому и настроен на переделку с применением правильного решения по документации, так как бывают ложные срабатывания, но пока не хватило еще времени и желания дойти до этого. И так потрачено было много времени на эту домашнюю поделку =)

 delvin-fil 21 марта 2018 в 17:31   

По поводу синтеза.

Прогой trans из пакета translate-shell звучит отлично. Может и гуглом и яндексом и еще многими движками. Я предпочитаю яндекс. Пять голосов.


echo «Тестовый звук» | trans :ru -b -e yandex -no-warn -no-view -sp -n ermil

 DollaR84 21 марта 2018 в 20:16    

Ну про более ранние версии windows не скажу, но начиная с версии 8.1 и 10 в поставку windows включен вроде неплохой синтезатор голоса.



В русской версии — это голос Ирина, по крайней мере я ее использовал.

Или у данного синтезатора есть недостатки еще какие?




wbsr

22 марта 2018 в 17:13







↑

Привет! Я автор сайта pythono.ru, с которого вы взяли пример с speech recognition  
Синтез речи я сейчас использую от Asapella Group голос Алёна, работает оффлайн из Python вызываю через консольную версию программы balabo работает отлично  
А вот с rocketsphinx у меня давно была идея чтобы использовать его для распознавания ключевой фразы типа «Окей гугл» (и только её), а остальные команды распознавать через гугл онлайн как и раньше.  
Но я в своё время так и не разобрался с rocketsphinx поэтому мой голосовой ассистент так и работает по нажатию на кнопку  
Спасибо вам за статью  
Если интересно поработать вместе над созданием голосового ассистента пишите мне в телеграм (написал вам контакты в личку здесь на сайте)



tutam

22 марта 2018 в 17:17



↑

Привет! Я автор сайта pythono.ru, с которого вы взяли пример с speech recognition

[habrahabr.ru/post/323570/#comment\\_10114606](https://habrahabr.ru/post/323570/#comment_10114606)  
И много авторов у этого сайта?




wbsr

22 марта 2018 в 17:24





↑

это мой второй аккаунт если что ))



tutam

22 марта 2018 в 17:15



↑

Комментарий случайно не туда написал написал, не обращайтесь внимания.

Только полноправные пользователи могут оставлять комментарии. [Войдите, пожалуйста.](#)

САМОЕ ЧИТАЕМОЕ

- Сутки
- Неделя
- Месяц

В Госдуму внесен законопроект об автономной работе рунета

↑

+63

👁

36k

📖

50

💬

388

Вопросы не мальчика, а джун. 22 вопроса работодателю на собеседовании на позицию «Middle Python-разработчик»

↑

+31

👁

23,1k

📖

223

💬

93

Повсеместное размытие спутниковых фотографий раскрывает местонахождение секретных баз

↑

+30

👁

16k

📖

28

💬

56

В гости к отцу

↑

+48

👁

107k

📖

55

💬

58

Иди-ка ты сам на... или правила общения в команде

↑

+176

👁

39,3k

📖

114

💬

255

[Пользователи](#)

[Соглашение](#)

[Семинары](#)

[Песочница](#)

[Конфиденциальность](#)



© 2006 – 2018 «ТМ»

[О сайте](#)

[Служба поддержки](#)

[Мобильная версия](#)

