**Veštačka inteligencija**

***Projekat 2***

**Zadavanje komandi glasom  
*Voice-User Interface (VUI)***

Profesor Leonid Stoimenov

Dušan Antić, 17004

Stefan Dobrosavljević, 17073

**1. Kratak opis problema**

Zadavanje komandi glasom predstavlja tehnologiju iz grupe trenutno aktuelnih koja omogućava korisnicima lakšu upotrebu električnih uređaja. Razvoj tehnologije vrtoglavo raste, a sa njim i olakšavanje života ljudi. Razumevanje mašine šta korisnik želi da kaže, odnosno šta očekuje od mašine da uradi i da pritom mašina pruži adekvatan odgovor, omogućeno je pomoću veštačke inteligencije.

**2. Pregled aktuelnih projekata**

Prepoznavanje govora i zadavanje komandi glasom se danas vrši na više načina, sa dobro razvijenim algoritmima. Međutim, nijedan metoda ne daje stopostotno dobre rezultate.

Kompanija Google je razvila *SpeechToText* API koji omogućava prevođenje govora u tekst. Prevođenje govora u tekst je veoma bitna karika u implementaciji zadavanja komandi glasom, jer u slučaju lošeg prevoda u tekst korisnik je još u startu osuđen na pogrešan odgovor mašine. Pored Google-ovog, postoje i mnoga druga rešenja koja pružaju identične usluge poput IBM-ovog *Watson Speech to Text*, *Sphinx* sa CMU-a (Carnegie Mellon University)…

**3. Formulacija problema**

U osnovi, postupak prevođenja govora u tekst je sledeći: uzimamo talasni oblik, delimo ga u odnosu na pauze u govoru, a zatim pokušavamo da prepoznamo šta se govori u svakom izgovoru. Da bismo postigli željeni rezultat, uzmemo sve moguće kombinacije reči i pokušamo da ih uskladimo sa izgovorom. (1)

Nakon prepoznavanja, potrebno je proveriti da li izgovorene reči pripadaju skupu komandi. Ukoliko se prepozna komanda, mašina izršava komandu uz eventualno korišćenje izgovorenih parametara.

Primer VUI-a kod pametnih telefona: Ukoliko osoba izgovori “call Stefan“, mašina će prepoznati komandu *call* i parametar *Stefan*, koji će biti iskorišćen za pretragu kontakata. Krajnji rezultat je poziv kontakta sa imenom *Stefan* ukoliko postoji u imeniku.

**4. Kratak opis rešenja**

Rešenje ćemo prikazati na primeru gotove aplikacije - kalkulatora(4) – *Python Voice Command Calculator* koji vrši osnovne matematičke operacije i probleme koji se zadaju usmeno na engleskom jeziku.

Aplikacija je napisana u programskom jeziku Python, uz pomoć bioblioteke speech\_recognition.

Osnovni zahtev je da uređaj na kome se izvršava aplikacija ima povezan mikrofon preko kojeg korisnici zadaju komande. Takođe, uređaj mora da poseduje instaliran Python interpreter.

**5. Detalji rešenja problema**

U aplikaciji *Python Voice Command Calculator* iskorišćene su dve biblioteke:

* operator – pruža mogućnost dinamičkog izbora operacije i njenog izvršenja u realnom vremenu.
* speech\_recognition – pruža direktan interfejs za prepoznavanje govora i njegovo prevođenje u tekst(string). U sklopu biblioteke postoji podrška za više različitih Application programming interface-a (API), među kojima je i Google Speech Recognition API koji će biti korišćen u ovoj aplikaciji.

Nakon uvoza biblioteka(linije 1 i 2), sledi izbor ulaznog uređaja – mikrofona i poziv funkcija za eliminisanje ambijentalnog šuma (linije 5-9).

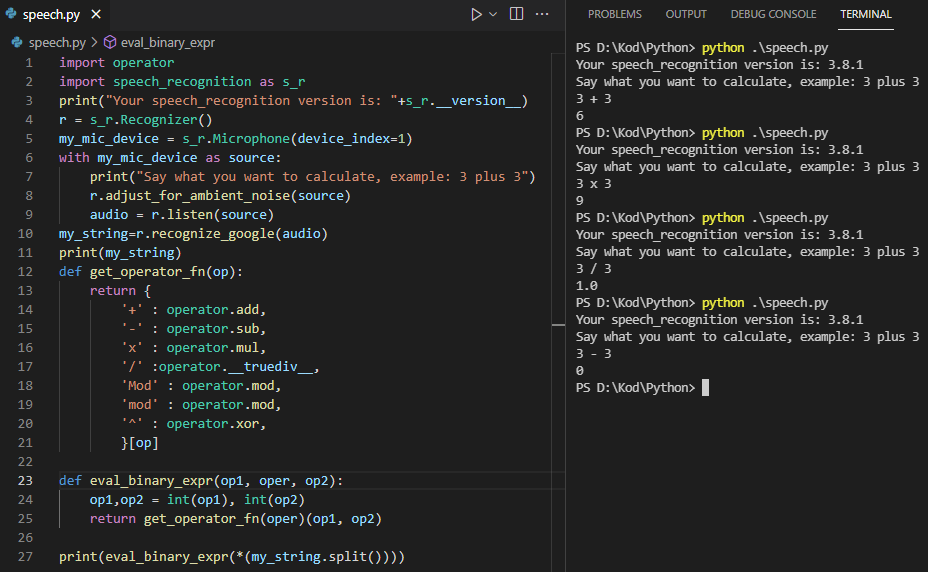
Kada je mikrofon izabran i podešen, pozivamo funkciju za prepoznavanje glasa. Sada se od korisnika očekuje izgovaranje komande – konkretnog matematičkog problema sa dva operatora i podržanom operacijom(sabiranje, oduzimanje, množenje, deljenje, stepenovanje, moduo) u formatu: {broj} {operacija} {broj}.

Prvi korak kod prevođenja govora u tekst je proces signala. Mikrofon konvertuje govor u digitalni signal koji mašina razume. Taj digitalni signal se metodom *Fast Fourier Transformation* (FFT) transformiše u spektrogram. Obradom, uz korišćenje fonetskih biblioteka, spektrogrami se transformišu u foneme. Fonemi (6) su najmanja jezička jedinica i njihovim kombinovanjem se dobijaju jedinice koje imaju značenje i koje su višeg reda u jezičkom sistemu – reči. U početku je za prepoznavanje fonema korišćen *Hidden Markov Model* (HMM). Nedostatak ovog modela je taj što se ne može adaptirati na veliku varijaciju fonema. Kako svi fonemi moraju da budu predefinisani i kako ljudski govor varira zbog akcenata i pogrešnog izgovora, HMM više nije bio adekvatna opcija. Nova alternativa su neuronske mreže. Prednost neuronskih mreža je taj što se one menjaju vremenom i bivaju sve bolje i bolje. Neuronske mreže zahtevaju mnogo podataka kako bi bile pouzdane. Fonemi se nakon prepoznavanja obrađuju jezičkim modelom. Jezički model analizira koja je verovatnoća da se jedan fonem pojavi nakon drugog. Spajanjem tih fonema dobija se reč. Zatim se proverava da li se ta reč nalazi u bazi. Ako se dobijena reč ne poklapa ni sa jednom, uzima se reč koja ima slične foneme. *Google Speech Recognition* API koristi najnaprednije metode neuronskih mreža za prepoznavanje govora.

Pored osnovnih poziva funkcija, korišćene su i dve pomoćne funkcije:

* get\_operator\_fn(op) – na osnovu teksta op, funkcija vraća izgovoreni operator.
* eval\_binary\_expr(op1, op, op2) – na osnovu izgovorenih operatora (op1 i op2), kao i operacije op, funkcija vraća rezultat izraza.

Na slici 1 se može videti izgovaranje izraza 3+3, 3x3, 3/3 i 3-3, kao i ispis rešenja u novom redu.



*Slika 1 – Implementacija aplikacije sa izgovorenim primerima*

**6. Literatura**

1. <https://cmusphinx.github.io/wiki/tutorialconcepts/>
2. <https://dev.to/mshrish/making-a-simple-voice-controlled-personal-assistant-interface-using-python-5ce1>
3. <https://www.rt-rk.uns.ac.rs/sites/default/files/e12369_miroslav_stefanovic.pdf>
4. <https://www.codespeedy.com/voice-command-calculator-in-python/>
5. <https://www.youtube.com/watch?v=iNbOOgXjnzE&ab_channel=SamStork>
6. <https://sr.wikipedia.org/sr-el/%D0%A4%D0%BE%D0%BD%D0%B5%D0%BC%D0%B0>