Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet
Matematički odsjek

Mogućnosti prepoznavanja govora korištenjem biblioteke SpeechRecognition

Margarita Tolja

Mentor: dr. sc. Goran Igaly, v. pred.

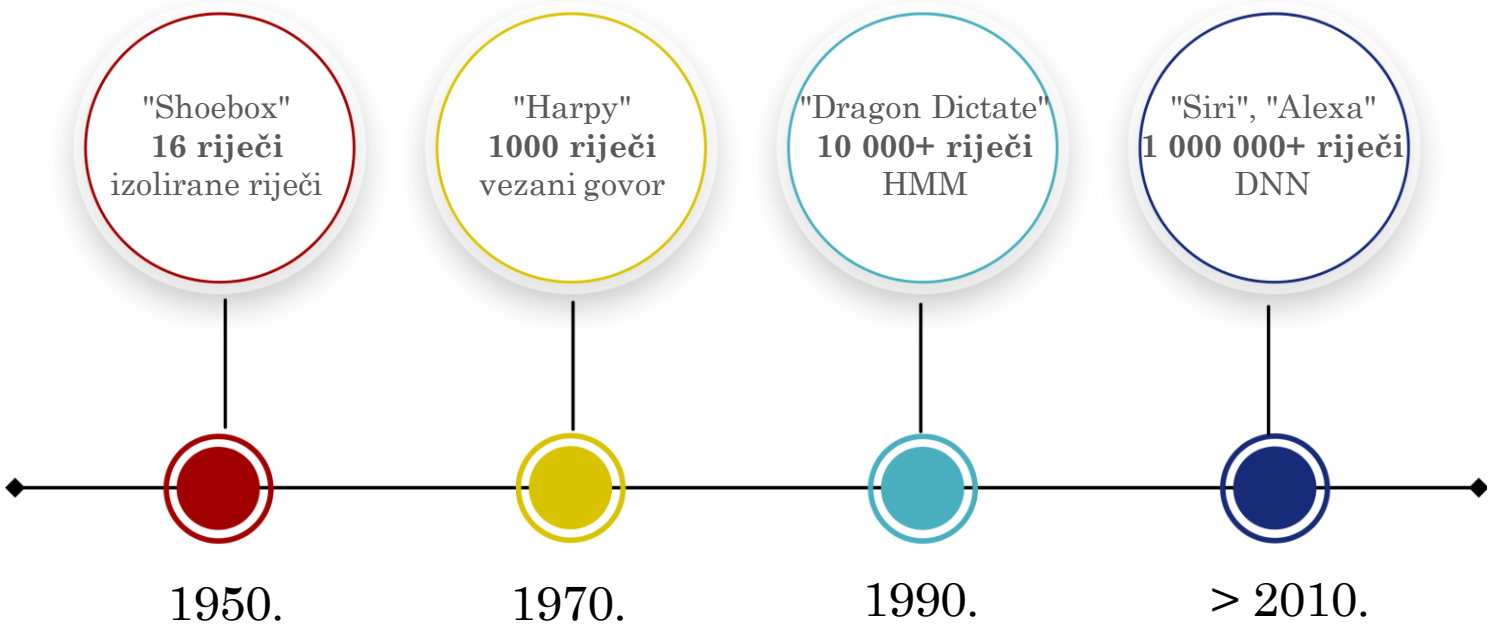
Zagreb, 2022.

1. **Automatsko prepoznavanje govora**
2. Biblioteka SpeechRecognition
3. Desktop aplikacija Transkripta

Automatsko prepoznavanje govora

Uvod u ASR sustave

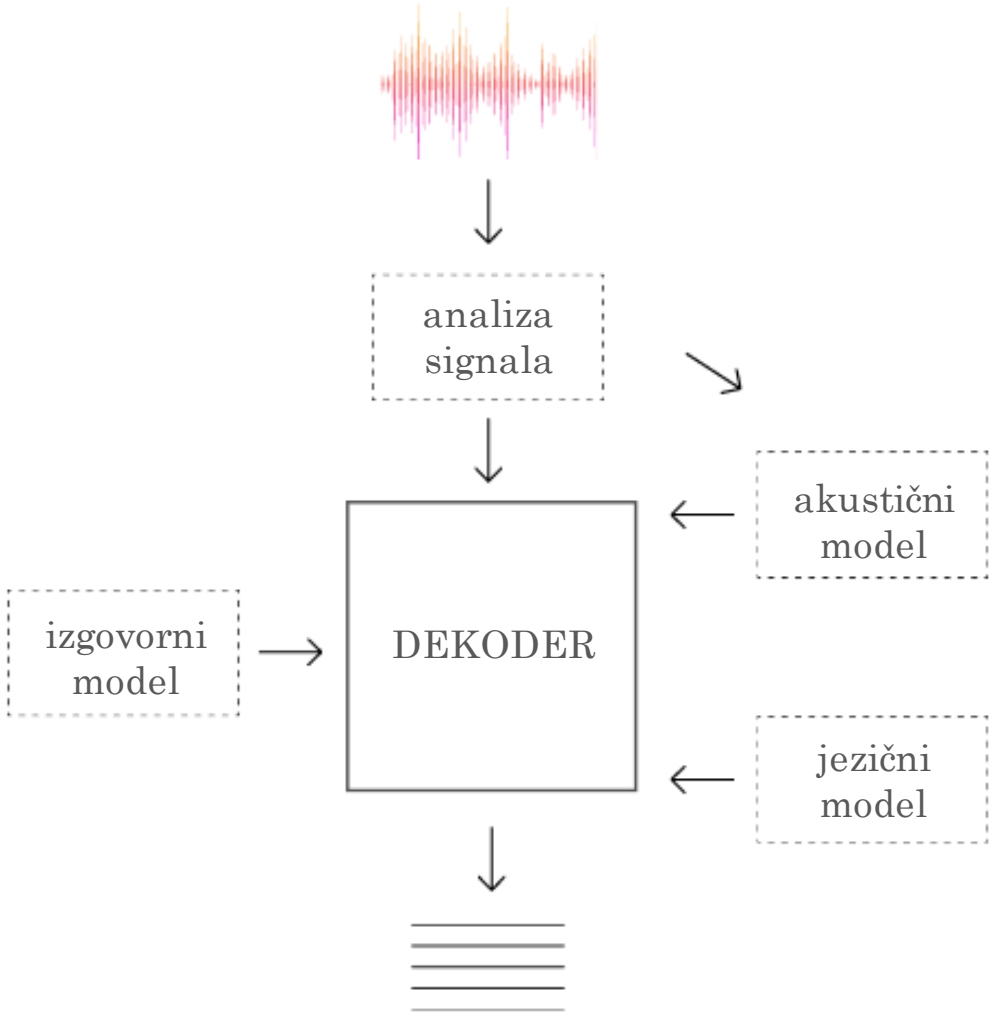
- *Automatic Speech Recognition* = pretvorba govora u tekst
- zahtjevan problem zbog puno izvora varijabilnosti
- konvencionalni i *end-to-end* sustavi



Razvoj ASR sustava

Konvencionalni ASR sustavi

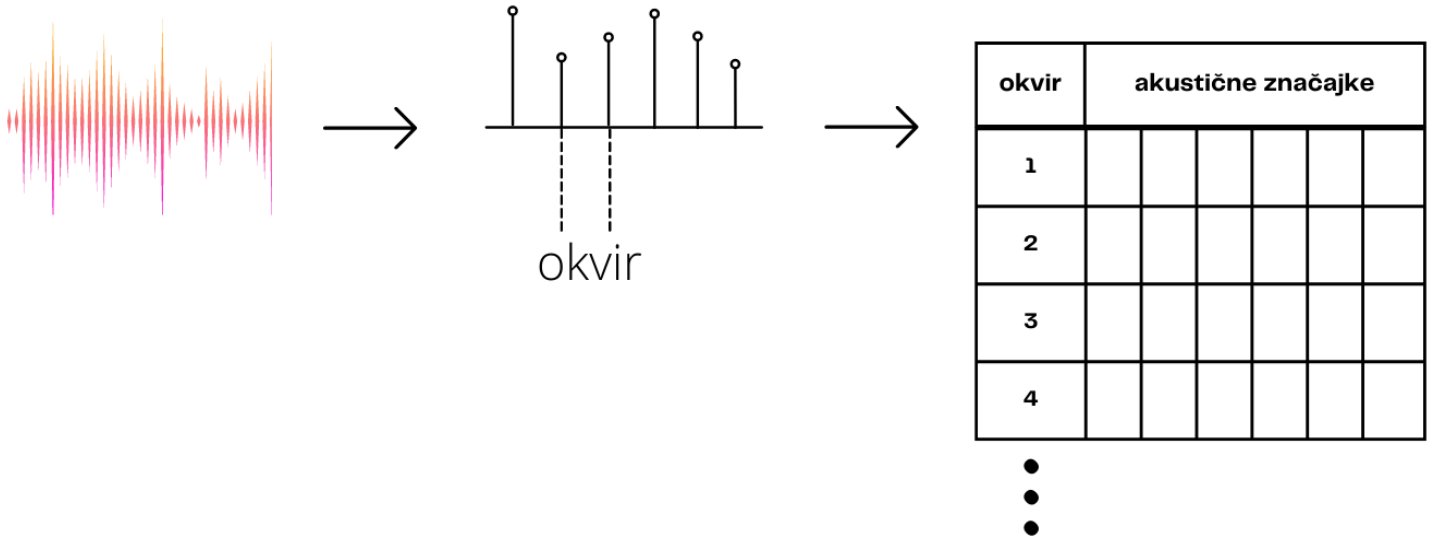
- najzastupljeniji u komercijalnoj upotrebi
- prepoznaje ulazni govor kroz faze (*pipeline)*
- 5 podsustava



Struktura sustava

Analiza zvučnog signala

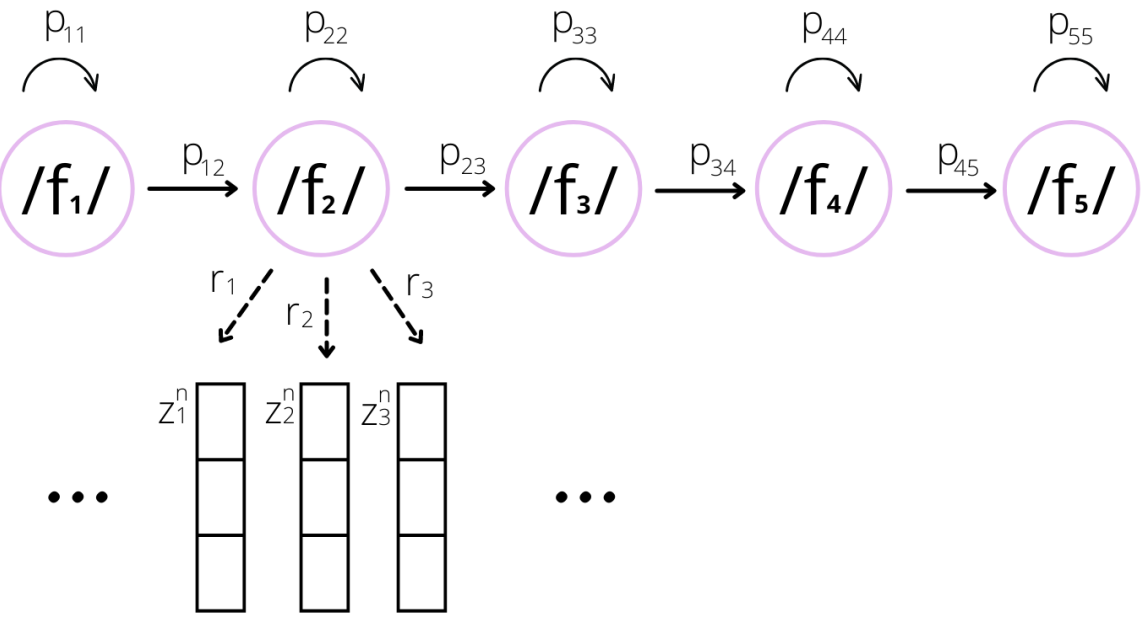
- digitalizacija zvučnog signala u 2 koraka
- kvazi-stacionaran signal se može diskretizirati
- metoda mel-frekvencijskih kepstralnih koeficijenata za određivanje značajki



Analiza po koracima

Akustični model

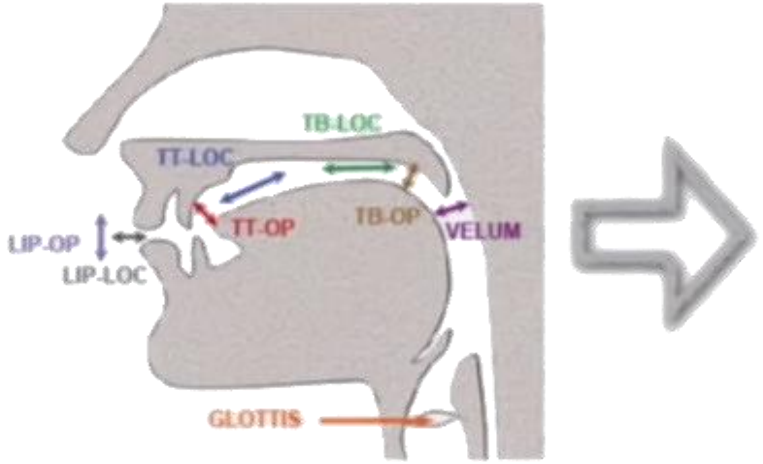
- određivanje niza fonema na temelju akustičkih značajki
- skriveni Markovljevi modeli za računanje *tranzicijskih* vjerojatnosti
- 2 načina za integraciju dubokih neuronskih mreža



Težinski graf – HMM

Izgovorni model

- grupiranje niz fonema u riječi
- problem: stvaran izgovor ne odgovara onom u rječniku
- rješenje: analiza stanja artikulatora umjesto fonema



Feature	Values
LIP-LOC	potruded, labial, dental, ...
LIP-OP	closed, critical, narrow, ...
TT-LOC	dental, alveolar, ...
TB-LOC	palatal, velar, uvular, ...
TT-OP, TB-OP	closed, critical, narrow, ...
GLOTTIS	closed, critical, open, ...
VELUM	closed, open, ...

Stanja artikulatora

Jezični model

- $\mathbb{P}(\text{"}I\ will\ be\ back\ soonish\text{"}) > \mathbb{P}(\text{"}I\ will\ be\ bassoon\ dish\text{"})$?

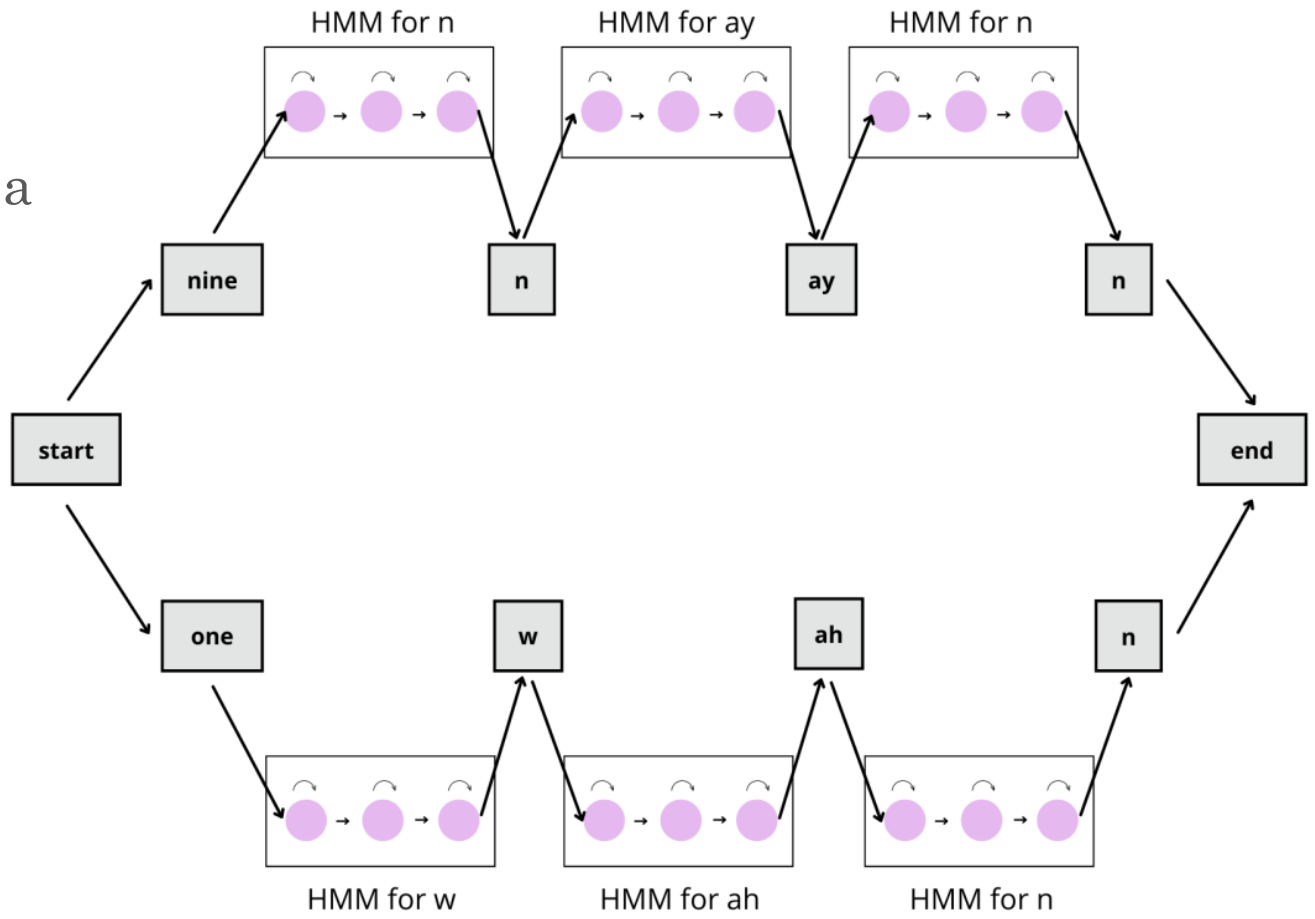
- određivanje najvjerojatnije sekvence riječi
- n-gram modeli su bazirani na sekvencama od n riječi:

$$\mathbb{P}(\text{"}back\text{"} \mid \text{"}I\ will\ be\text{"}) \approx \frac{\#(\text{"}I\ will\ be\ back\text{"})}{\#(\text{"}I\ will\ be\text{"})}$$

- funkcije za zaglađivanje (*smoothing methods)* pridaju vjerojatnosti sekvencama koje nisu u korpusu za učenje

Dekoder

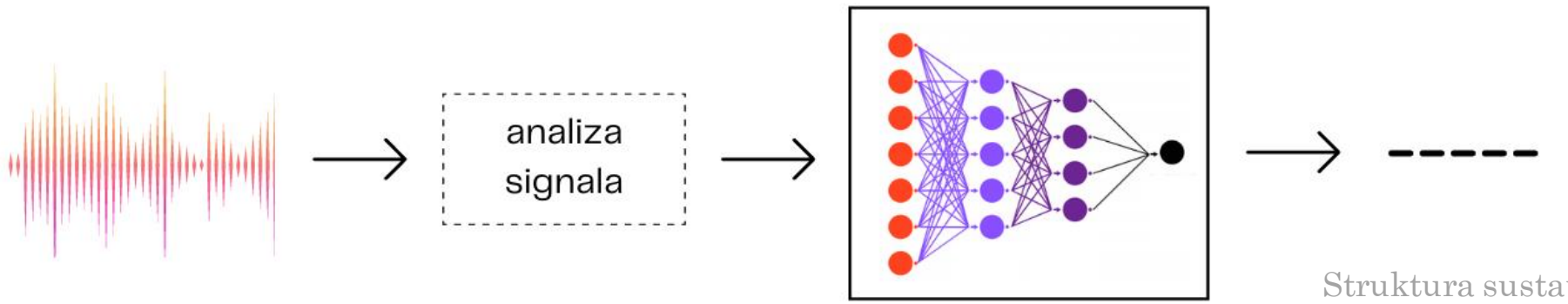
- Komponenta sabirnog tipa
- problem pretraživanja grafa
- graf se pretražuje aproksimacijskim tehnikama



Pretraživanje grafa za riječi "nine" i "one"

E2E ASR sustavi

- direktno preslikavanje sekvence zvučnog ulaza u sekvencu tekstualnog izlaza
- jedinstveni DNN model
- *spareni* podaci za treniranje – zvuk + tekst
- kriterij optimalnosti – *word error rate* (WER)



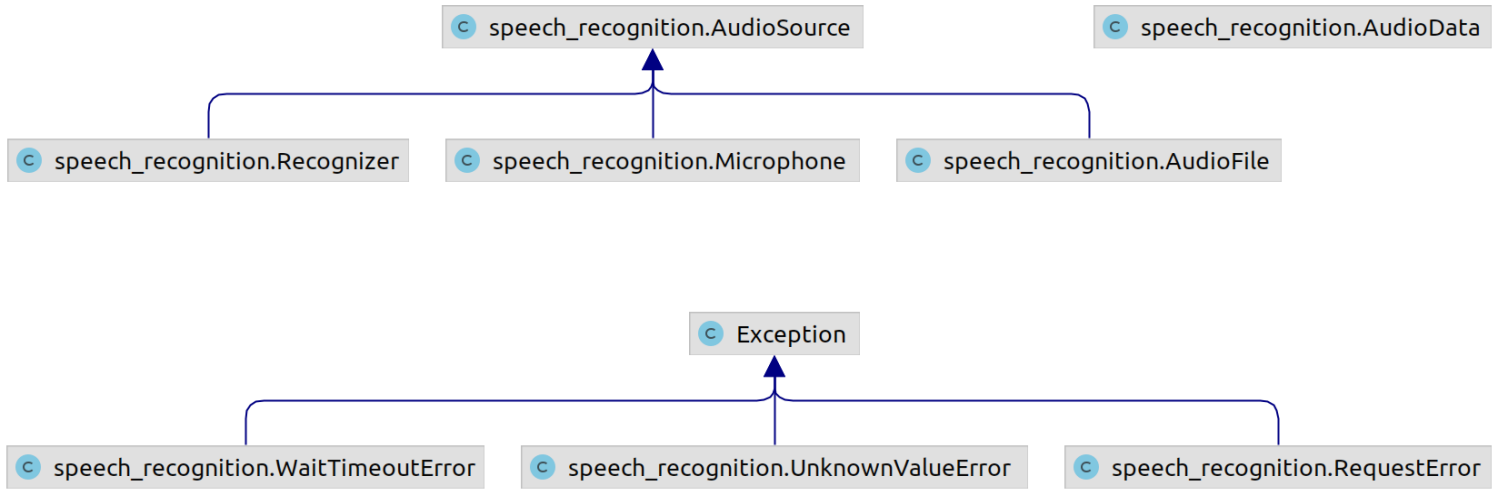
Struktura sustava

1. Automatsko prepoznavanje govora
2. **Biblioteka SpeechRecognition**
3. Desktop aplikacija Transkripta

Biblioteka SpeechRecognition

Uvod u biblioteku SR

- Python biblioteka otvorenog koda za prepoznavanje govora i upravljanje izvorom zvuka
- omotač prema vanjskim API-jima
- 3 faze prepoznavanja



Sučelje biblioteke

Izvor zvuka

- izvor se definira instanciranjem neke od AudioSource potklasa
- *ContextManager* klase

```
>>> audioFile = speech_recognition.AudioFile('example_file.wav')
```

Audio datoteka

```
>>> speech_recognition.Microphone.list_microphone_names()
['HDA Intel PCH: ALC272 Analog (hw:0,0)', 'default']
>>> microphone = speech_recognition.Microphone(device_index=1)
```

Mikrofon

Obrada zvuka

- dohvaćanje audio podataka pomoću klase Recognizer
- postavke pozadinske buke
  - svojstva: energy_threshold, dynamic_energy_threshold
  - funkcija adjust_for_ambient_noise

```
>>> with audioFile as source:
        recognizer.adjust_for_ambient_noise(source, 1)
        audio = recognizer.record(source, 5, 5)
```

Audio datoteka

```
>>> with microphone as source:
        recognizer.adjust_for_ambient_noise(source, 1)
        audio = recognizer.listen(source, 5, 5,
                ['home/project/snowboy','home/models/siri.umdl'])
```

Mikrofon

Prepoznavanje govora

- funkcija recognize_*

```
>>> result = recognizer.recognize_google(audio)
```

- API-ji:
  1. Google Speech API
  2. **Google Cloud Speech-to-Text API**
  3. **Alat Pocketsphinx**
  4. Houndify Speech To Text Only API
  5. Wit.ai API
  6. ~~Microsoft Bing Speech API~~
  7. ~~IBM Watson Speech to Text API~~

- posebne opcije: preferirane fraze, ključne riječi, gramatika

Audio datoteka

```python
import speech_recognition as sr

recognizer = sr.Recognizer()
audioFile = sr.AudioFile('example_file.wav')

with audioFile as source:
    recognizer.adjust_for_ambient_noise(source, 1)
    audio = recognizer.record(source, 5, 5)

result = recognizer.recognize_google(audio)
```

Mikrofon

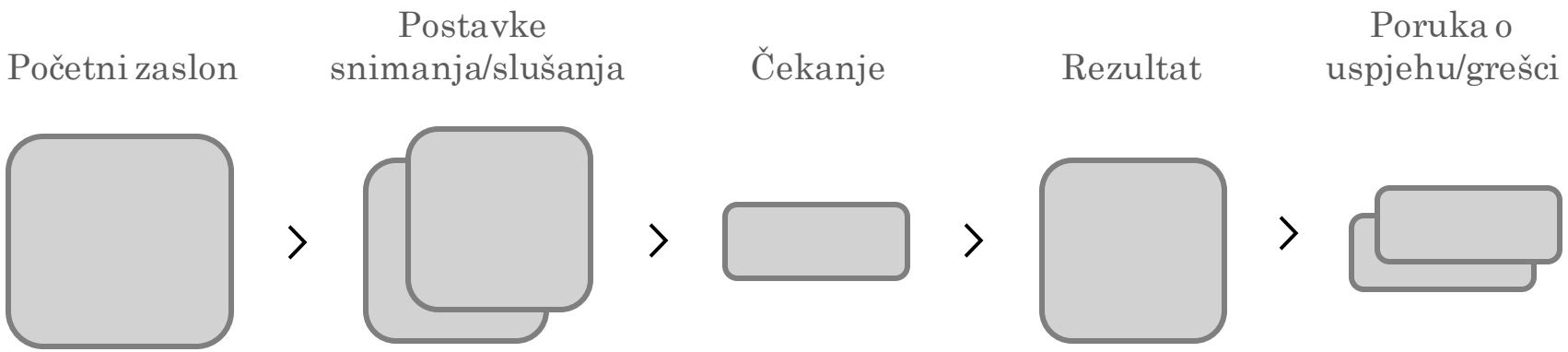
```python
import speech_recognition as sr

recognizer = sr.Recognizer()
microphone = sr.Microphone(1)

with microphone as source:
    recognizer.adjust_for_ambient_noise(source, 1)
    audio = recognizer.listen(source, 5, 5,
                ['home/project/snowboy','home/models/siri.umdl'])

result = recognizer.recognize_google(audio)
```

1. Automatsko prepoznavanje govora
2. Biblioteka SpeechRecognition
3. **Desktop aplikacija Transkripta**

Desktop aplikacija Transkripta

Funkcionalnost

- aplikacija za stvaranje transkripata
- ulaz – audio datoteka ili govor s mikrofona
- grafičko sučelje bazirano na dijalozima



Tok rada aplikacije

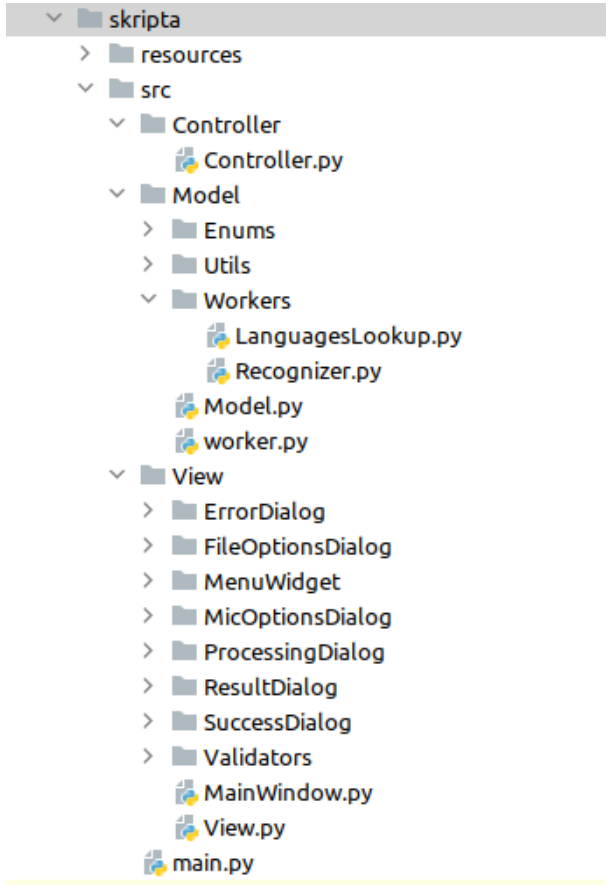
Razvojno okruženje

- Python 3.8.10
- PyCharm
- GUI – biblioteka PyQt 6.1.0

- pokretanje iz izvornog koda ili izvršnog direktorija

Implementacija

- oblikovni obrazac *Model-View-Controller*
- *View*
  - upravljanje GUI elementima
- *Controller*
  - povezivanje signala i utora
- *Model*
  - dohvaćanje podataka
  - prepoznavanje govora



Struktura direktorija

1. Automatsko prepoznavanje govora
2. Biblioteka SpeechRecognition
3. **Desktop aplikacija Transkripta**