Публикации

Пользователи

Хабы Ко

Компании

Песочница



Регис



Распознавание речи с помощью CMU Sphinx

Звук
Из песочницы

CMU Sphinx сейчас является крупнейшим проектом по распознаванию человеческой речи. В инструментарий входят следующие программи библиотеки:

- Pocketsphinx небольшая программа, которая принимает на вход произвольные акустические модели, грамматики и словари, а также звуковой поток(либо звуковой файл, либо сам берет поток с микрофона). На выходе получается распознанный текст. Написана на C, ра быстро.
- Sphinxbase библиотека необходимая для работы Pocketsphinx
- Sphinx4 гибкая библиотека для распознавания, написана на Java.
- Sphinxtrain программа для обучения акустических моделей.

Для работы со CMU Sphinx важно запомнить несколько определений и понять их отличия.

- Акустическая модель отвечает за сопоставление звуку произнесенной фонемы. Акустическую модель для русского языка можно ска сайте проекта. Русская акустическая и языковая модели. А также словарь.
- Словарь это файл, в котором написаны сопоставлены лексемы и фонемы (слово и его транскрипция). Например, калькулятор (k ay ja t ay r). Он необходим для преобразования фонем, распознанных акустической моделью в лексемы.
- Грамматика это формальные правила, которые описывают простые правила построения предложений. Лексемы, полученные на предыдущем шаге пытаются сопоставиться с грамматикой и если удачно, то выводится результат.
- Языковая модель это статистическая модель языка. Она описывает вероятности слов и их комбинаций. Таким образом распознаван лексем это максимизация правдоподобности распознанной фразы.

Чем сложнее язык, чем обширней правила и размер словаря, тем хуже точность распознавания. Поэтому, для минимизации ошибки, имеет создания упрощенных правил, которые будут описывать конкретную задачу.

Наш формальный язык

Пусть наша программа будет реагировать на фразы «ок, компьютер» или «ок, калькулятор» за которой будет идти простое выражение. Наг

ок, компьютер сорок семь плюс двадцать один.

Начнем описывать формальную грамматику.

Приветствие или активация (символ | означает или, скобки используются в качестве группирующего оператора):

= (ок|окей) (компьютер|калькулятор);

Однозначные числа:

= (ноль | один | два | три | четыре | пять | шесть | семь | восемь | девять);

Двузначные числа (это числа от 10 до 19, а также 20,30,40,50,60,70,80,90 с опциональным однозначным):

= (десять | одинадцать | двенадцать | тринадцать | четырнадцать | шестнадцать | семнадцать | восемнадцать | девятнадцать) | (двадцатридцать | семнадцать | семнадцать | девятнадцать | семнадцать | девятнадцать | семнадцать | семнадцать | девятнадцать | девятнадцать | девятнадцать | семнадцать | девятнадцать | девятн

Трехзначные числа (100, 200, 300, 400, 500, 600, 700, 800, 900 плюс опциональное двузначное или однозначное):

= (сто|двести|триста|четыреста|пятьсот|шестьсот|девятьсот)[|];

https://habr.com/post/267539/

```
Macroshaчные числа (п тысяч плюс опциональные двузначные или однозначные):

= [||] (тысяча|тысяч|тысячи)[||];

На этом я решил остановиться, но не сложно добавить еще одно правило для миллиона, миллиарда и прочих больших чисел.

Общее правило для числа:

= |||;

Теперь опишем возможные операции:

= плюс | минус | умножить на | разделить на;

Выражение:

= ;

И собственно сам запрос, то что мы и хотим распознавать:

public = ;
```

Составление словаря

Теперь составим словарь всех возможных слов. В принципе, можно использовать готовый словарь для русского языка, который скачивается вместе с акустической моделью. Ведь все эти слова наверняка там есть, а грамматика просто не пропустит слово, ко в ней не описано. Однако чем меньше словарь - тем быстрее поиск по нему и зачем нам хранить в оперативке словарь на 500000 скогда мы используем лишь несколько десятков. Но если вам лень составлять свой, то можно просто подключить готовый, я уверен убудет работать. Сначала запишем все слова, которые будут корректны для нашего языка. Вот возьмем прямо все слова что мы описк грамматике и запишем их в файл по слову на строку. Получится что-то вроде этого.

Для того чтобы получить транскрипцию для этих слов можно воспользоваться проектом ru4sphinx. Скрипт dict2transcript.pl прини на вход 2 аргумента(наш файл со списком слов и выходной файл, куда записать слова и их транскрипции).

Получившийся словарь имеет следующий вид:

Файл с грамматикой можно взять с гитхаба calc.jsgf.

спойлер

И доступен для скачивания.

Проверка работы

Попробуем распознать несколько команд, воспользуется для этого pocketsphinx. Pocketsphinx_continuous принимает множество разпараметров я опишу лишь некоторые из них, они будут необходимы для запуска:

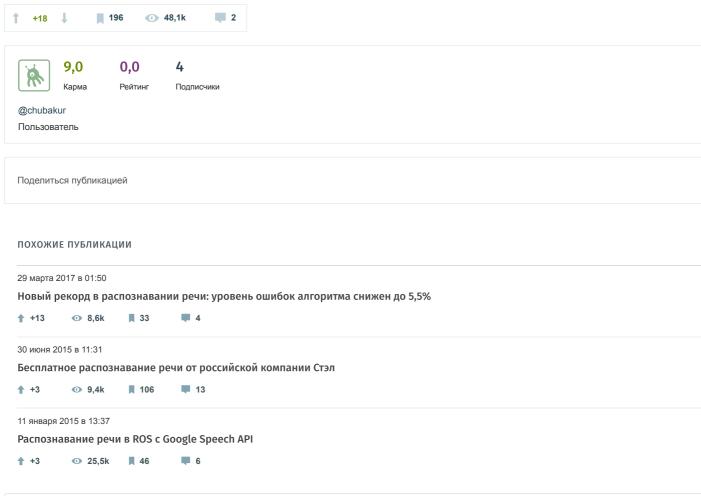
- -hmm <путь к акустической модели>. Если вы скачивали модель по приведенной выше ссылке, то акустическая модель будет нахо в папке zero_ru_cont_8k_v3/zero_ru.cd_cont_4000 .
- -dict <путь к словарю>
- -jsgf <путь к грамматике>
- -lm <путь к языковой модели>
- -logfn <путь к файлу для логгирования>. По умолчанию лог пишется в stdout.
- -infile <путь к файлу с голосом>. Надо удостовериться что частота дискретизации файла совпадает с частотой дискретизации модели.
- \bullet -inmic <yes|no>. Звук в реальном времени с микрофона.
- -remove_noise <yes|no>. Фильтрация шумов. По умолчанию yes.

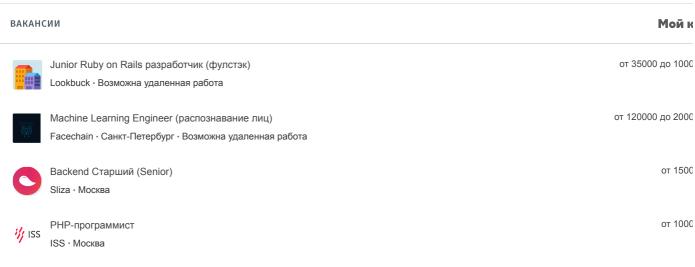
pocketsphinx_continuous -hmm zero_ru.cd_cont_4000 -dict calc_lang/vocabular.dict -jsgf calc_lang/calc.jsgf -inmic yes

https://habr.com/post/267539/

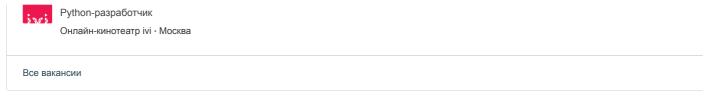


Теги: cmu sphinx, cmusphinx, распознавание речи, распознование голоса

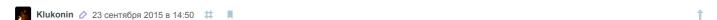




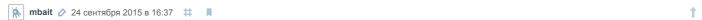
от 140000 до 1800



Комментарии 2



А на деле придется учитывать то что числа распознаются отдельно. Если следом за числом не идет действие — числа суммируются до тех пор, пок парсер не встретит математическое дейтсвие. И потом производит его с такой же суммой чисел. Как-то так я это вижу.



Для приветствия существует тип поиска keyword search, читайте справку по параметрам `-kws` и `-kws_search`. Этот метод распознает лучше чем грамматика.

Однако чем меньше словарь — тем быстрее поиск по нему и зачем нам хранить в оперативке словарь на 500000 слов

Словарь весь не грузится — только те слова, что используются в языковой модели.

<n3> = (сто|двести|триста|четыреста|пятьсот|шестьсот|девятьсот)[<n2>|<n1>];

Можно переписать оптимальнее:

```
<n34> = ( три | четыре ) ста ; <n59> = ( пять | шесть | семь | восемь | девять ) сот ; <n3> = ( сто | двести | <h34> | <h59> <n2> | <n1> ] ;
```

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ



В Госдуму внесен законопроект об автономной работе рунета

↑ +63 ③ 36k ■ 50 ■ 388

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

Повсеместное размытие спутниковых фотографий раскрывает местонахождение секретных баз

В гости к отцу

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

↑ +176 ③ 39,3k ■ 114 ■ 255

Распознавание речи с помощью CMU Sphinx / Хабр

Аккаунт	Разделы —— Публикации		Информация — Правила		Услуги Реклама	Приложения
Войти						Загрузите в App Store Google
Регистрация	Хабы		Помощь		Тарифы	
	Компании	До	окументация	Я	Контент	
	Пользователи	Co	оглашение		Семинары	
	Песочница		Конфиденциальность			
TM © 2006 – 2018 «TM»	О сайте	Служба подд	цержки	Мобильная верси	Я	

https://habr.com/post/267539/