Анализ звукозаписей уроков иностранного языка

Определение уровня освоения лексического минимума при помощи доступных решений ASR

Капелюшник Денис

Цель проекта

Разработать инструмент анализа аудиозаписей уроков иностранного языка на базе доступных решений в области автоматического распознавания речи для дальнейшей оценки освоения лексического минимума.

Основные задачи

- ✓Подавление тишины
- ✓Сегментирование
- ✓Кластеризация
- ✓Подбор модели ASR
- ✓Распознавание речи
- ✓Поиск целевой лексики в полученном результате.

Используемые инструменты

- ✓Ubuntu 18.04
- ✓ffmpeg
- ✓webrtcVAD
- ✓Kaldi
- ✓VOSK Speech Recognition API
- ✓fuzzywuzzy

Характеристика данных

Ввод

любая аудио- или видеозапись урока >16 kHz

Тренировка модели

Телефонные записи 8-16 kHz ~15-30% WER

Модель

vosk-model-small-en-us-0.3 ~36МБ

Pipeline

1. Обработка входного сигнала
2. Предобработка целевой лексики
3. Подавление тишины и сегментирование
4. Извлечение индивидуальных характеристик сегментов
5. Кластеризация
6. Сбор статистики

Обработка входного сигнала

```python
def read_audio(path, sample_rate):
    """returns raw data of the file
    Takes the path, and returns PCM audio data.
    """
    # get raw data
    ffmpeg_process = subprocess.Popen(['ffmpeg', '-loglevel', 'quiet', '-i',
                                       path,
                                       '-vn', # disable video
                                       '-ar', str(sample_rate), # set frame_rate
                                       '-ac', '1', # get only left channel
                                       '-f', 's16le', '-' # set format
                                       ],
                                       stdout=subprocess.PIPE)
    pcm_data = ffmpeg_process.stdout.read()
    return pcm_data
```

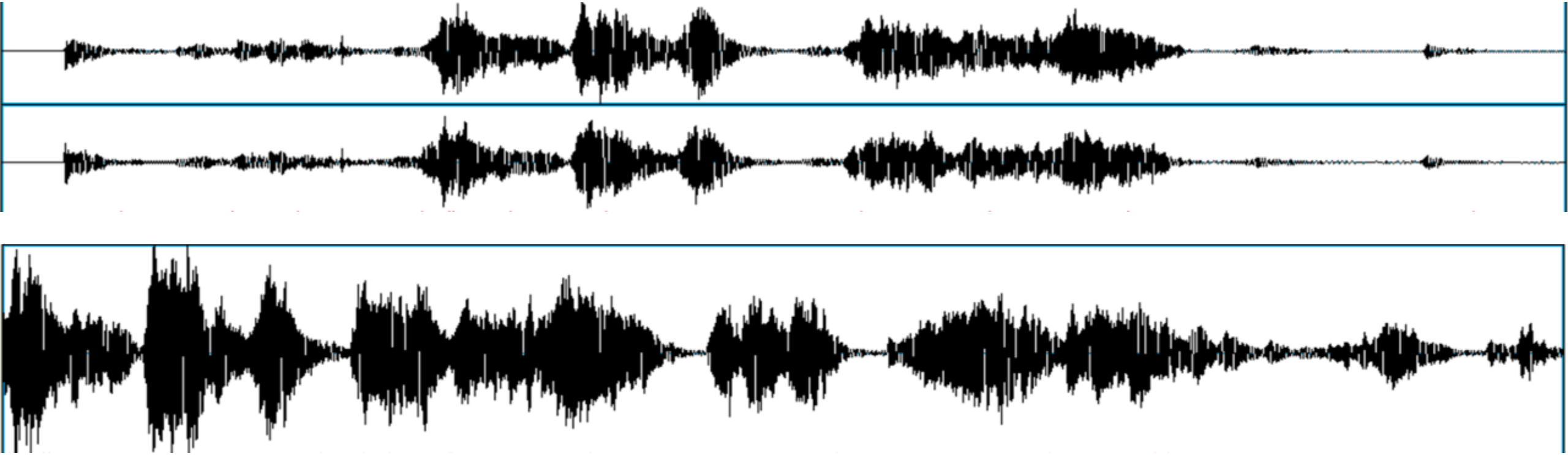
```python
class LessonSegment(Vocabulary):

    """docstring for LessonSegment."""
    def __init__(self, target_vocabulary, bytes):
        super().__init__(target_vocabulary)
        self.bytes = bytes
        self.transcript = []
        self.statistics = {}
```

Предобработка целевой лексики

['shop', "butcher's", 'stationery', "greengrocer's", 'shoe', "sport's", 'department', 'store', "jeweller's", "chemist's", 'post', "newsagent's"]

butcher‘	butcher
stationeri	stationeri
greengrocer‘	greengroc
shoe	shoe
depart	depart
store	store
jeweller‘	jewel
chemist‘	chemist
newsagent'	newsag

Подавление тишины и сегментирование



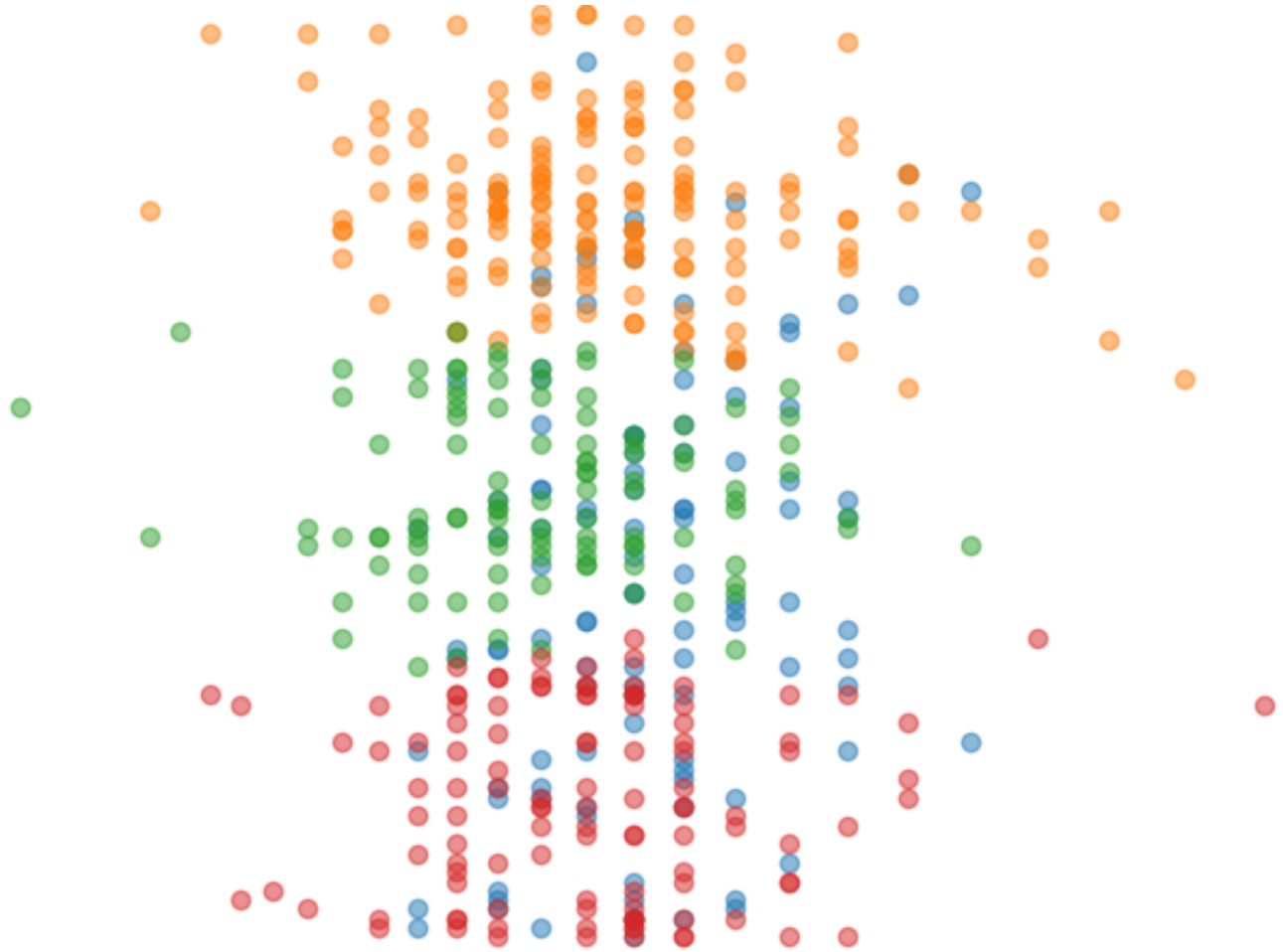
Извлечение индивидуальных характеристик

- Высота звука
- Громкость
- Темп

```
def get_features(self, sr):
    """
    calculates tempo and pitch using librosa
    documentation https://librosa.github.io/librosa/
    """
    timeseries = buf_to_float(self.bytes)
    pitch = estimate_tuning(timeseries, sr)
    # onset_env = onset_strength(timeseries, sr)
    # temp = tempo(onset_env, sr)[0]
    return([pitch])
```

Кластеризация

```
from sklearn.preprocessing import MinMaxScaler

from sklearn.mixture import GaussianMixture
```



Сбор статистики

```
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
```

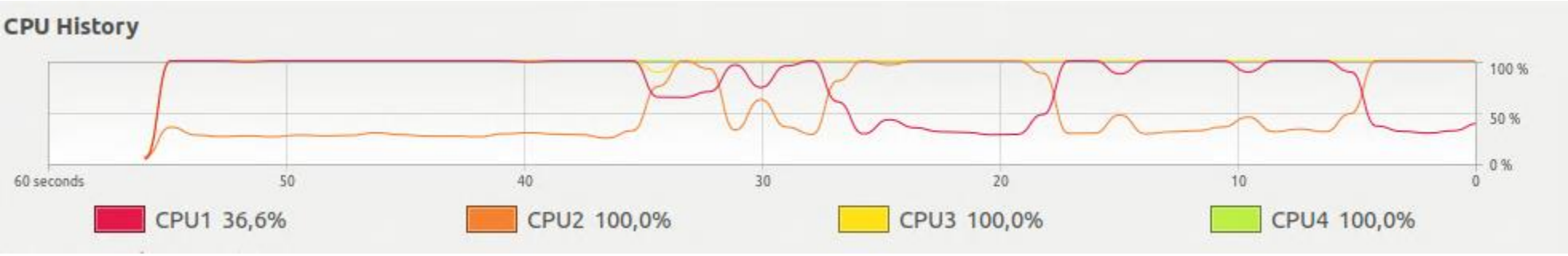
```
>>> fuzz.ratio('bike','like')
75
>>> fuzz.ratio('stare','stairs')
73
>>> fuzz.ratio('stare','stair')
80
>>> fuzz.ratio('rests','stair')
40
>>> fuzz.ratio('rast','stair')
44
>>> fuzz.ratio('tarse','stair')
60
>>> fuzz.ratio('tairs','stair')
80
>>> fuzz.ratio('airts','stair')
60
>>> fuzz.ratio('air','airplane')
55
>>>
```

```
def count_vocab(word, words):
    """Counts how many occurances of a word there are in a transcript"""
    return(len(
        process.extractBests(word, words,
                             scorer=fuzz.ratio,
                             score_cutoff=81)))
```

```
>>> process.extractBests('department', words, scorer=fuzz.partial_ratio, score_cutoff=80)
[('a', 100), ('a', 100), ('a', 100), ('a', 100), ('a', 100)]
>>> process.extractBests('department', words, scorer=fuzz.ratio, score_cutoff=80)
[('department', 100), ('department', 100)]
>>>
```

Направление дальнейшей работы

- ✓Реализация многопроцессорной обработки



- ✓Совершенствование / отказ от кластеризации
- ✓Выравнивание доменов модели ASR и входных данных
- ✓Создание интерфейса взаимодействия для имитации online ASR
- ✓Код: https://github.com/deniskapel/diarization