Metoda automatske detekcije naglašenih riječi u zvučnom zapisu

Stojanović, Aleksandar

Sveučilište u Zagrebu

FILOZOFSKI FAKULTET

Aleksandar Stojanović

METODA AUTOMATSKE DETEKCIJE NAGLAŠENIH RIJEČI U ZVUČNOM ZAPISU

DOKTORSKI RAD

Mentor:
prof. dr. sc. Nikolaj Lazić

Zagreb, 2019

University of Zagreb

FACULTY OF HUMANITIES AND SOCIAL SCIENCES

Aleksandar Stojanović

A METHOD FOR AUTOMATIC DETECTION OF EMPHASIZED WORDS IN RECORDED SPEECH

DOCTORAL THESIS

Supervisor:
Nikolaj Lazić, PhD

Zagreb, 2019

O mentoru

Prof. dr. sc. Nikolaj Lazić rođen je 1973. godine u Zagrebu gdje je završio osnovnu i srednju školu. Diplomirao je 1999. godine na Filozofskom fakultetu Sveučilišta u Zagrebu na Odsjeku za informacijske znanosti i Odsjeku za fonetiku, grupe Informatologija (smjer Opća informatologija i smjer Fonetika). Nakon diplomiranja radio je u Nacionalnoj i sveučilišnoj knjižnici na poslovima voditelja informatizacije.

Poslijediplomski studij Informacijskih znanosti upisao je 1999. Doktorirao je 2006. godine obranom teme pod naslovom Modeliranje strojnih postupaka za izgovaranje teksta pisanoga hrvatskim jezikom.

Godine 2000. prihvaćen je kao znanstveni novak na projekt 130751 Hrvatska standardna prozodija riječi kojega je voditelj bio prof. dr.sc. Ivo Škarić. Nakon toga je sudjelovao na projektu 0130451 Istraživanje općega svehrvatskoga govora, istoga voditelja, koji se izvodio na Odsjeku za fonetiku Filozofskog fakulteta u Zagrebu. Bio je voditelj projekta Slobodne i uvjetovane izgovorne mijene jezičnih čestica u općem hrvatskome kojeg je financiralo MZOS.

Na Filozofskom fakultetu održava nastavu iz kolegija Teorija informacije i komunikacije, Računalna analiza i sinteza govora, Govorna tehnologija, Uvod u računalnu sintezu govora. Sudjeluje u izvođenju preddiplomskog i diplomskog studija Informacijskih znanosti na Filozofskom fakultetu u Mostaru.

Sudjeluje u izvođenju Doktorskog studija informacijskih i komunikacijskih znanosti na Filozofskom fakultetu Sveučilišta u Zagrebu, doktorskom studiju na Filozofskom fakultetu u Mostaru, te doktorskom studiju Sveučilišta Sjever.

Zahvale

Prije svega, zahvaljujem se mentoru prof. dr. sc. Nikolaju Laziću na podršci, suradnji i smjernicama tijekom izrade doktorskog rada. Naše tehničke diskusije bile su mi izuzetno korisne. Zahvaljujem se i voditeljici doktorskog studija, prof. dr. sc. Jadranki Lasić Lazić, na velikoj pomoći oko svih administrativnih stvari, te na poticanju i suradnji oko mnogih drugih aspekata mog studija. Od samog početka studija profesorica Lasić Lazić uvijek je bila na raspolaganju kada su mi trebale upute i savjeti oko daljnjih koraka mog studija. Njena pomoć bila mi je neizmjerno korisna, uključujući i podršku tijekom moje obrane doktorskog rada.

Također, zahvaljujem prof. dr. sc. Slavici Ćosović Bajić, bivšoj dekanici Tehničkog veleučilišta u Zagrebu (TVZ), na velikoj podršci ne samo u doktorskom radu nego i mom radu u nastavi na TVZu. Iz osobnog iskustva mogu reći da je profesorica Bajić uvijek pokazivala poštovanje ne samo prema meni nego i prema svima s kojima je radila i uvijek je bila spremna pomoći kad god je to bilo moguće. Dok je još bila dekanica TVZa profesorica Bajić podržavala je moj doktorski studij pa je velikim dijelom zaslužna za ovaj doktorat. Također joj zahvaljujem na velikoj podršci i lijepim riječima prilikom i nakon moje obrane doktorskog rada.

Sažetak

Prozodija je važan aspekt govora jer poboljšava informativnost izgovorenog. Jedan segment prozodije uključuje naglašavanje riječi kojim se ističe važnost jedne riječi u kontekstu sadržaja onoga što je izgovoreno, čime se može utjecati i na semantiku tog sadržaja. U tekstu, međutim, taj je aspekt izgubljen, pa je time izgubljena i ta dodatna informativnost napisanog sadržaja.

Cilj ovog rada bio je istražiti mogućnosti automatskog vraćanja informacija o naglašenim riječima u tekst koji je spremljen kao podnatpis ili transkript. To se željelo postići bez upotrebe potpuno automatskog sustava za prepoznavanje govora.

Naglašenost riječi analizira se kroz tri dimenzije:

- pojačani intenzitet,
- povišeni ton,
- produljeni (usporeni) izgovor.

Vraćanje ovih informacija u tekst obogaćuje njegovu informativnost, dok istovremeno, s tehničke strane, takav tekst zahtijeva puno manje memorijskog kapaciteta od zvuka, pa u tom obliku može biti pogodan tamo gdje postoji potreba za spremanjem velikih količina podataka, kao što je arhiviranje ili računalno pretraživanje. Isto tako, ovako obogaćeni tekst može biti koristan za osobe s oštećenim sluhom ili gluhonijeme osobe jer bi se njima na ovaj način olakšalo razumijevanje sadržaja time što bi im se približio izvorni oblik onoga što je i kako je bilo izgovoreno.

Ključne riječi: prozodija, prepoznavanje govora, naglašavanje riječi, podnatpis, intenzitet, frekvencija, ton, prepoznavanje govora, tekst

Summary

Prosody is an important aspect of speech because it complements the meaning of spoken communication. One segment of prosody includes word emphasis by which the importance of one word is emphasized in the context of what was spoken, which can affect the semantics of the spoken content. In written text, however, that aspect is lost, thereby losing this additional information.

The goal of this work was to develop a method of returning the prosodic component of speech back into text which is present through subtitles or transcript. Additionally, our goal was to achieve that without developing a full-scale speech recognition system.

Word emphasis is examined through three dimensions:

- increased intensity
- increased pitch
- extended duration of speech at particular words

Returning these aspects back into text enhances its informational contents, while at the same time, from technical perspective, such text would require much less storage space than sound, so such format can be useful in aplications that store large amounts of data, like archiving or information retrieval. Also, such enhanced text can be useful for people with hearing disabilities because it would make it easier for them to get a better understanding of how was something uttered.

This disertation is organized into several parts. The first part is the introduction. In the second part basic speech accoustics is described: physical properties of sound, frequency, tone, intensity, F0, formants, and accoustic properties of some phonemes with graphical representation of their spectrum and other accoustic properties. This part will also contain description of some accoustic properties of emphasized words that set them appart from other, nonemphasized words.

Part 3 contains description of some sound analysis techniques: spectrum, spectrogram, oscilogram, spectral analysis, LTAS, together with some methods of sound preparation which are important for its analysis, like speech annotation. This part also describes some capabilities of program Praat used in this research, together with some Python libraries.

Part 4 contains basics of machine learning and neural networks used in this research for phoneme classification. This part consists of basic introduction into machine learning and neural networks where their advantages compared to some other computational models are described in relation to sound analysis. After that one way of using such neural network in this work is described.

Part 5 contains detailed description of a method of speech analysis with the goal of detecting emphasized words. That method consists of several activities divided into the following steps:

- Speech annotation, where for each phoneme its sound segment is isolated (by hand). This is necessary for neural network training. This is a tedious process because a recording of just a few minutes contains hundreds of phonemes that need to be carefully annotated. Also, determining the beginning and the end of a phoneme is not always simple because phoneme can be uttered only partially, and can also appear one after another where it can be difficult to determine the phoneme boundaries.
- Creation of Praat script to iterate over segmented speech and perform spectral analysis for each phoneme. The result consists of LTAS values for each phoneme together with the letter categorizing the phoneme. These values are later used for training the neural network with speech of several randomly chosen speakers.

After this data preparation steps the next step is training the neural network. This process consists of several steps:

1. Elimination of variations in intensity. For neural network training we only need the spectral shape, so variations in intensity can create more variations for neural network to learn. To speed up the training process we need to eliminate variations in intensity as much as possible. One way to do this, as used in this research, is to increase or decrease all LTAS values by the amount necessary such that the largest value does not exceed some given intensity, but keeping all other values in the same ratio to each other as before.
2. Since the LTAS value range is not in the 0..1 interval the values need to be scaled. This is done because the neural network works only with the values in this interval.
3. The values are then organized into a data structure which contains the LTAS values and the category of the phoneme which these values represent. After that, neural network training is performed. The goal of this training is to later use the neural network to classify phonemes from new recordings not used for the training.

4. After the previous step the result would be a neural network trained for phoneme classification.

The next phase is the process of emphasized word detection. Before that, however, we extracted the transcripts from the media file to get the information on when on the recording these transcripts appear. This is important for determining later which words the classified phonemes belong to. For example, if in a speech segment phonemes „d..ava“ have been recognized and the text of the transcript in that sound segment contains letters „država” (croatian for *state*) then it is likely that these phonemes belong to this word. Then the analysis of pitch and intensity would determine if the word was emphasized.

After neural network training the detection of emphasized words consists of the following steps:

1. Phoneme classification from a speaker not used for neural network training. For phoneme classification we used two steps: First, the recording is partitioned into segments of 10 ms and for each of the segments the LTAS is calculated. Then, in the second step, the recording is marked with positions that contain glotal pulses (as calculated by Praat) and for each of those positions a segment of 5 ms before and after is selected for which LTAS is calculated. This second step helps avoiding skipping over some important sounds.
2. The result of the previous step is a sequence of phonemes which were the result from the classification process performed on those 10 ms sound segments. This phoneme sequence will contain the letter (phoneme) and time at which it appears in the recording. Some of those phonemes will be classified correctly, but some will not. For example, instead of classifying a phoneme as *m* the neural network might classify it incorrectly as *v* or some other phoneme. In order to determine which words were emphasized it is necessary to determine word boundaries. It is clear that the more correctly classified phonemes there are the easier it will be to find the word to which those phonemes belong. However, since many phonemes will be classified incorrectly, the text from the transcript needs to be matched against the phonemes produced by the neural network. This will be solved by using an alignment algorithm that will try to align the sequence of phonemes with the letters of the text from the transcript.
3. The result of the alignment will provide approximate information about where each word from the transcript begins and ends in the recording. Then the sound segment of

each words is analysed from the perspective of F0, intensity and duration, which determines if a word was emphasized.

Most of the previously described steps assumes creation of Praat and Python scripts by which these processes will be automated, which includes modules for testing and analysis of the results. Figures 1 and 2 show this process.

Part 6 contains results obtained from recordings of new speakers (those whose speech was not used for neural network training). These recordings include several speakers thereby showing how this method functions in different environments from those used for testing (regarding speech tempo, pitch, voice, speech patterns, etc.). Also, it shows the speech-to-text alignment precision.

Part 7 contains the conclusion. Here, the advantages and disadvantages of this method as compared to some others is discussed. Also, some alternatives are described as well, together with some possible improvements. Additionally, this part underlines the importance of having a larger corpus of annoated speech in croatian as a condition for many usefull future research in this area. Since the automatic recognition of phonemes in croatian is important for many research activities in this area (such as emotion detection, speaker identification, prosody analysis, etc.), such corpus would be an essential tool for this research.

Keywords: Neural network, phoneme, LTAS, spectrum, recognition, alignment, emphasized word, frequency, pitch, intensity

Sadržaj

1. Uvod

Za razliku od teksta, govor uključuje velik broj elemenata koji su dio govorne komunikacije. Jedan od tih elemenata je prozodija. Prema Škariću (Babić, i dr., 1991) prozodija uključuje ton i intonaciju, glasnoću i naglasak, boju glasa, spektralni sastav, stanke, govornu brzinu, ritam, govorne modulacije, način izgovora glasnika, te mimiku i geste. Naglasak je isticanje pojedinih slogova u riječi ili cijelih riječi unutar rečenice. To se očituje kroz govorna svojstva kao što su glasnoća, duljina izgovora pojedinih slogova i tona. Naglašavanje riječi daje dodatnu dimenziju značenju izgovorenog sadržaja, nešto što u pisanom obliku ne postoji, pa je stoga korist od vraćanja te informacije u tekst značajna. Da bi analizirali ove aspekte zvuka moramo upotrijebiti odgovarajuće alate koji nam omogućavaju detaljan uvid u karakteristike zvuka i mogućnost njegove analize sa stajališta relevantnih aspekata. Jedan od najvažnijih alata za ovakvu analizu zvuka je spektralna analiza. Spektar nam detaljno pokazuje intenzitet i frekvencije od kojih se zvuk sastoji, pa se pomoću toga mogu ustanoviti relevantna svojstva nekog segmenta zvuka - naglašenost se može vidjeti kao pojačani intenzitet, kao lagano povišenje osnovne frekvencije ili kao kombinacija ova dva pokazatelja. Da bi nekom automatskom metodom ustanovili je li neka riječ bila naglašena nije dovoljno samo analizirati zvuk sa stajališta ovih dvaju aspekata, već je neophodno tom metodom utvrditi o kojim se izgovorenim glasovima radilo na nekoj digitalnoj snimci govora. To je najteži problem kod ovakvih istraživanja zato jer su glasovi i riječi u govoru povezani; oni znaju biti prigušeni, neizgovoreni (u potpunosti), promijenjeni (zbog susjednih glasova), a gotovo uvijek postoje i dodatni zvukovi koji nisu dio jezika, kao što su razni šumovi i zvukovi koje proizvodi vokalni trakt. Kada bi promatrali detaljni prikaz (kao što je spektrogram) izgovorene riječi jednog govornika koji bi više puta izgovorio jednu te istu riječ, taj prikaz nikada ne bi u potpunosti bio isti. Možda bi intenzitet na nekom mjestu bio za nijansu manji ili veći, možda bi frekvencija na jednom mjestu bila malo niža ili viša, možda bi neki glas trajao malo kraće ili duže. To su stvari koje u slušanju govora ne primjećujemo, ali kada radimo njegovu detaljnu analizu onda ove razlike mogu doći do izražaja. Iz tog razloga nije moguće definirati jedan algoritam koji bi, na primjer, prepoznavao neki glas ili neku riječ - morali bi definirati na stotine ili čak tisuće takvih algoritama da bi pokrili sve moguće varijante izgovorenih glasova i riječi. Zbog toga je potreban drugačiji pristup, a to je da se umjesto nekog egzaktnog algoritma napravi postupak treniranja neuralne mreže na osnovu većeg broja testnih podataka. Ovdje neuralna mreža služi tome da se iz tih podataka "izvuku" neke karakteristike koje se uvijek pojavljuju kod takvih

podataka (što je ovdje zvučni signal), s očekivanjem da će te karakteristike biti prisutne i u stvarnim podacima (koji nisu bili upotrebljeni za treniranje), čime bi tada neuralna mreža mogla utvrditi o kojem se glasu radilo na osnovu toga što je "naučila" karakteristike takvih glasova iz svih prethodnih primjera upotrebljenih za njeno treniranje. Nakon što je ovakav postupak napravljen, neuralna mreža se može upotrijebiti za automatsko određivanje segmenta u zvuku unutar kojeg se nalazi neki izgovoreni glas. Kada se utvrdi gdje su unutar zvučnog signala izgovoreni pojedini glasovi, te unutar kojeg vremenskog intervala su ti glasovi izgovoreni, mogli bi utvrditi o kojoj se riječi radi. S obzirom da će u ovom radu zvučni signal biti popraćen podnatpisom, to će olakšati utvrđivanje izgovorene riječi jer često nisu svi glasovi prepoznati, a neki mogu biti prepoznati pogrešno. Na kraju, ako znamo gdje je unutar zvučnog segmenta izgovorena neka riječ možemo analizom intenziteta i frekvencije tog segmenta ustanoviti je li ta riječ bila naglašena ili ne.

Dvije osnovne metode prepoznavanja govora temelje se na neuralnim mrežama i skrivenim markovljevim modelima (engl. HMM, Hidden Markov Model). U posljednje vrijeme HMM je više zastupljen zbog toga što se pokazalo da daje bolje rezultate. Međutim, kao što su pokazali Varmuelen, Bernard, Yan, Fanty i Cole (Vermeulen, Bernard, Yan, Fanty, & Cole, 1996), neuralne mreže imaju prednosti što se tiče potrebnih računalnih resursa, a i nude neke mogućnosti za identifikaciju govornika. Većina radova na temu analize naglašavanja riječi u zvučnoj snimci govora bazira se na korištenju softverskih alata za prepoznavanje govora (engl. Automatic Speech Recognition, ASR) koji se najčešće temelje na jednoj od ovih dviju metoda. Takvi su alati danas prilično sofisticirani, u smislu da je točnost njihovog prepoznavanja riječi u povezanom govoru velika. Međutim, upotreba tih alata zahtijeva vremenski zahtjevnu pripremu koja se sastoji od konfiguriranja takvog sustava za jezik za koji ga želimo upotrijebiti. To uključuje izradu riječnika, definiranje jezičnog modela (kao što su liste riječi i gramatike), te izradu akustičkog modela (što uključuje „treniranje" sustava tako da mu se daju primjeri izgovorenih riječi ili rečenica). Ako se ovakve pripreme obave kako treba takvi sustavi su u stanju prepoznati većinu onog što je izgovoreno i, zavisno od sustava, označiti vremenske periode izgovorenih riječi. Međutim, da bi se to obavilo za neki jezik potrebno je dosta vremena (kod nekih sustava više od mjesec dana samo za treniranje akustičkog modela).

Arons B. u svom članku (Arons, 1994) opisuje postupak detekcije naglašenih riječi pomoću analize osnovnog tona kod govornika, gdje se sustav prvo „prilagodi" na visinu osnovnog tona, a nakon toga označi mjesta u govoru gdje je taj ton bio znatno povišen. J. Brenier, D. Cer i D. Jurafsky upotrebljavaju akustičku zajedno s leksičkom analizom (Brenier, Cer, & Jurafsky,

2005). Osnovni ton, intenzitet i trajanje upotrebljeni su zajedno s određivanjem vrste riječi, njenim mjestom u rečenici, frekvencijom pojavljivanja i drugim leksičkim informacijama. A. Slujter i V. Heuven pokazuju u (Slujter & Heuven, 1996) da se naglašeni slogovi očituju kao povišenje amplitude, a također se analizira osnovna frekvencija, trajanje, ukupni intenzitet, formanti i drugi parametri. U svom radu D. Kuijk i L. Boves (Kuijk, 1999) prikazuju akustičke razlike između naglašenih i nenaglašenih samoglasnika. R. Silipo i F. Crestani u (Silipo & Crestani, 2000) opisuju metode bazirane na naglašavanju riječi za određivanje teme govora, a slične metode obrađuju i M. Heldner, E. Strangert i T. Deschamps u (Heldner, Strangert, & Deschamps, 1999).

Iako je cilj ili tema gore spomenutih radova analiza naglašavanja riječi u govoru, ovaj rad je specifičan po tome da je ulazni podatak snimka govora zajedno s podnatpisima. To dobrim dijelom omogućava promjenu metode analize govornog signala jer je sam tekst onoga što se izgovara već prisutan na snimci, pa se stoga sustav za potpuno prepoznavanje govora može preskočiti. Nadalje, podnatpisi se često emitiraju s mnogih televizijskih kanala, pa se ova metoda može upotrijebiti za automatsko nadograđivanje samog teksta u svrhu njegovog objavljivanja ili arhiviranja.

Cilj ovog istraživanja je odgovoriti na pitanje kako je iz jedne snimke govora u prirodnom okruženju koja uključuje podnatpise (kao što su vijesti na televiziji) moguće automatski (to jest, upotrebom odgovarajućih softverskih alata) doći do informacije o tome koje su riječi bile naglašene, bez da se radi potpuno automatsko prepoznavanje govora. U tu svrhu bila bi definirana metoda analize digitalnog zapisa (koji sadrži zvuk i podnatpise) televizijskih vijesti kojom bi se ustanovilo koje su riječi koje se pojavljuju u podnatpisima bile naglašene. Ta bi analiza bila napravljena upotrebom softverskog alata za analizu zvuka.

Hipoteza ovog rada je da se informacija o naglašenim riječima iz jedne zvučne snimke govora s podnatpisima može vratiti u tekst bez da se u potpunosti radi automatsko prepoznavanje govora.

S obzirom da se u ovom radu opisuje metoda automatske detekcije naglašenih riječi, upotreba odgovarajućeg softvera je od temeljne važnosti. Za ovo istraživanje bit će upotrebljen program Praat koji je napravljen specifično za proučavanje govora. To je program koji sadrži velik broj modula za analizu zvuka, od kojih mnogi podržavaju grafički prikaz podataka. Za ovo istraživanje potrebno je nekoliko osnovnih stvari: učitavanje snimljenog zvuka u digitalnom obliku, rastavljanje zvuka na segmente, označavanje segmenata slovima (koja predstavljaju

glasove), mogućnost analize pojedinačnih segmenata, te upotreba neuralne mreže čiji ulazni podaci mogu biti rezultati dobiveni iz analize odgovarajućih segmenata zvuka. Praat podržava sve ove mogućnosti, ali i mnoge druge, što je dobro ako se javi potreba za nekim dodatnim tehnikama analize zvuka. Nadalje, s obzirom da će se kao ulazni signal upotrebljavati snimka vijesti s HRTa, prije same analize govora bit će neophodno ulazni signal pripremiti tako da se iz njega izdvoje samo zvuk i podnatpisi. Za to također postoje programi koji rade sa standardnim formatom ovakvih podataka, kao što je TS (engl. Transport Stream), i koji mogu biti upotrebljeni i za ovo istraživanje.

Ovaj rad će biti organiziran u nekoliko dijelova. U II dijelu biti će opisani osnovni pojmovi vezani za akustiku govora: fizikalne karakteristike zvuka, frekvencija, ton, intenzitet, osnovna frekvencija (F0), formanti, te akustičke karakteristike nekih glasova uz grafički prikaz njihovog spektra i ostalih akustičkih svojstava. U ovom će dijelu također biti opisane neke akustičke karakteristike naglašenih riječi po kojima ih se može razlikovati od drugih, nenaglašenih riječi.

U III dijelu biti će prikazane relevantne tehnike analize zvuka: spektar, spektrogram, oscilogram i spektralna analiza, LTAS (engl. Long Term Avarage Spectar) ili usrednjeni spektar, a bit će i prikazani neki postupci u radu sa zvukom koji su važni za njegovu analizu, kao što je označavanje (anotacija), te će biti opisane neke mogućnosti programa Praat koji će biti upotrijebljen za ovaj rad, te primjena upotrebljenih biblioteka za programski jezik Python.

U IV dijelu razmatrat će se osnove strojnog učenja i neuralnih mreža koje se u ovom istraživanju upotrebljavaju za prepoznavanje pojedinačnih glasova. Taj će se dio sastojati od općeg uvoda u strojno učenje i neuralne mreže u kojem će biti opisane njihove prednosti u odnosu na neke druge računalne modele, specifično kod primjene u analizi zvuka, a nakon toga će biti prikazan način na koji će se jedna takva neuralna mreža upotrijebiti u ovom radu. Isto tako, biti će napravljen osnovni pregled područja strojnog učenja, čiji su neuralne mreže jedan od modela.

U V dijelu biti će prikazan detaljni postupak analize snimljenog signala kao mogući način pronalaženja naglašenih riječi. Taj se postupak sastoji od nekoliko aktivnosti koje su ovdje podijeljene u sljedeće korake:

- Označavanje (anotiranje) zvuka, gdje se za svaki izgovoreni glas ručno odredi vremenski interval unutar snimljenog uzorka koji sadrži zvuk tog glasa. To je nužno za treniranje neuralne mreže. Ovo je dugotrajni proces jer jedna snimka od samo nekoliko minuta sadrži na stotine glasova koje treba pažljivo anotirati. Isto tako, određivanje

početka i završetka nekog glasa nije uvijek jednostavno jer glasovi mogu biti izgovoreni djelomično, a mogu se i nadovezivati jedni na druge, pa je u takvim slučajevima teško odrediti gdje prvi glas završava, a drugi počinje.

- Izrada skripta za Praat softver kojima će se iz anotirane (označene) snimke govora spremiti rezultati spektralne analize svakog pojedinog glasa. Taj će se rezultat sastojati od vrijednosti usrednjenog spektra za svaki glas i oznake glasa (slovo, što uključuje i dvoslove *lj* i *nj*) kojem pripadaju. Te vrijednosti dobiju se tako da se prođe kroz zvučni segment svakog anotiranog glasa i za njega se napravi usrednjena spektralna analiza. Ti će podaci nakon obrade biti upotrebljeni za treniranje neuralne mreže. Ovim će se podacima neuralna mreža „učiti“ da prepoznaje razne glasove (foneme) u govoru nekoliko slučajno odabranih govornika.

Nakon što su u prethodnim koracima podaci pripremljeni uslijedio bi postupak treniranja neuralne mreže s tim podacima. Taj bi se postupak sastojao od nekoliko koraka:

1. Eliminacija varijacija u intenzitetu. Za treniranje neuralne mreže bitan je samo spektralni oblik glasa. S obzirom da u govoru intenzitet konstantno varira on može utjecati na efikasnost treniranja time što povećava raznolikost vrijednosti usrednjenog spektra koje neuralna mreža mora obraditi. Da bi se treniranje ubrzalo potrebno je eliminirati što je više moguće varijacije u intenzitetu. Jedan način upotrebljen za ovo istraživanje je taj da se sve vrijednosti usrednjenog spektra podignu ili spuste za onoliko koliko je potrebno da najveća vrijednost ne prelazi neki zadani intenzitet, ali tako da sve ostale vrijednosti ostanu u istom odnosu, to jest da se sačuva prvobitni spektralni oblik.
2. S obzirom da raspon vrijednosti usrednjenog spektra nije unutar intervala [0, 1] ove je vrijednosti potrebno prvo skalirati na taj interval. To se radi zbog toga što računalni programi za upotrebu neuralnih mreža obično rade samo s vrijednostima u tom intervalu.
3. Prethodno skalirane vrijednosti organiziraju se u strukturu podataka u kojoj su vrijednosti usrednjenog spektra zadane zajedno s kategorijom kojoj pripadaju, što je u ovom slučaju slovo za glas. Nakon toga slijedi treniranje neuralne mreže s tim podacima. Cilj ovog treniranja je da se omogući automatsko prepoznavanje pojedinačnih glasova iz novih snimki (onih koje nisu bile upotrebljene za treniranje), čime bi se omogućilo automatiziranje postupka pronalaženja riječi iz podnatpisa.

4. Nakon prethodnog koraka rezultat bi bio neuralna mreža trenirana za prepoznavanje pojedinačnih fonema u govoru.

U sljedećoj fazi slijedi postupak detekcije naglašenih riječi. Prije toga, međutim, biti će napravljeno izdvajanje podnatpisa iz snimke tako da oni pored teksta sadrže i informaciju o tome unutar kojeg vremenskog intervala se pojavljuju na toj snimci. To je važno da bi se na osnovu prepoznatih glasova moglo automatski zaključiti kojoj riječi oni pripadaju. Na primjer, ako su unutar nekog segmenta zvuka prepoznati glasovi "d..ava", a tekst s riječju "država" se pojavljuje u okolini vremenskog intervala tih glasova onda je vjerojatno da se ti glasovi odnose na tu riječ. Tada bi uslijedila analiza osnovne frekvencije i intenziteta tih prepoznatih glasova pomoću koje bi se ustanovilo je li ta riječ bila naglašena.

Nakon što je neuralna mreža trenirana detekcija naglašenih riječi sastoji se od sljedećih koraka:

1. Klasifikacija glasova iz neke snimke s „novim“ govornikom, odnosno govornikom čiji govor nije bio upotrebljen za treniranje. Za klasifikaciju glasova upotrebljene su dvije tehnike: Kod prve se snimka podijeli na segmente od 10 milisekundi, za svaki se taj segment izračuna usrednjeni spektar koji se spremi u datoteku. Kod druge se tehnike spektar izračunava samo na dijelovima snimke kod kojih su prisutni glotalni pulsevi, odnosno kod kojih postoji zvuk. Kada se dobije pozicija pulsa odredi se segment od 5 milisekundi prije i poslije pulsa, ukupno 10 milisekundi, te se za taj segment izračuna spektar i podaci spreme u datoteku. Ova druga tehnika daje korisne informacije jer ona osigurava da se neki važni dio zvuka ne preskoči (što se može desiti kod prve tehnike gdje se cijela snimka podijeli na segmente od 10 milisekundi).
2. Rezultat prethodnog koraka je niz glasova koji su rezultat klasifikacije gore spomenutih segmenata od 10 milisekundi. Taj će niz glasova sadržavati glas i vrijeme u kojem se pojavljuje na snimci. Neki će od tih glasova biti ispravno prepoznati, ali će biti i dosta onih koji nisu ispravno prepoznati. Primjerice, umjesto glasa *m* izgovorenog na snimci mreža može taj glas prepoznati kao *v* ili neki drugi glas. Da bi se ustanovilo koje su riječi bile naglašene potrebno je ustanoviti gdje na snimci neka riječ počinje i završava. Očito je da što je više glasova neuralna mreža ispravno klasificirala to će biti lakše utvrditi na osnovu tih glasova o kojim se riječima radi. Međutim, s obzirom da velik broj glasova neće biti ispravno klasificiran potrebno je tekst onoga što je izgovoreno na neki način poravnati s glasovima koje je dala neuralna mreža. To će se riješiti

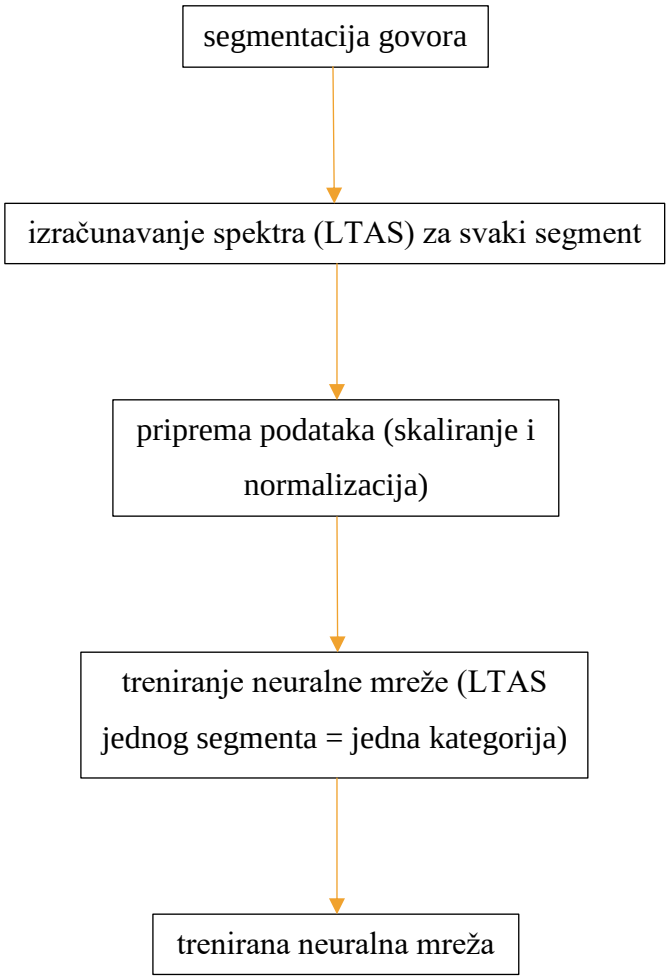
upotrebom algoritma za poravnanje koji će tekst onoga što je izgovoreno (niz glasova) pokušati poravnati s nizom glasova koji su rezultat klasifikacije.

3. Rezultat poravnanja iz prethodnog koraka dati će informacije o tome gdje (otprilike) na snimci počinje i završava neka riječ iz podnatpisa (teksta). Sada je taj segment zvuka potrebno analizirati sa stajališta osnovne frekvencije, intenziteta i trajanja, što će dati informacije o tome je li ta riječ naglašena.

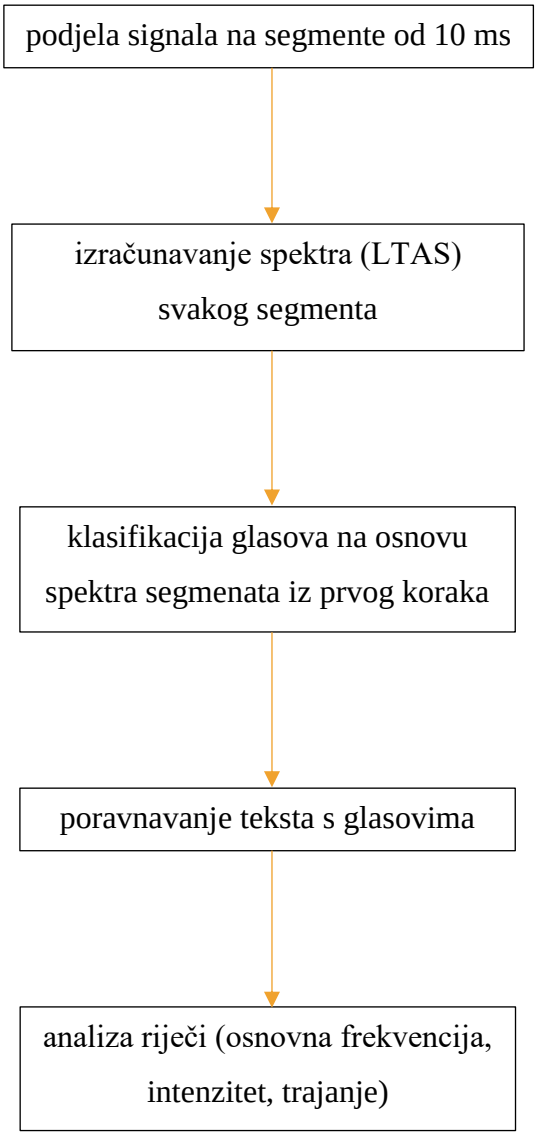
Većina prethodno opisanih koraka podrazumijeva izradu Praatovih i Pythonovih modula koji će to omogućiti, što uključuje module za testiranje i analizu rezultata. Na slikama 1 i 2 gornji je postupak prikazan shematski.

U VI dijelu biti će prikazani rezultati na primjerima novih govornih snimki (s podnatpisima), to jest onih koje nisu bile upotrebljene za treniranje neuralne mreže. Ove bi snimke uključivale nekoliko govornika gdje bi se vidjelo kako opisana metoda funkcionira s drugačijim okruženjima od onih u probnim uzorcima (sa stajališta brzine govora, intonacije, visine glasa, načinom izgovora i sl.). Isto tako, biti će prikazani rezultati poravnavanja teksta s glasovima gdje će biti izmjerena preciznost upotrebljenog algoritma sa stajališta odstupanja od točnih pozicija riječi na snimci.

VII poglavlje sadržavat će zaključno razmatranje. Ovdje će biti opisane prednosti i nedostaci ove u odnosu na druge metode analize naglašavanja riječi. Također će biti opisane neke moguće varijante na ovu metodu, a na kraju će se razmatrati moguća poboljšanja. Pored toga, bit će istaknuta važnost stvaranja većeg korpusa anotiranog govora na hrvatskom jeziku kao preduvjet za mnoga korisna istraživanja iz područja informacijsko komunikacijskih znanosti i fonetike. S obzirom da je automatsko prepoznavanje glasova hrvatskog jezika preduvjet za razna istraživanja temeljena na govoru (kao što je detekcija emocija, identifikacija govornika, analiza raznih prozodijskih obilježja i mnoga druga), ovakav bi korpus uvelike olakšao ovakva i slična istraživanja jer bi poslužio kao jedan od alata potrebnih za njihovo izvođenje.



Slika 1: Postupak treniranja neuralne mreže.



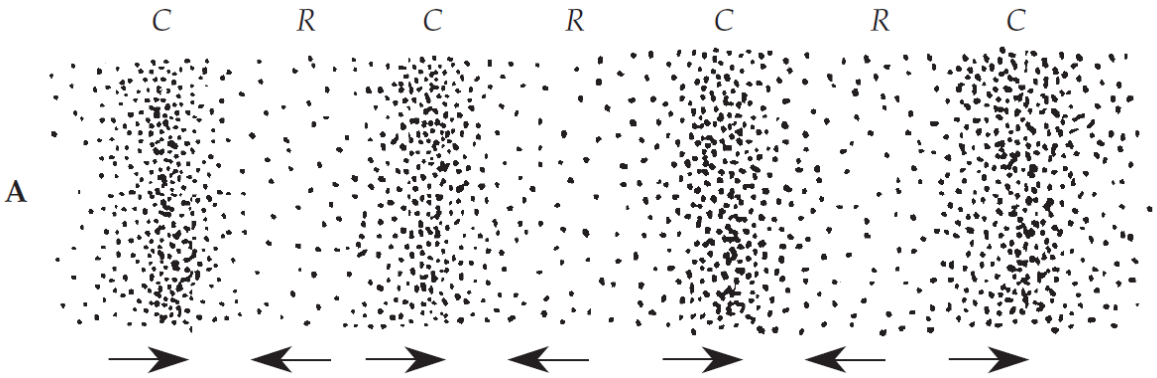
Slika 2: Postupak detekcije naglašenih riječi.

2. Osnove akustike govora

Akustika govora pripada području fonetike, specifično akustičke fonetike, čiji je cilj proučavanje zvuka govora (Bakran, 1996). Ovdje se pod zvukom govora podrazumijeva samo onaj zvuk kojim se materijalizira jezik, ali ne i ostali zvukovi koje čovjek svojim govornim traktom može proizvesti. Sa stajališta ovog istraživanja pored temeljnih fizikalnih karakteristika zvuka kao što su frekvencija, ton, intenzitet i osnovna frekvencija, važnu ulogu igraju i formanti, te akustičke karakteristike pojedinačnih glasova.

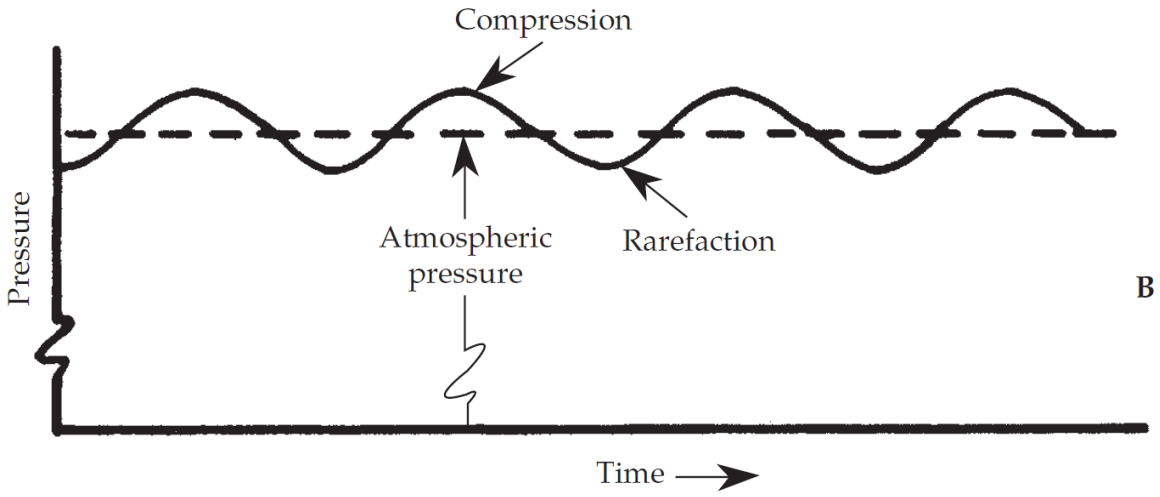
2.1 Fizikalne karakteristike zvuka

Neformalno, zvuk se može definirati kao posljedica varijacija tlaka u nekom elastičnom mediju, kao što su zrak, voda i razne krute tvari (Hansen, 2016). Zvuk se širi u valovima na način da se elastični medij (zrak) zgušnjava i razrijeđuje, što stvara razlike u tlaku zraka, kako pokazuje slika 3:



Slika 3: Promjene u tlaku zraka širenjem zvučnih valova (F. A. Everest & K. C. Pohlmann, 2009).

Ovdje C označava zgušnjavanje (engl. *compression*), a R razrijeđivanje (engl. *rarefaction*). Strelice označavaju smijer pomicanja molekula zraka u određenim fazama širenja zvučnog vala. Ova se pojava može prikazati i kao na slici 4.



Slika 4: Promjene tlaka u zraku širenjem zvučnog vala (F. A. Everest & K. C. Pohlmann, 2009).

Brzina zvuka u zraku je 343 m/s. Što je medij gušći to je brzina zvuka veća. Na primjer, brzina zvuka kroz željezo je oko 5.500 m/s. Brzina zvuka označava brzinu kojom se zvučna energija giba kroz medij.

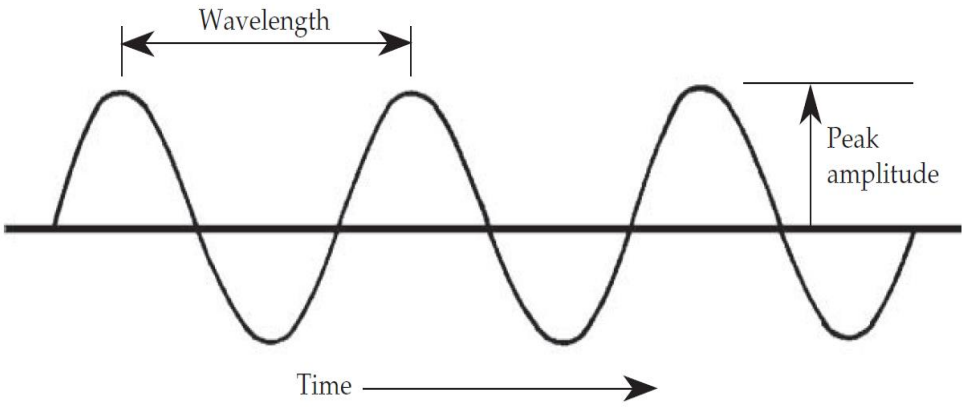
2.2 Frekvencija

Zvučni se valovi najčešće promatraju i proučavaju kao sinusoida. Na slici 5 prikazan je čisti ton. Osnovni oblik ovakvog zvučnog vala opisan je formulom

$$y(t) = A \sin(2\pi ft + \varphi)$$

gdje je

- A = amplituda
- f = frekvencija
- φ = pomak ili *faza* izražena u radijanima što pokazuje gdje je početak ciklusa



Slika 5: Zvuk prikazan sinusoidom (F. A. Everest & K. C. Pohlmann, 2009).

Jedna od najosnovnijih karakteristika sinusoide je *periodičnost*, to jest uzastopno ponavljanje jednog te istog valnog oblika. Jedan *ciklus* je jedno ponavljanje tog valnog oblika, a *period* je vrijeme potrebno da se izvrši jedan ciklus. Za analizu govora najčešća mjera za period je milisekunda, što je 1/1000 sekunde.

Frekvencija nekog zvuka je broj ciklusa u sekundi i izražava se u Hertzima (Hz). Na primjer, ako je na slici 5 prikazan vremenski interval od jedne sekunde, onda se radi o zvuku frekvencije od 3 Hz, a ako je prikazan interval od 1/10 sekunde onda je frekvencija 30 Hz. Odnos frekvencije i perioda određen je formulom

$$f = 1/t$$

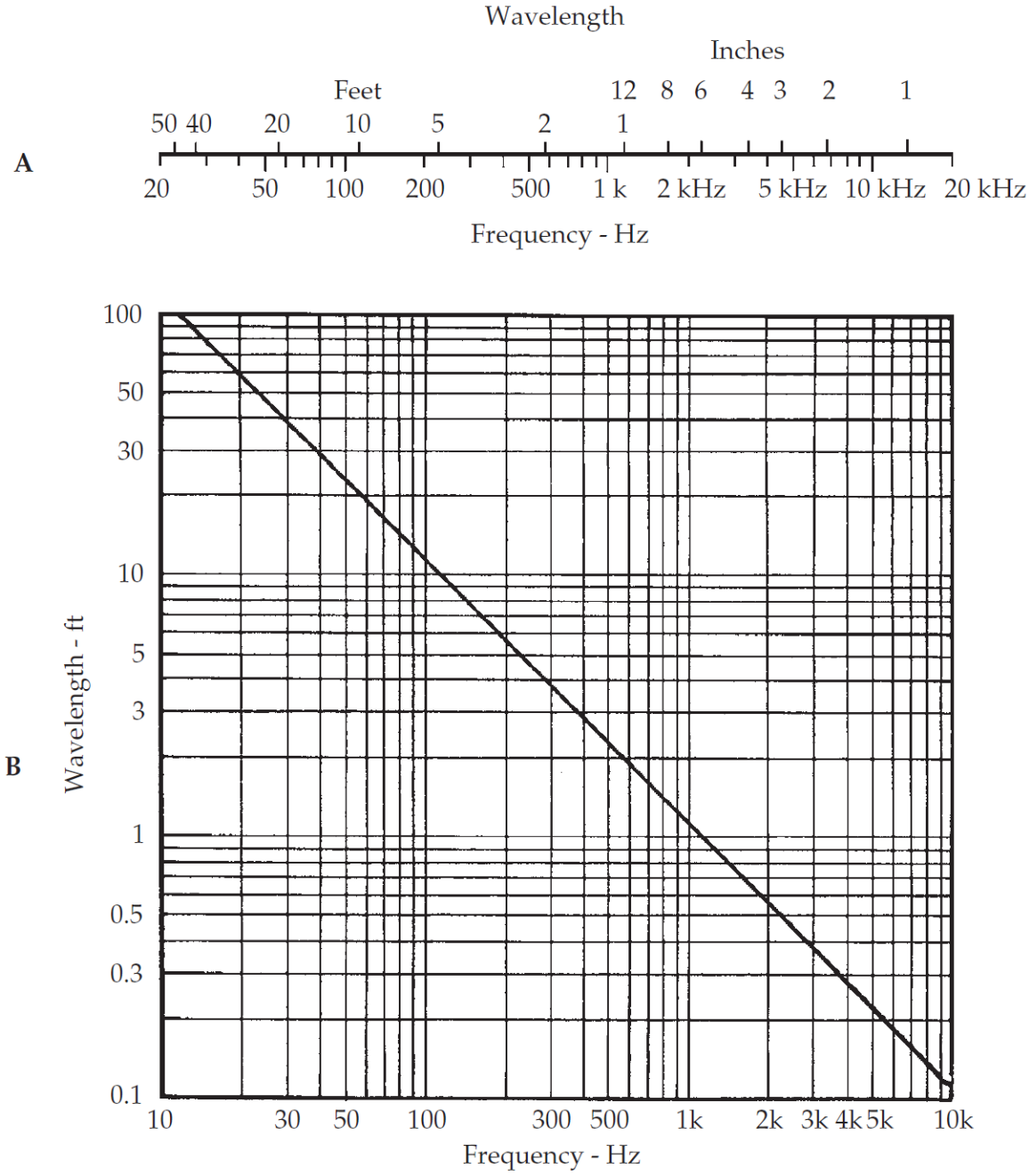
gdje je *f* frekvencija, a *t* period u sekundama.

Amplituda određuje jačinu, to jest intenzitet zvuka. Ako promatramo sliku 3 onda visina vrha svakog perioda pokazuje njegovu amplitudu – što je ta visina veća, intenzitet je jači.

Valna duljina je razmak između vrhova (ili bilo koje dvije točke na istim pozicijama) dvaju uzastopnih ciklusa. Jedno od najvažnijih pravila u akustici je odnos između valne duljine i frekvencije:

$$valna\ duljina = \frac{\text{brzina zvuka}}{\text{frekvencija}}$$

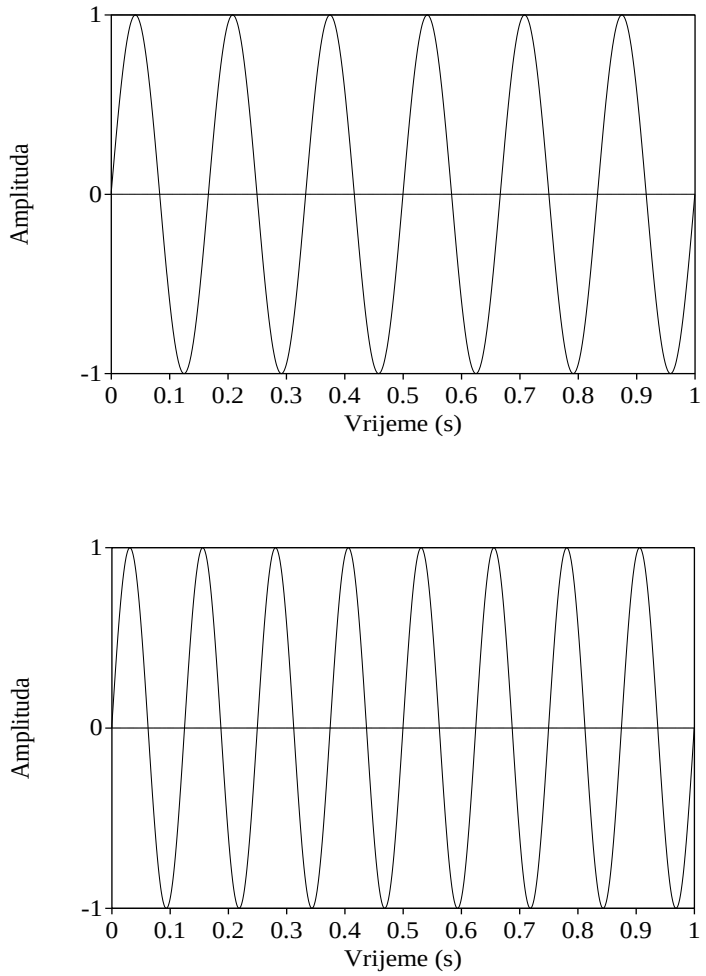
Ovaj se odnos može grafički prikazati kao na slici 6.



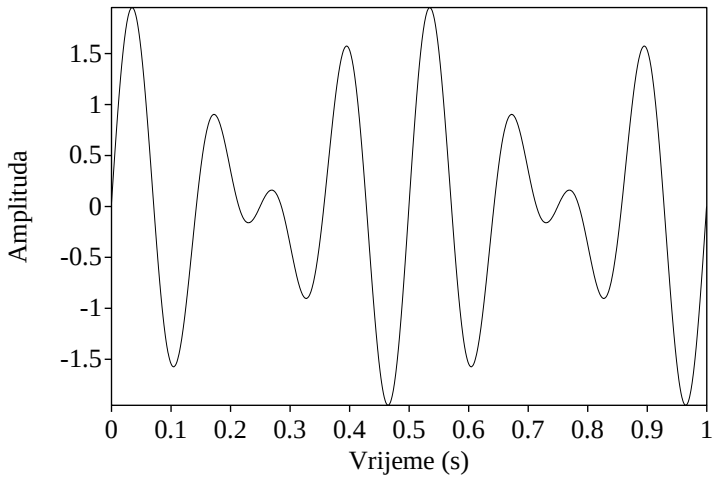
Slika 6: Inverzan odnos između frekvencije i valne duljine. U dijelu A je skala koja aproksimira taj odnos, a u dijelu B je detaljniji graf (F. A. Everest & K. C. Pohlmann, 2009).

2.3 Kompleksni tonovi

Zvuk kao onaj prikazan na slici 5 naziva se *čistim tonom* zato jer sa sastoji od samo jedne frekvencije, to jest od samo jedne sinusoide. Takvi tonovi su, međutim, gotovo nepostojeći u prirodi, što uključuje i govorne zvukove. Za razliku od čistih tonova, *kompleksni tonovi* se sastoje od najmanje dvije sinusoide. Na slici 7 vide se dva čista tona frekvencije 6 Hz i 8 Hz. Ako bi ove dvije sinusoide označili s f_1 i f_2 onda bi $f_1 + f_2$ bio kompleksan ton prikazan na slici 8. Taj se ton sastoji od dvije frekvencije: 6 Hz i 8 Hz, istih amplituda. Na taj se način može formulirati bilo kakav kompleksan ton koji se sastoji od bilo kojeg broja sinusoida. Zvuk govora se, na primjer, sastoji od puno većeg broja frekvencija od kojih nije svaka zastupljena jednakim intenzitetom. Ta će nam činjenica biti važna kada dođemo do postupaka analize zvuka.



Slika 7: Dva zvučna vala od 6 Hz (gore) i 8 Hz (dole).



Slika 8: Kompleksan ton sastavljen od dva tona sa slike 7.

2.4 Periodičnost i osnovna frekvencija (F_0)

Na slici 8 vidi se kompleksan ton sastavljen od dva čista tona frekvencije 6 i 8 Hz. Period tih dvaju tonova je 1/6s (za onaj od 6 Hz) i 1/8s (za onaj od 8 Hz), što je lako vidjeti jer se radi o sinusoidama, pa je potrebno samo prebrojati cikluse. Kod kompleksnih tonova kao što je onaj na slici 8, međutim, jedan period je teže uočiti jer se on sastoji od većeg broja drugih perioda koji su rezultat činjenice da sa taj ton sastoji od zbroja dviju ili više sinusoida. Ton na slici 8 sastoji se od dva kompleksna ciklusa, gdje prvi završava, a drugi počinje točno na sredini tog prikaza (vidi se da su prva i druga polovina te krivulje jednake). S obzirom da je prikazani vremenski raspon tog tona 1 sekunda, onda je njegov *osnovni period* duljine 0.5 sekunde. Drugim riječima, osnovni period je vrijeme potrebno da se izvrši jedan ciklus kompleksnog tona. Iz ovoga proizlazi pojam *osnovne frekvencije*, što je frekvencija koja se dobije iz osnovnog perioda po istoj formuli:

$$f_0 = 1/t_0$$

gdje je f_0 osnovna frekvencija, a t_0 osnovni period. Prema tome, osnovna frekvencija tona sa slike 6 je 2 Hz jer je $f_0 = 1/1/2 = 2$. Osnovna frekvencija se može izračunati i tako da se nađe najveći zajednički dijelitelj sastavnih frekvencija. U gornjem primjeru, najveći zajednički dijelitelj (frekvencija) 6 i 8 je 2.

2.5 Intenzitet zvuka i zvučni tlak

Za analizu govora je, pored frekvencije, važan i intenzitet. Intenzitet zvuka okvirno odgovara onome što se percipira kao glasnoća ili jačina zvuka. Za analizu zvuka najčešće se upotrebljava decibel kao jedinica intenziteta zvuka. Decibel je definiran kao logaritam odnosa između nekog zvučnog intenziteta i referentnog intenziteta (to je intenzitet na pragu čujnosti). Formula za intenzitet zvuka u decibelima je

$$dB_{IL} = 10\ log_{10} \frac{I}{I_{ref}}$$

Oznaka dB_{IL} označava da se radi o razini intenziteta (engl. *Intensity Level*). Referentni intenzitet se uzima da je 10^{-12} W/m^2. To je intenzitet zvuka od 1000 Hz koji se smatra da je na pragu čujnosti za prosječnu osobu. Tablica 1 prikazuje neke tipične zvukove i njihove intenzitete. Prema gornjoj formuli odnos između mjerenog i referentnog intenziteta za normalan razgovor je 1.000.000, a onaj za buku u prometu je 1.000.000.000. U prikazane su vrijednosti odnosa između mjerenog i referentnog intenziteta. Da bi se pojednostavnio rad s intenzitetima vrijednost decibela se dobije tako da se odnos I_m/I_{ref} prikaže u eksponencijalnom

obliku, te se od njega uzme samo eksponent pomnožen s 10[1]. S obzirom da je intenzitet zvuka u praksi teško mjeriti (s nekakvim uređajem), umjesto intenziteta zvuka češće se uzima u obzir *zvučni tlak*. Na primjer, jedan tipičan uređaj koji je osjetljiv na zvučni tlak je mikrofon. Zvučni tlak se izražava u mikropascalima (µPa), gdje je prag čujnosti na 20 µPa. Kao i kod decibela intenziteta, zvučni tlak se također izražava pomoću slične logaritamske skale. Formula za zvučni tlak je

$$dB_{SPL} = 20\ log_{10} \frac{p}{p_{ref}}$$

gdje je p mjereni, a p_{ref} referentni zvučni tlak od 20 µPa. Oznaka SPL označava zvučni tlak (engl. *Sound Pressure Level*). Formule za dB_{IL} i dB_{SPL} su ekvivalentne u smislu da će dati iste vrijednosti za isti intenzitet zvuka zato jer je intenzitet proporcionalan tlaku na kvadrat. Razlika je samo u tome što kod dB_{IL} uzimamo u obzir odnos I_m/I_{ref}, a kod dB_{SPL} odnos p/p_{ref}. Na primjer, za zvuk intenziteta koji je 1000 puta iznad praga čujnosti dobili bi 30 dB_{IL} jer je 10 * $\log_{10}1000$ = 30. S obzirom da je intenzitet proporcionalan tlaku na kvadrat, taj bi omjer bio 31.6 (što je korijen od 1000), pa bi po formuli za dB_{SPL} dobili 20 * $\log_{10}31.6$ = 30 dB_{SPL}.

Slika 9 prikazuje govorni signal u *vremenskoj domeni*, to jest promjene zvučnog tlaka kroz vrijeme. Na *x*-koordinati je vrijeme, a na *y*-koordinati zvučni tlak. Na gornjem dijelu slike vrhovi (ili zubci) signala nastaju zbog porasta zvučnog tlaka u tom trenutku, dok je na mjestima gdje je signal gotovo gladak zvučni tlak slab. Na donjem dijelu slike prikazana je krivulja promjena zvučnog tlaka, gdje se vidi da je ona višlja tamo gdje su skokovi veći na gornjem dijelu, a niža tamo gdje su manji.

Važno je imati u vidu to da 0 dB SPL ne znači da nema zvuka, nego da je zvučni tlak na razini referentnog zvučnog tlaka. Isto tako, zvuk jačine -20 dB SPL označava zvučni tlak koji je 10 puta slabiji od referentnog zvučnog tlaka, to jest 2 µPa.

[1] Ako se samo uzme eksponent (bez da ga se pomnoži s 10) onda bi dobili mjernu jedinicu *bell*.

Tablica 1: Tipični zvukovi i njihovi intenziteti (Hillenbrand, 2016).

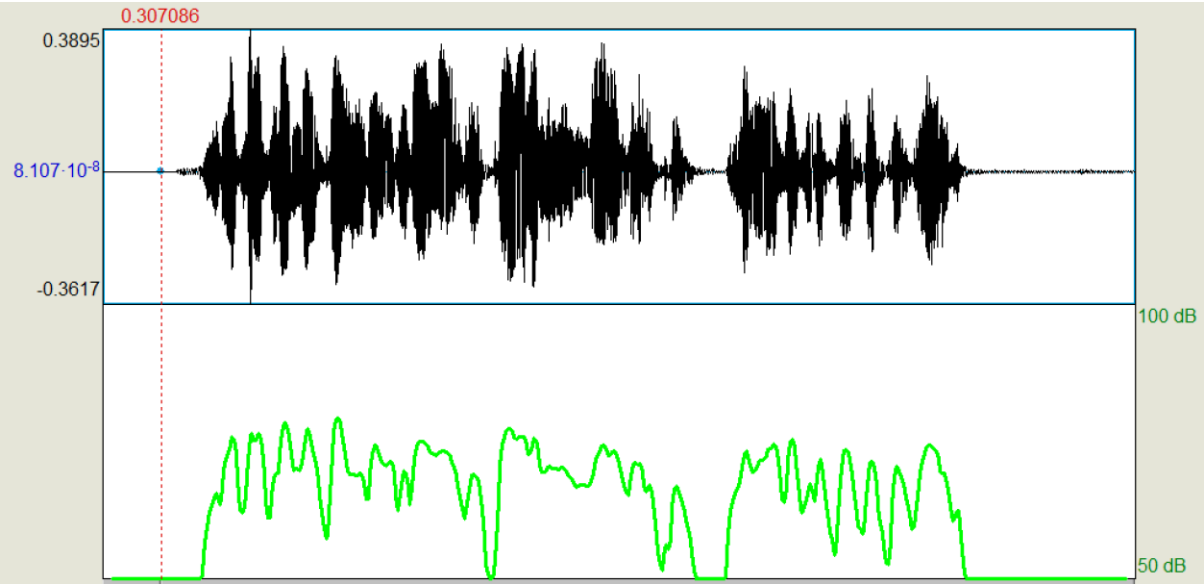
Zvuk	**Intenzitet**
Prag na 1000 Hz	10^{-12} W/m^2
Šapat	10^{-8} W/m^2
Normalan razgovor	10^{-6} W/m^2
Buka u prometu	10^{-4} W/m^2
Rok-koncert	10^{-2} W/m^2
Mlazni motor	10^{0} W/m^2

Tablica 2: Odnos između mjerenog i referentnog intenziteta za tipične zvukove (Hillenbrand, 2016).

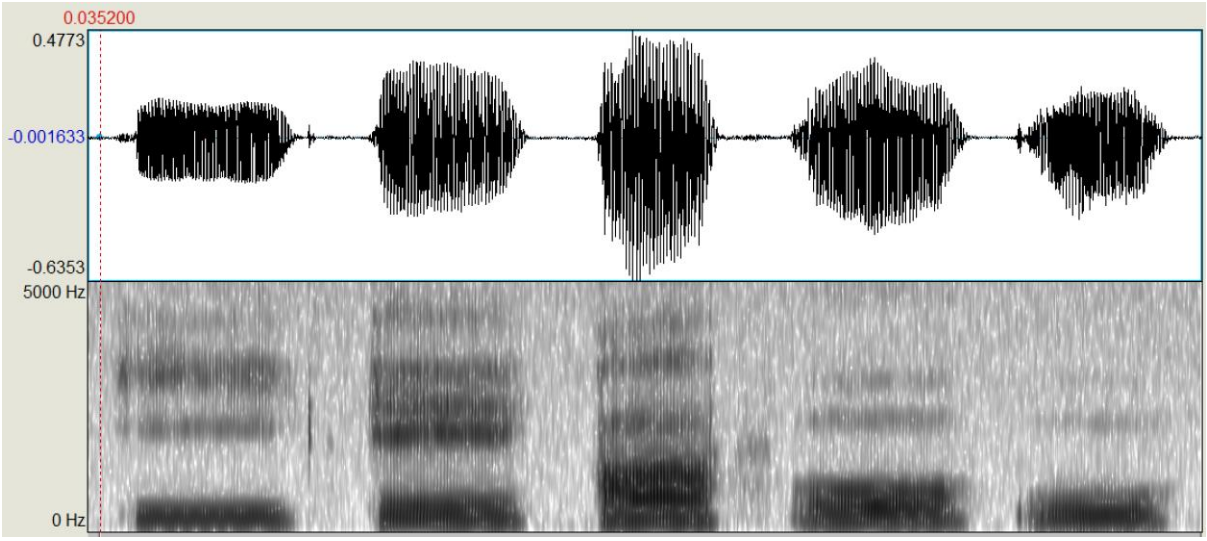
Zvuk	**Odnos I_m/I_{ref}**	**Eksponencijalni oblik**	**Decibel**
Prag na 1000 Hz	1	10^0	0
Šapat	10.000	10^4	40
Normalan razgovor	1.000.000	10^6	60
Buka u prometu	100.000.000	10^8	80
Rok-koncert	10.000.000.000	10^{10}	100
Mlazni motor	1.000.000.000.000	10^{12}	120

2.6 Spektar

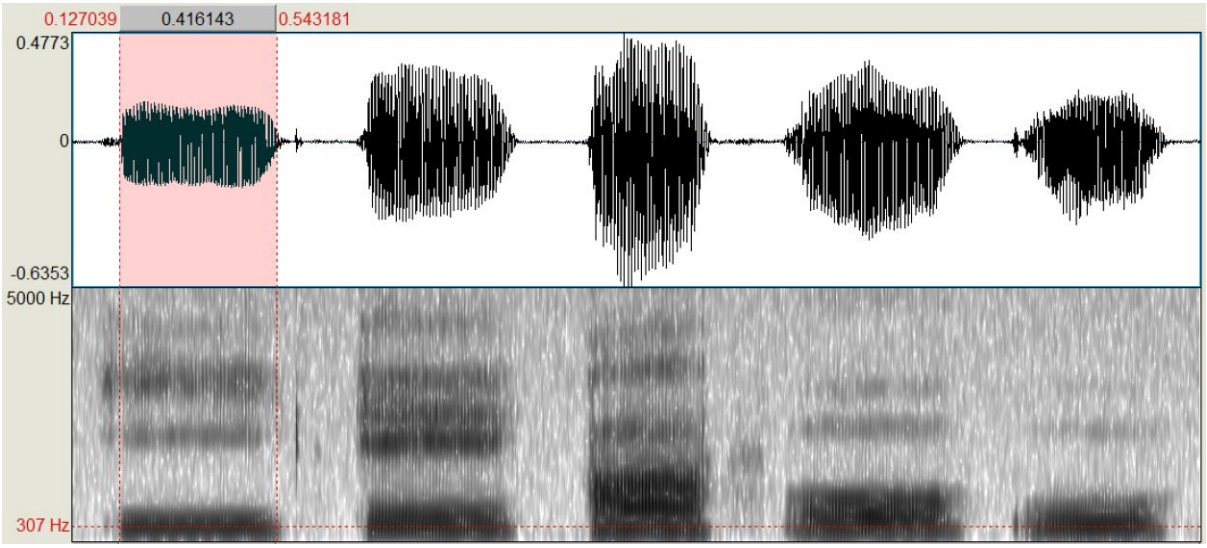
Na slici 9 vide se izgovorene rečenice prikazane u vremenskoj domeni. U takvom se prikazu ne vide frekvencije od kojih je taj zvuk sačinjen, nego samo promjene zvučnog tlaka, to jest intenzitet. Da bi se došlo do sastavnih frekvencija nekog zvuka potrebno je dobiti njegov *spektar* – niz frekvencija od kojih je taj zvuk sastavljen gdje se vidi kojim je intenzitetom svaka od njih zastupljena. Na slici 10 (gore) vide se izgovoreni vokali *i*, *e*, *a*, *o* i *u* prikazani u vremenskoj domeni. Na donjem dijelu te slike vidi se *spektrogram* koji pokazuje zastupljenost frekvencija u analiziranom zvuku. Na ordinati su frekvencije, a na apscisi vrijeme. Dijelovi koji su zacrnjeni pokazuju pojačani intenzitet na tom frekvencijskom rasponu. Spektar zvuka se može prikazati i kao na slikama 11 i 12 gdje se jasno vide istaknute frekvencije i njihovi intenziteti. Na slici 12 (stranica 21) prva skupina vrhova se nalazi na frekvencijskom rasponu koji je vidljiv kao prvo zacrnjenje (promatrajući odozdo) na slici 10. Na slici 12 nakon toga slijede još dva vrha koji odgovaraju drugom i trećem zacrnjenju sa slike 10.



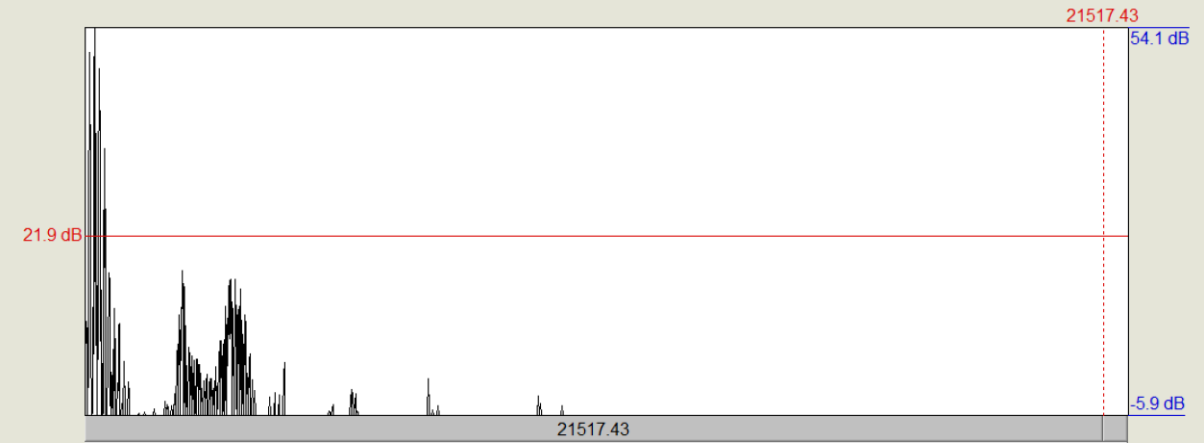
Slika 9: Govorni signal u vremenskoj domeni.



Slika 10: Spektrogram vokala i, e, a, o *i* u.



Slika 11: Spektrogram glasa „i“.



Slika 12: Spektar glasa „i“.

2.7 Akustičke karakteristike glasova

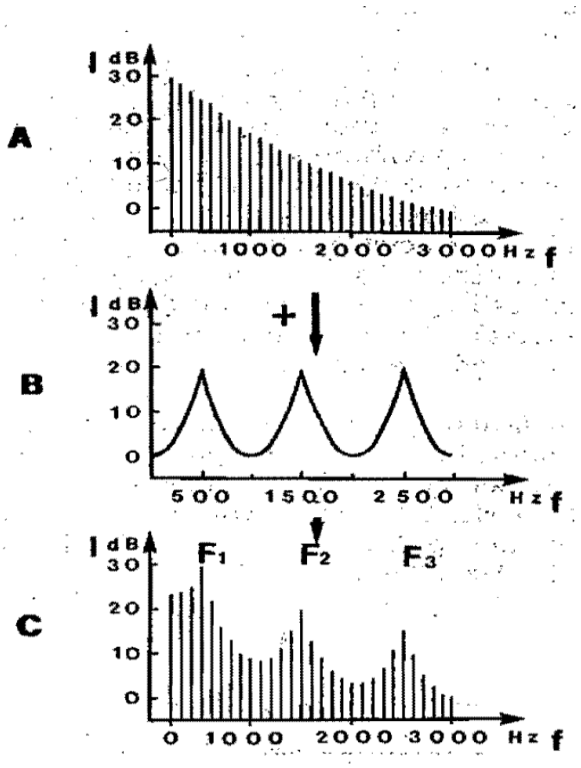
Pod pojmom *glasa* podrazumijevamo najmanju govornu jedinicu koja se zove *fonem*. Fonemi su glasovi kao što su vokali *a, e, i, o* i *u,* nazali *m* i *n,* te mnogi drugi. Glasovi su jedan od ključnih elemenata za ovo istraživanje jer će se pomoću njih utvrđivati prepoznavanje i moguća naglašenost riječi i njena pozicija u podnatpisu. Jedna vrsta glasova koja ovdje ima centralnu ulogu su vokali jer se većina aspekata naglašenosti očituje isključivo na vokalima, što će kasnije biti detaljnije analizirano.

Prema (Babić, i dr., 1991) postoji šest zvučnih osobina koje tvore govorne zvukove:

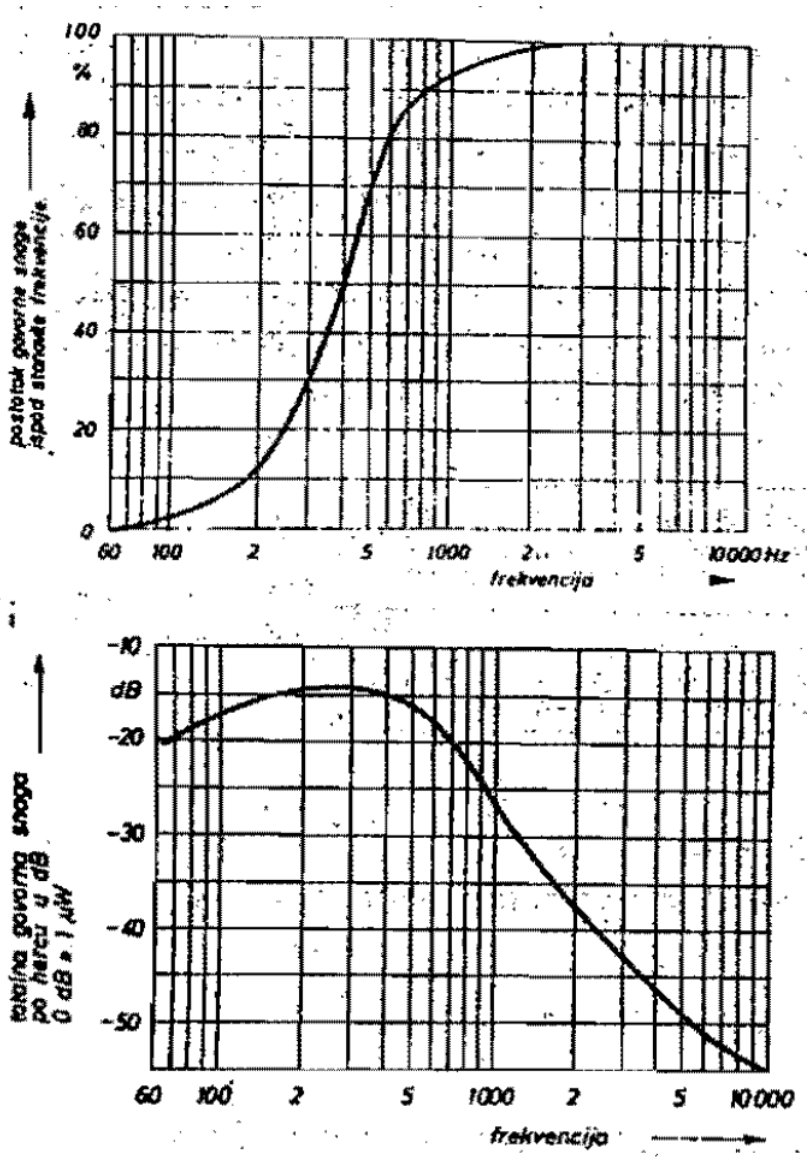
1. Spektralni oblik (boja)
2. Spektralni sastav (šuman-harmoničan)
3. Promjene zvuka u vremenu
4. Trajanje
5. Jakost (glasnoća)
6. Učestalost periodičnih titraja (ton)

2.7.1 Spektralni oblik glasova

Za ovo istraživanje spektralni oblik jedna je od najvažnijih akustičkih osobina ljudskog glasa. On nastaje zbog oblika ljudskog govornog trakta koji ima specifične rezonantne karakteristike. Prema (Babić, i dr., 1991), u prosječnom su govoru najistaknutije frekvencije oko 300 Hz, pretežna količina zvuka nalazi se na frekvencijama ispod 1000 Hz, a oko 50% ukupne zvučne snage nalazi se na frekvencijama ispod 400 Hz. Ovo je prikazano na slici 12. S obzirom da frekvencijska karakteristika govornog prolaza ime više rezonantnih vrhova i antirezonantnih prigušenja, spektralni oblik govornog zvuka imati će više vrhova, ali takvih da je svaki viši vrh (na višoj frekvenciji) slabiji zbog padajućeg oblika prvotnog zvuka (slika 13).



Slika 13: Zbrajanje prvog i dodatnog izvor zvuka. A – grkljanski zvuk, B – frekvencijska karakteristika govornog prolaza dužine 17 cm pri izgovoru neutralnog samoglasnika, C – zvuk na izlazu (preuzeto iz (Babić, i dr., 1991), str. 179)



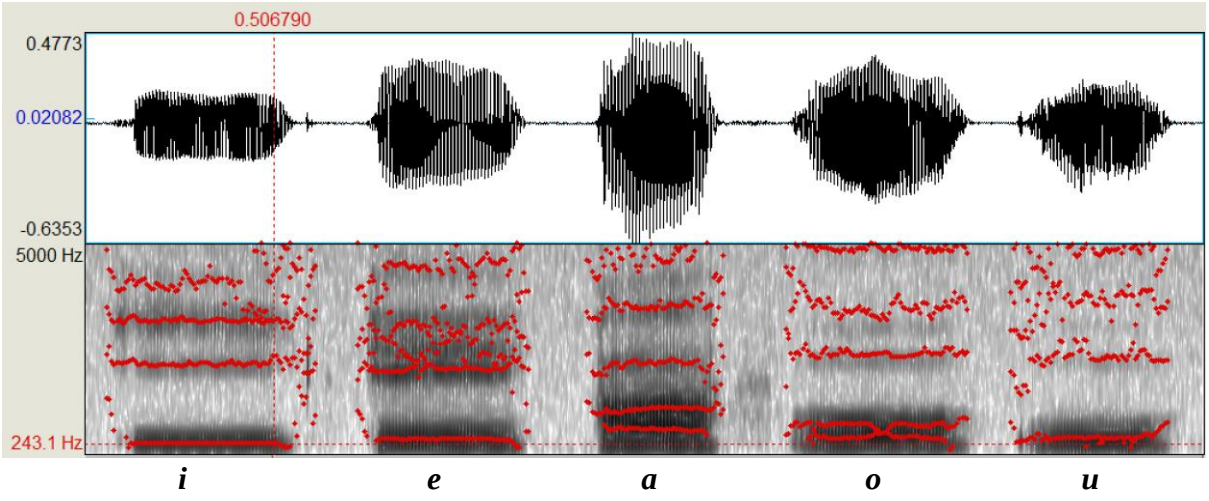
Slika 14: Prosječni spektralni oblik čovječjeg govora. Na gornjoj slici je prosječni spektralni oblik govora, a na donjoj postotak zvučne snage koju sadrži govorni zvuk u spektru ispod određene frekvencije (preuzeto iz (Babić, i dr., 1991), str. 178).

2.7.1.1 Formanti

Formanti su pojačanja zvučnog intenziteta na određenim frekvencijskim područjima. Oblik vokalnog trakta utječe na proizvedeni spektar zvuka na dva načina (Hawkins, 2016):

1. On određuje prisutnost supralaringalnog izvora zvuka.
2. On određuje frekvencije formanata.

Frekvencijsko područje na kojem se očituje takvo pojačanje intenziteta zvuka zavisi od formanata. Za prvi formant (F1) frekvencije se kreću između 200 i 900 Hz, za drugi formant (F2) taj je raspon od 600 do 2400 Hz, a za treći (F3) od 900 do 2800 Hz (Bakran, 1996). Na slici 15 prikazani su spektrogrami za vokale. Crvenom bojom označeni su formanti. Na primjer, za vokal *i* vidi se da je prvi formant otprilike na 240 Hz, dok su drugi i treći formanti negdje na 2200 i 3000 Hz. Na ovakvom prikazu formanti se očituju kao istaknuta zacrnjenja spektrograma na specifičnim frekvencijskim intervalima jer je na tim frekvencijama pojačan intenzitet.



Slika 15: Spektar vokala „i“, „e“, „a“, „o“ i „u“.

Formanti su vidljivi i na slici 12 na kojoj frekvencije prva tri lokalna maksimuma odgovaraju frekvencijama prva tri formanta.

U sljedećem dijelu nalazi se opis spektralnih oblika svih vrsta glasova u hrvatskom jeziku prema (Babić, i dr., 1991). Spektralni oblici glasova pokazuju zvučnu raznolikost govornog zvuka za koju je potrebno trenirati sustav za prepoznavanje glasova, pa je stoga korisno dati jedan kratki prikaz akustičkih osobina glasova.

2.7.1.2 Spektralni oblici samoglasnika

Jedna od ključnih grupa glasova za ovo ostraživanje su samoglasnici. Oni su važni jer se na njima očituju mnoge prozodijske osobine govora, a jedna od njih je i naglašavanje riječi. Druga osobina samoglasnika važna za ovo istraživanje (što su pokazali i rezultati) je ta da u prosjeku traju duže od ostalih glasova (prema (Bakran, 1996)), posebno kada služe isticanju prozodijskih osobina, pa su zbog toga lakše uočljivi za strojno učenje i prepoznavanje. Shematizirani spektralni oblik samoglasnika prikazan je na slici 16.

Samoglasnika ima malo (šest, u hrvatskom jeziku, uključujući i samoglasnički *r*). Ali s obzirom da se oni mogu izgovoriti na velik broj načina, u praksi njihov broj je praktički beskonačan. Prema (Bakran, 1996) frekvencijski raspon za funkcioniranje vokala kreće između 250 i 3000 Hz, ako se u obzir uzmu frekvencije formanata F1 i F2.

U (Bakran, 1996) prikazane su prosječne frekvencije formanata hrvatskog standardnog govora za tri vrste govornika: odrasle muškarce (u dobi 20 do 80 godina), žene (u dobi 20 do 54 godine) i djece (u dobi 7 do 11 godina). Te su frekvencije prikazane u tablicama 3, 4 i 5.

Tablica 3: Prosječne frekvencije F1, F2 i F3 odraslih muških govornika hrvatskog standardnog govora.

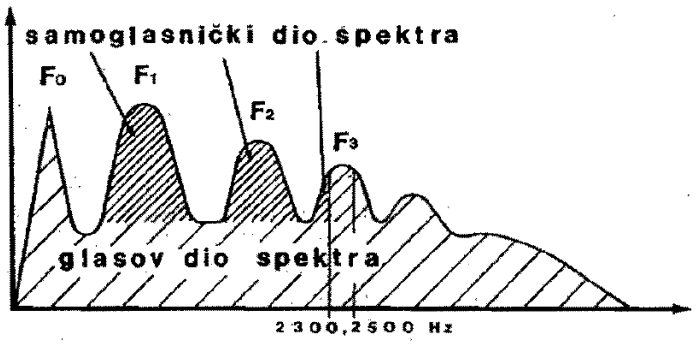
	i	e	a	o	u
F1	282	471	664	482	324
F2	2192	1848	1183	850	717
F3	2713	2456	2433	2472	2544

Tablica 4: Prosječne frekvencije F1, F2 i F3 odraslih ženskih govornika hrvatskog standardnog govora.

	i	e	a	o	u
F1	302	493	884	576	353
F2	2623	2360	1393	980	758
F3	3246	2930	2709	2776	2764

Tablica 5: Prosječne frekvencije F1, F2 i F3 djece, govornika hrvatskog standardnog govora.

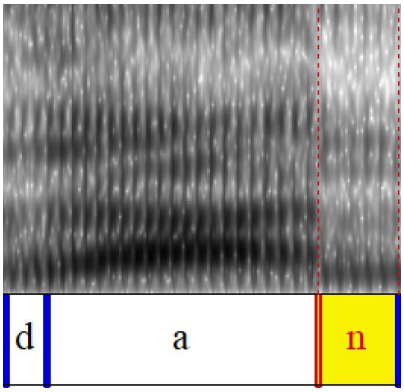
	i	e	a	o	u
F1	375	500	984	585	463
F2	3033	2569	1581	1095	962
F3	3487	3255	3024	2173	3224



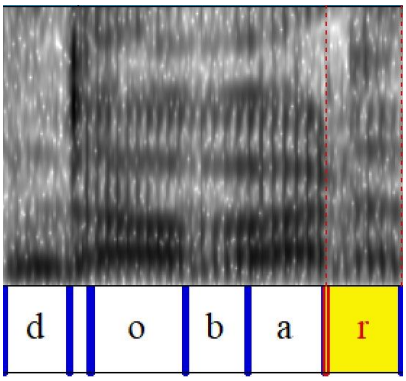
Slika 16: Shematizirani prikaz samoglasničkog i glasovog dijela spektra (preuzeto iz (Babić, i dr., 1991), str. 183).

2.7.1.3 Spektralni oblici zvonkih suglasnika

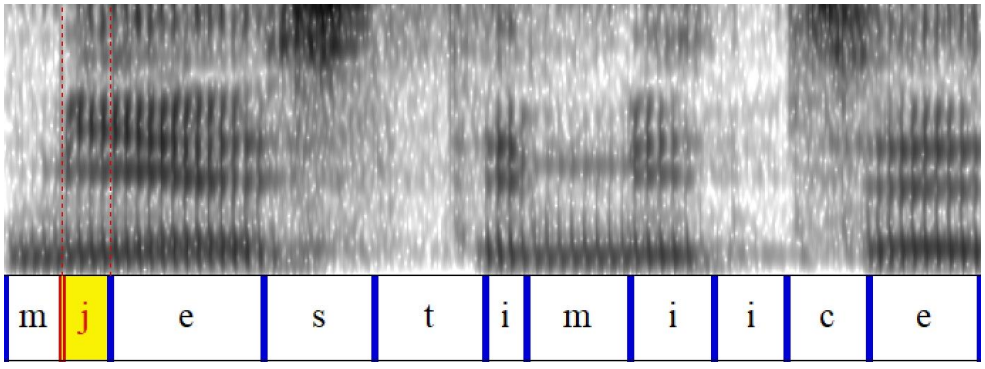
Zvonki su suglasnici *j, v, r, l, m, n, nj* i *lj*. I takvi suglasnici, kao i samoglasnici, imaju formantski oblik spektra jer je i njihov zvuk rezultat rezonantnog oblikovanja periodičnog zvuka kojeg stvaraju glasnice. Međutim, njihov je zvuk relativno slabiji nego onaj u samoglasnika. Jedna karakteristika zvonkih suglasnika bitna za ovo istraživanje je da oni služe kao okidači i prekidači slogova, a slogovi imaju prozodijsku ulogu, pa zbog toga ovi glasovi imaju važnost i u prozodiji. Umjesto izoliranih glasova, slike 17, 18, 19, 20, 21, 22, 23 i 24 pokazuju spektralne oblike zvonkih suglasnika u kontekstu izgovorenih riječi.



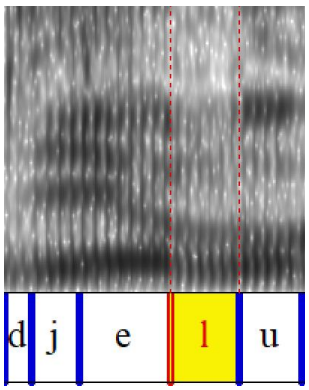
Slika 17: Spektrogram glasa 'n' izgovorenog u riječi 'dan'.



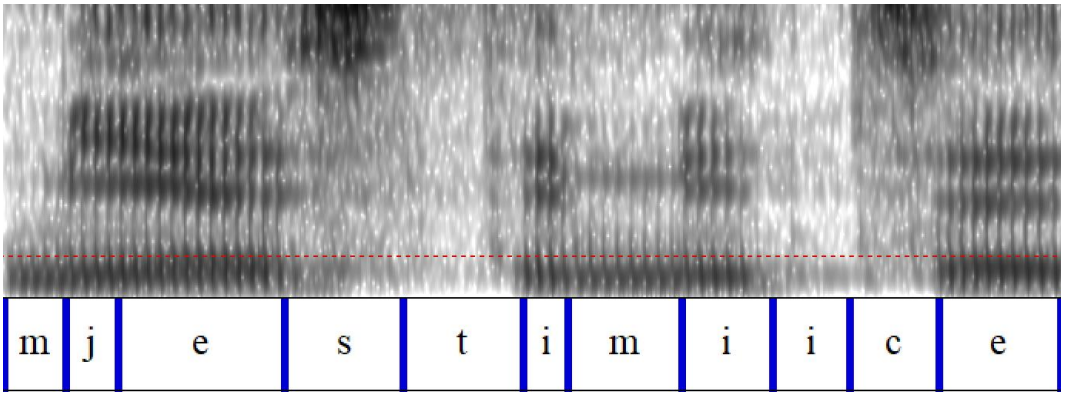
Slika 18: Spektrogram glasa 'r' izgovorenog u riječi 'dobar'.



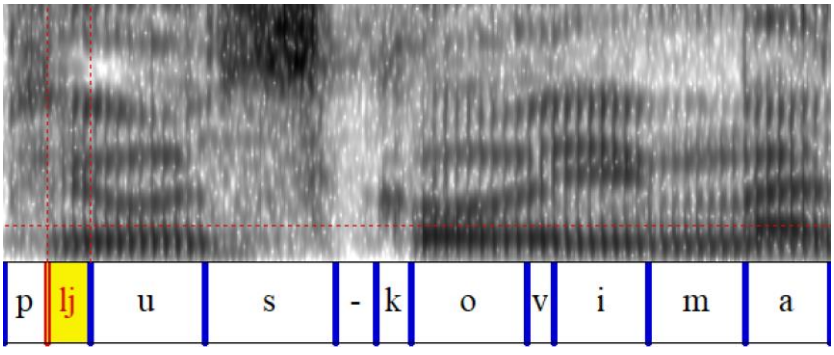
Slika 19: Spektrogram glasa 'j' izgovorenog u riječi 'mjestimice'.



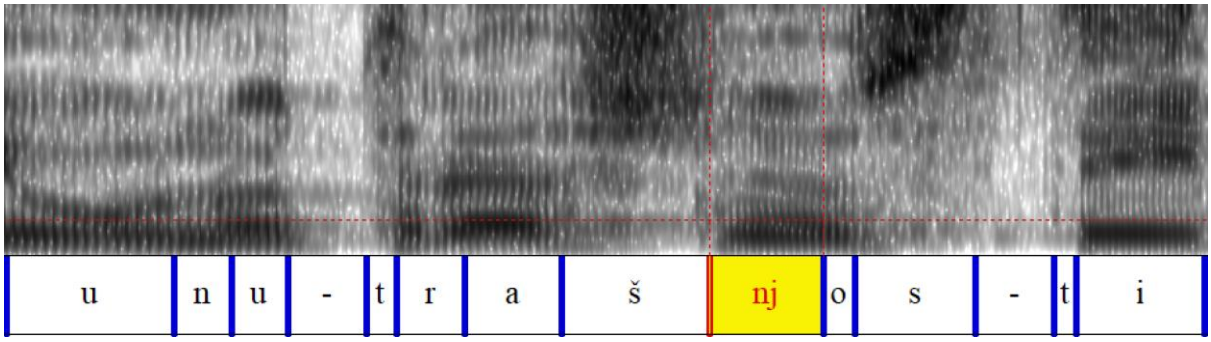
Slika 20: Spektrogram glasa 'l' izgovorenog u riječi 'djelu'.



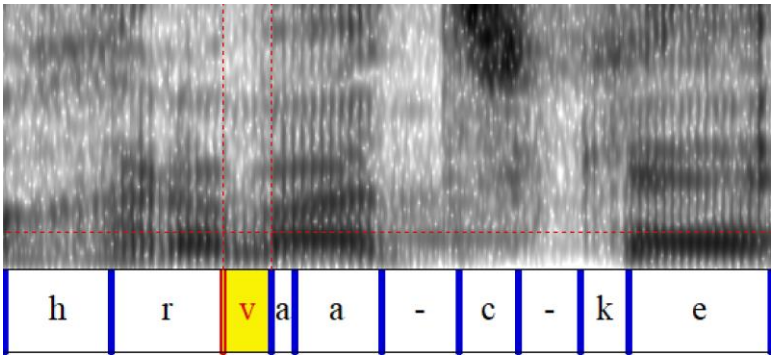
Slika 21: Spektrogram glasa 'm' izgovorenog u riječi 'mjestimice'.



Slika 22: Spektrogram glasa 'lj' izgovorenog u riječi 'pljuskovima'.



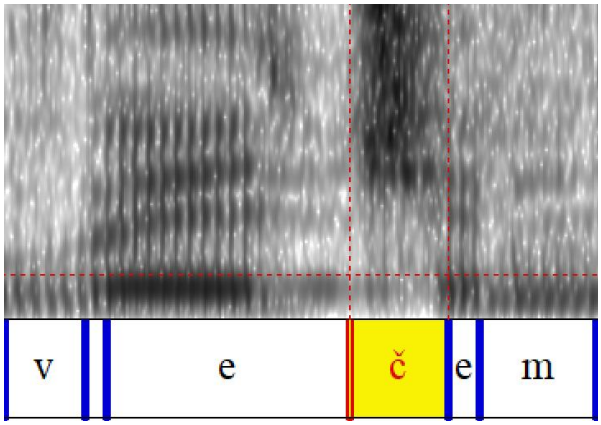
Slika 23: Spektrogram glasa 'nj' izgovorenog u riječi 'unutrašnjosti'.



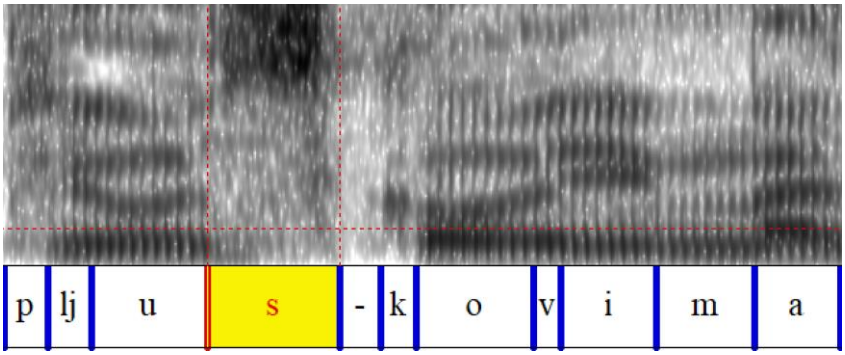
Slika 24: Spektrogram glasa 'v' izgovorenog u riječi 'hrvatske'.

2.7.1.4 Spektralni oblik tjesnačnih i poluzatvornih suglasnika

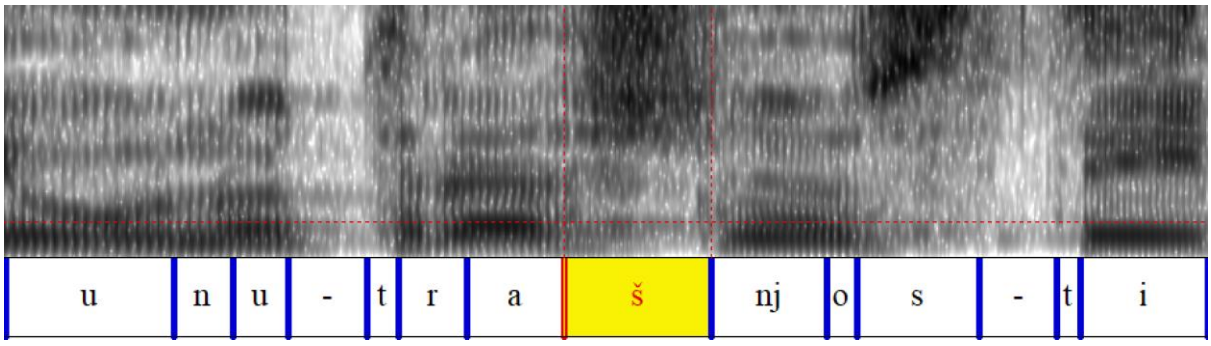
Tjesnačni suglasnici koji su važni za ovo istraživanje su *f, s, z* i *š*, a poluzatvorni *c, č* i *dž*. Ovi suglasnici imaju sličan spektralni oblik. Oni nastaju vrtloženjem zračne struje u tjesnacu na mjestu izgovora, pa je njihov zvuk u stvari šum u frekventnom rasponu između 3000 Hz i 6000 Hz i gotovo uopće ne zavisi od rezonatora u šupljinama iza mjesta izgovora. Na spektrogramima ovih suglasnika se zbog toga neće uočavati zupci formanata, nego široko na spektru raspršena zvučna energija.



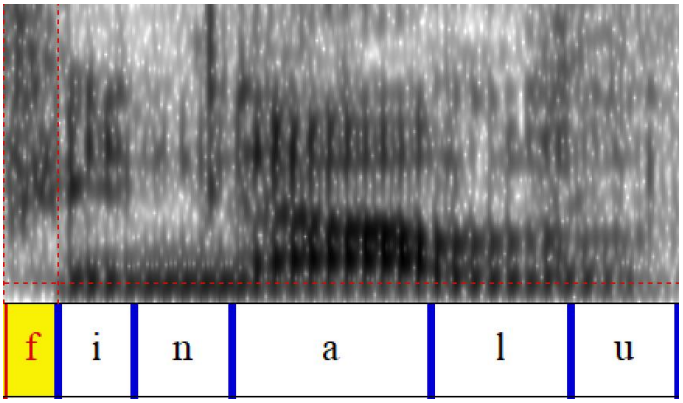
Slika 25: Spektrogram glasa „č“ izgovorenog u riječi „većem“. Prostor između „e“i „č“ na kojem nema zacrnjenja je okluzija koja spada pod „č“.



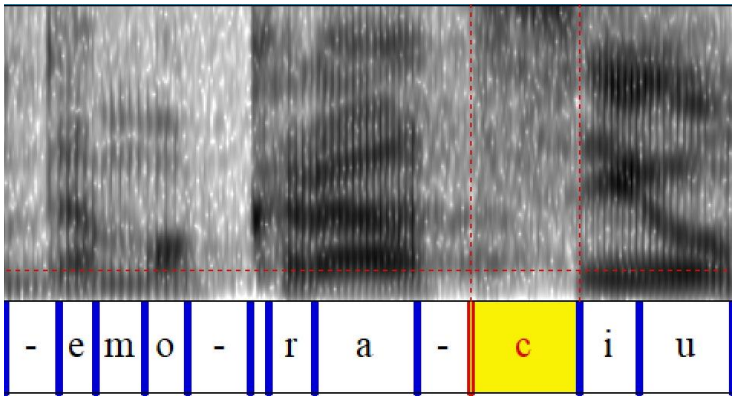
Slika 26: Spektrogram glasa „s“ izgovorenog u riječi „pljuskovima“. Prostor označen s „-„ je okluzija koja spada pod „k“.



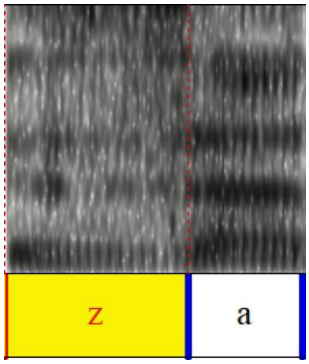
Slika 27: Spektrogram glasa „š“ izgovorenog u riječi „unutrašnjosti“. Prostor označen s „-„ je okluzija koja spada pod „t“.



Slika 28: Spektrogram glasa „f“ izgovorenog u riječi „finalu“.



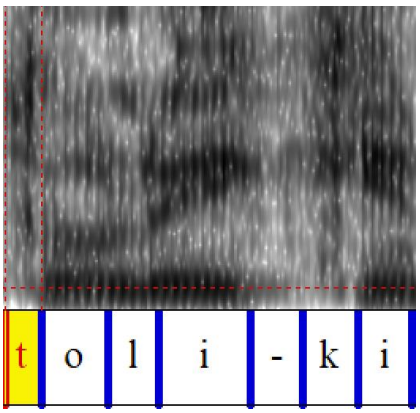
Slika 29: Spektrogram glasa „c“ izgovorenog u riječi „komemoraciju“. Prostor označen s „-„ispred „c“ je okluzija koja spada pod „c“. Ostali dijelovi označeni s „-„ su prekidi koji nemaju zvuka.



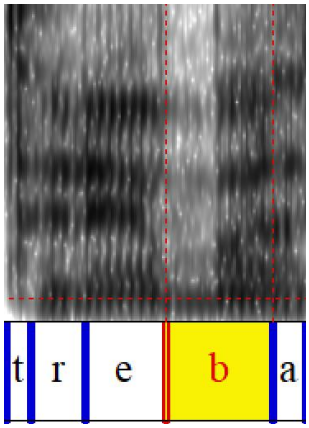
Slika 30: Spektrogram glasa „z“ izgovorenog u riječi „za“.

2.7.1.5 Spektralni oblik zatvornih suglasnika

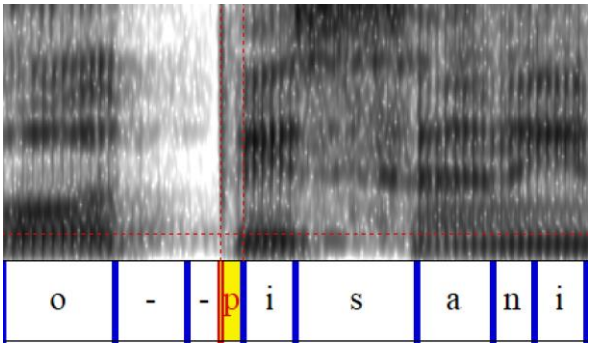
Zatvorni suglasnici (okluzivi) jesu *p, t, k, b, d* i *g*. Spektralni oblik ovih samoglasnika nije jednostavno opisati jer u središnjoj fazi njihova izgovora oni uopće nemaju zvuka ili je zvuk znatno prigušen. Okluzije mogu biti zvučne ili bezvučne. Zvuk okluziva uglavnom dolazi s ruba njihovog izgovora, ali na taj zvuk često utječu okolni glasovi, tako da je spektar ovih glasova određen okolnim glasovima i zvukom kratkotrajnog šuma eksplozije. Na slikama 31, 32, 33, 34 i 35 prikazani su spektrogrami ovih suglasnika.



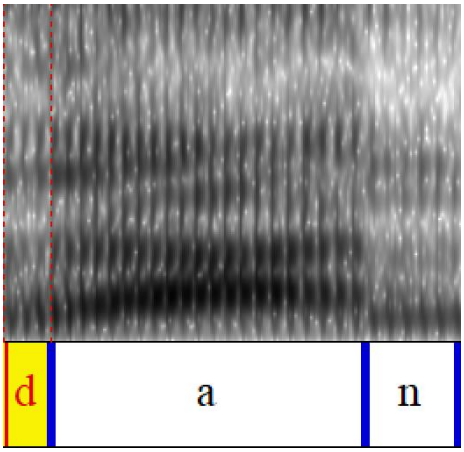
Slika 31: Spektrogram glasova „t“ i „k“ izgovorenih u riječi „toliki“.



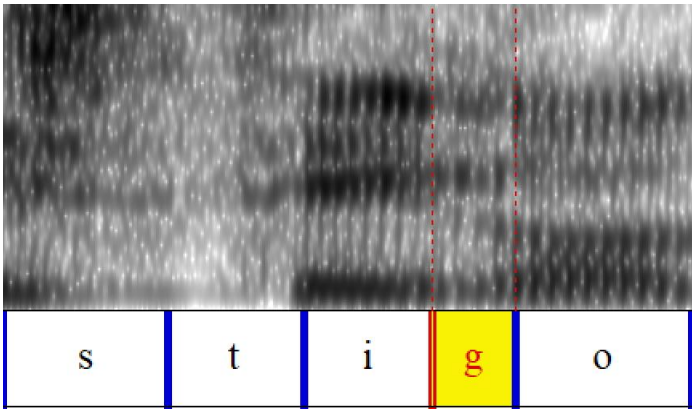
Slika 32: Spektrogram glasa „b“ izgovorenog u riječi „treba“.



Slika 33: Spektrogram glasa „p“ izgovorenog u riječi „opisani“.



Slika 34: Spektrogram glasa „d“ izgovorenog u riječi „dan“.



Slika 35: Spektrogram glasa „g“ izgovorenog u riječi „stig'o“.

Iz gornjih prikaza može se primijetiti da mnogi glasovi imaju slične spektralne oblike. Ispod su prikazane grupe glasova sličnog spektralnog oblika.

- i, j
- u, v
- s, z, c
- š, ž, č
- p, b
- t, d
- k, g

2.8 Akustičke karakteristike naglašenih riječi

2.8.1 Prozodijska sredstva

Naglašavanje riječi spada u prozodijska sredstva govora, a ostvaruje se isticanjem jednog sloga u riječi. Prema (Babić, i dr., 1991) prozodijska sredstva uključuju sljedeće:

- Ton i intonacija
- Glasnoća i naglasak
- Boja glasa
- Spektralni sastav
- Stanke
- Govorna brzina
- Ritam
- Govorne modulacije
- Način izgovora glasnika
- Mimika i gesta
- Znakovi u glasu

2.8.1.1 Ton

Ton određuje osnovna frekvencija (F_0) i jedan je od elemenata naglašavanja riječi. Intonaciju tvore uzastopne promjene tona. Intonacija se može mjeriti raznim uređajima, kao što je sonograf, i digitalnim pokazivačima. Na slici 36 plavom bojom (crta pri dnu slike) prikazane su promjene tona u rečenici „*Odgovorit ću im javnom šutnjom*“. Govorna se intonacija opisuje na sljedeće načine:

- Prema smjeru tonske promjene
- Prema veličini zamaha
- Prema vremenu kada se zbiva
- Prema tonu oko kojeg se zbiva

Smjer tonske primjene može biti

- Ravan
- Uzlazni
- Silazni
- Silazno-uzlazni
- Uzlazno-silazni

- Silazno-uzlazno-silazni
- i drugi

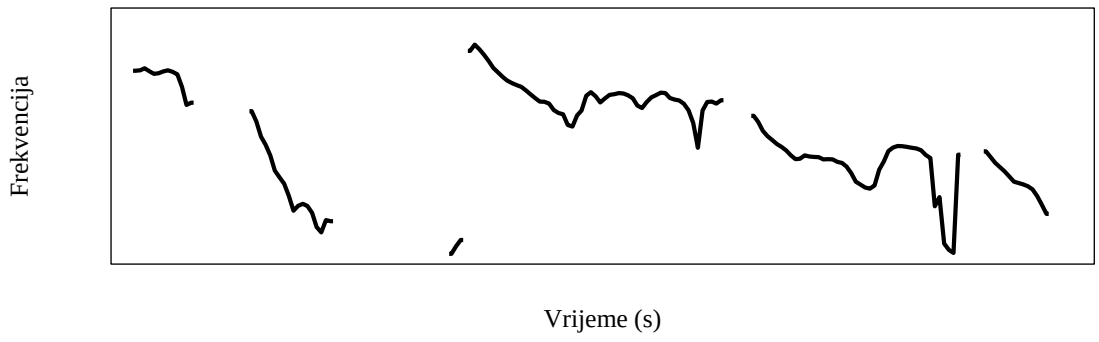
Veličina zamaha može biti

- Mali
- Srednji
- Velik

Ton oko kojeg se tonska promjena zbiva može biti

- Vrlo niska
- Niska
- Središnja
- Visoka
- Vrlo visoka

Na razini fonema intonacija se očituje uglavnom na samoglasnicima (iako je moguća i kod zvonkih i zvučnih suglasnika, ali je tu nevažna).

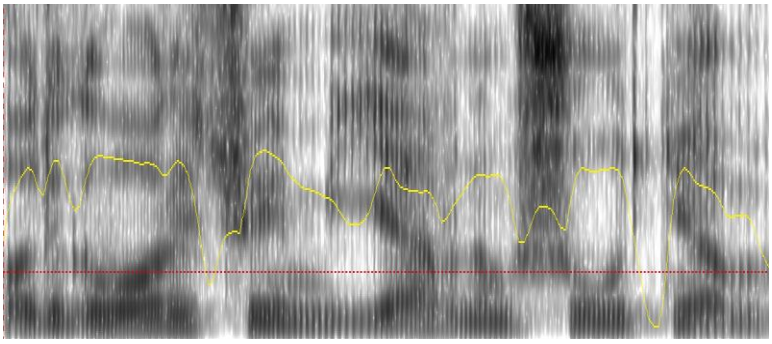


Slika 36: Prikaz kretanja tona u Praatu.

2.8.1.2 Glasnoća i naglasak

Glasnoća i naglasak su povezani s tonom i intonacijom jer često govornu glasnoću prati i kretanje tona. Kao i ton i intonaciju, glasnoću je danas lako mjeriti digitalnim pomagalima. Na slici 37 prikazan je primjer kretanja glasnoće (intenziteta) u uzgovoru rečenice „*Odgovorit ću im javnom šutnjom*“. Kao i kod tona, promjenu glasnoće prati način govora, pa je tako u glasnijem govoru veća govorna snaga, s većim zamasima slogova, jačim izgovorom suglasnika, duljim i otvorenijim suglasnicima i sporijom govornom brzinom.

Glasnoća se govora mjeri u decibelima. Prema (Babić, i dr., 1991) glasnoća šaptanja je na jedan metar od usta 35 dB SPL, dok je u uobičajenom razgovoru oko 65 dB SPL, a pri urlanju oko 100 dB SPL.



Slika 37: Prikaz kretanja glasnoće u Praatu (vijugava linija) u izgovoru rečenice „Odgovorit ću im javnom šutnjom“.

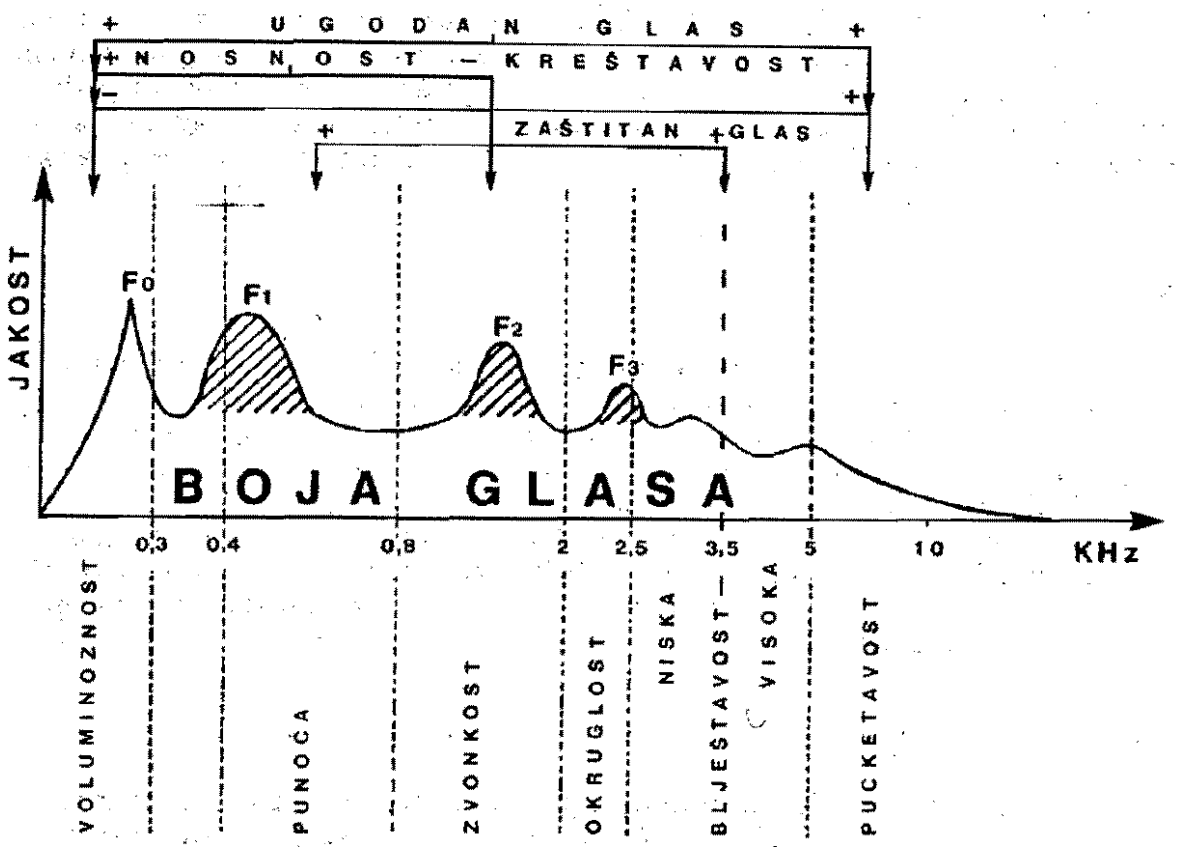
Raspoznatljivost govora ovisi i od jakosti govornog zvuka – što je jakost veća to je raspoznatljivost bolja. Za povezan govor potrebno je da glasnoća bude između 20 dB SPL i 25 dB SPL da bi on slušatelju bio u potpunosti raspoznatljiv. Na slici 39 (stranica 44) prikazane su razine prepoznatljivosti jednosložnih riječi u odnosu na njihovu glasnoću (A) i utjecaj buke na prepoznatljivost govora (B).

2.8.1.3 Boja glasa

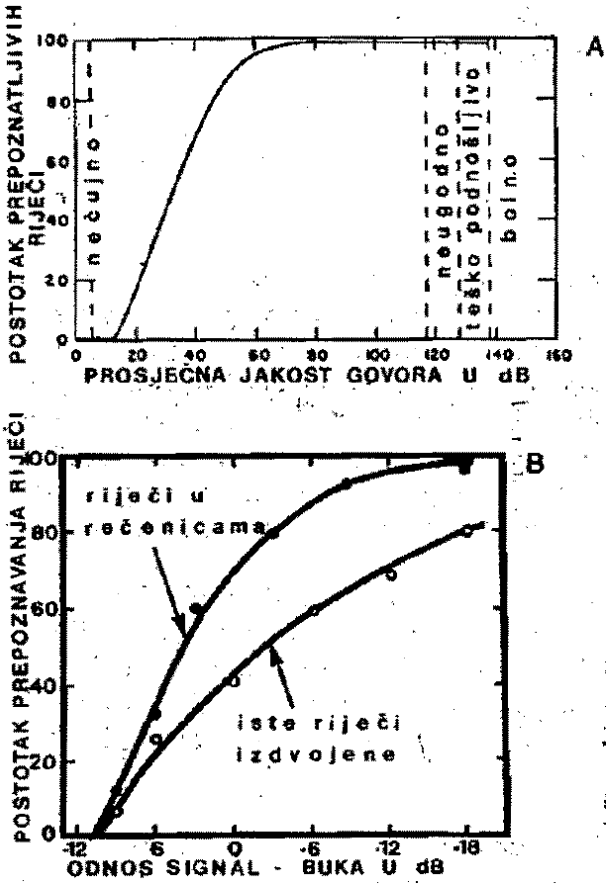
Boja govornog zvuka određena je prosječnim oblikom spektra. Pod bojom se glasa u užem smislu podrazumijeva boja samoglasnika, što se zove vokalna boja. Najvažnije osobine boja glasa su sljedeće:

- Voluminoznost
- Punoća
- Zvonkost
- Okruglost
- Blještavost
- Pucketavost
- Nosnost
- Kreštavost
- Zaštitni glas
- Ugodan glas

Na slici 38 prikazani su spektralni korelati ovih osobina boje glasa.



Slika 38: Pretežni spektralni korelati temeljnih kvaliteta boje glasa ((Babić, i dr., 1991)).



Slika 39: A – krivulja prepoznatljivosti jednosložnih riječi u ovisnosti o njihovoj zvučnoj jakosti; B – utjecaj buke na prepoznatljivost govora te konteksta riječi na smanjenje negativnog utjecaja buke ((Babić, i dr., 1991), str. 289)

2.8.1.4 Spektralni sastav

Govorni je zvuk po spektralnom sastavu harmoničan i šuman. Harmoničan zvuk uglavnom sadrže samoglasnici, a šuman suglasnici, ponajviše stridentni bezvučni suglasnici. Odnos trajanja samoglasnika naspram suglasnika je prozodijsko sredstvo.

2.8.1.5 Stanke

Stanke su važno prozodijsko sredstvo. Govorne stanke su odsječci govornog vremena bez teksta. One se razlikuju od šutnje u kojoj vrijeme nije govorno. Govorna stanka može trajati najviše dvije minute, u protivnom smatra se da je došlo do prekida govora. U spikerskom čitanju vijesti na stanke otpada oko 15% ukupnog trajanja govora, dok u spontanom govoru na stanke otpada oko 40 do 50% vremena. Stanke se mogu podijeliti na pet uloga:

- Stanke razgraničenja
- Stanke isticanja
- Leksičke stanke
- Stanke procesiranja
- Stanke prekida govora

2.8.1.6 Govorna brzina

Govorna brzina ili tempo govora izražava se brojem glasnika, slogova, riječi ili rečenica u jedinici vremena. Brzina govora najčešće se izražava brojem slogova u sekundi, pri čemu se razlikue brzina govora (tempo govora) koja uključuje i stanke od brzine izgovora (tempo artikulacije) koja je brzina govora bez stanka. Prema (Babić, i dr., 1991) normalna je brzina govora 4 do 7 slogova u sekundi, dok je brzina izmjerena na zagrebačkim TV dnevnicima 6,3 sloga u sekundi, što je blizu gornje granice normale brzine govora. Povećanje govorne brzine izvodi se skračivanjem i smanjenjem broja stanka te skračivanjem faze držanja glasnika, uglavnom nenaglašenih i nezavršnih samoglasnika. Usporavanjem se povećava broj i duljina stanka, a glasnici se produljuju (opet, uglavnom samoglasnici).

2.8.1.7 Ritam

Ritam je oblik koji čini ravnomjerni niz jednakih elemenata. Ritmotvorni su elementi u govoru mőre (slogovnost), stope (govrne riječi) ritmičke skupine odvojene stankama, te ritmičke fraze (rečenice). Na razini slogova, ritmička se raznolikost ostvaruje nizanjem jezično dugih ili kratkih slogova.

2.8.1.8 Govorne modulacije

Govorna se modulacija očituje u promjenama govornog zvuka u vremenskom slijedu. Zvučni dojam promjena zvuka naziva se dinamična boja zvuka. Prozodijska se modulacija uglavnom svodi na dvije osobine:

- Zvučni prijelazi (tranzijenti)
- Periodične vokalne modulacije

2.8.1.9 Način izgovora glasnika

Izgovaranje je zasebno prozodijsko sredstvo koje sudjeluje u izražajnosti govora. Izgovaranjem glasnika se ostvaruju fonemi tako da se mogu slušno prepoznati. Glasnici mogu biti okrugli, spljošteni, izduženi, krupni, tanki. Mogu biti takvi da se osjete njihovi pokreti kao brzi, usporeni, balistični, vođeni, oštri, blagi, treperavi. Sve ovo pokazuje veliku, gotovo beskonačnu raznolikost izgovora glasnika.

2.8.1.10 Mimika i gesta

Mimiku čine pokreti lica, a gestu pokreti ruku, glave, ramena i drugih dijelova tijela. S obzirom da je ovo prozodijsko sredstvo uglavnom vizuelno, manje je relevantno za ovo istraživanje.

2.8.1.11 Znakovi u glasu

Postoji nekoliko značajki glasovnih znakova:

- Govornikove trajne osobine
- Raspoloženja i stanja za vrijeme govora
- Komunikacijski uvjeti

- Odnos prema tekstu i jeziku teksta

2.8.2 Naglašavanje riječi

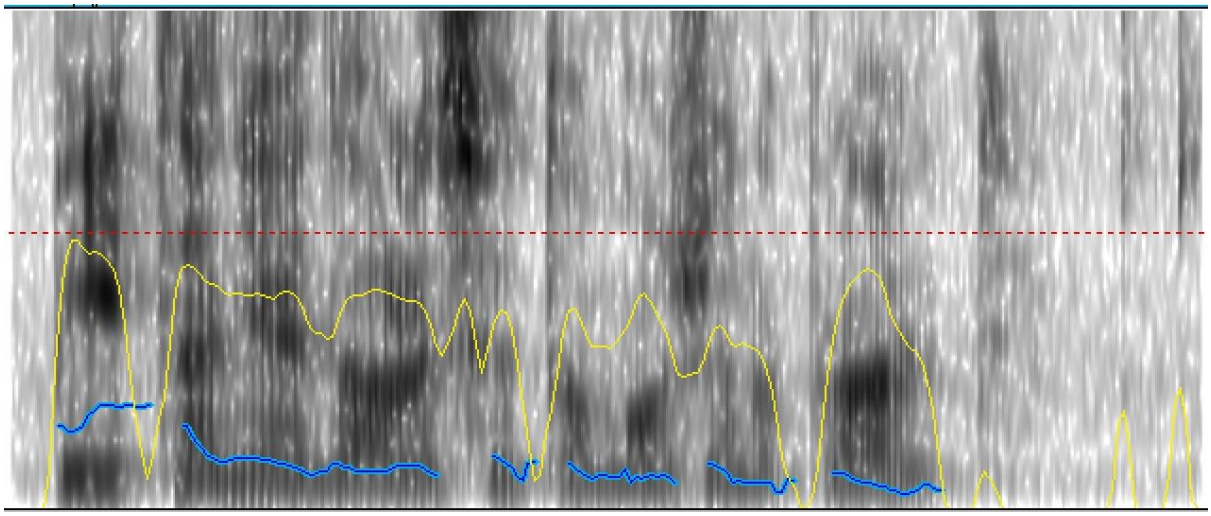
Naglašavanje riječi u govoru podrazumijeva određene akustičke karakteristike koje odstupaju od onih vezanih za nenaglašene riječi. Prema (Barić, i dr., 1995) "naglasak je istodobni ostvaraj siline, tona i trajanja". S akustičkog stajališta, postoje tri osnovna kriterija po kojima možemo utvrditi je li neka riječ naglašena:

- Trajanje
- Intenzitet
- Osnovna frekvencija (F_0)

Trajanje se odnosi na vremensku duljinu trajanja nekog segmenta zvuka. Na primjer, kada netko izgovori rečenicu „*Ma nisam!"* tako da produlji izgovor vokala *i* onda je na taj način istaknuo riječ *nisam*. Istovremeno, taj vokal se za tu svrhu može izgovoriti s pojačanim intenzitetom tako da se i na taj način ova riječ može istaknuti. Nadalje, riječ se može istaknuti tako da se povisi ton na tom mjestu, što bi se očitovalo kao povišenje osnovne frekvencije.

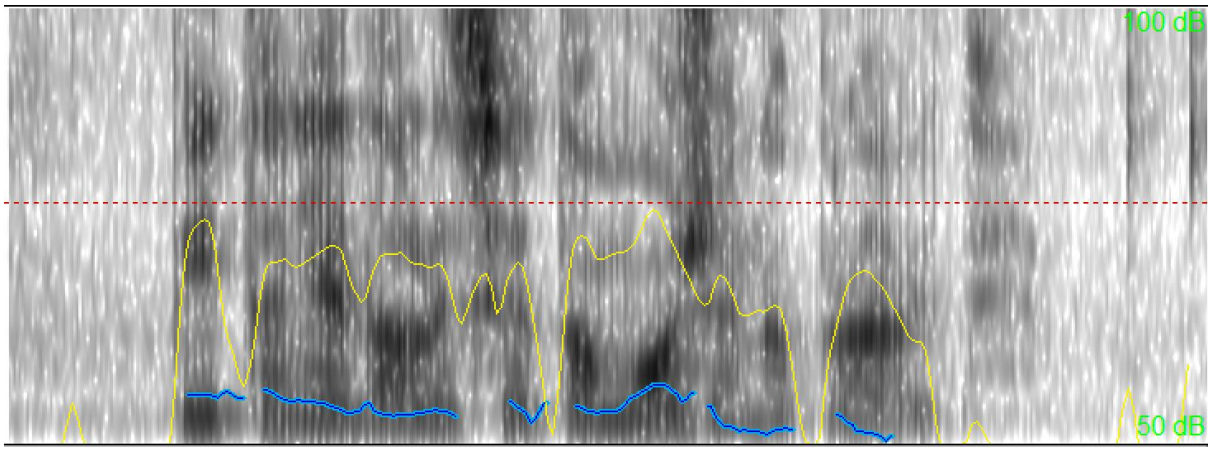
Na slici 40 prikazan je spektrogram rečenice „*Pitanje je da li se to može napraviti.",* tako da je riječ „pitanje" naglašena. Na slici 41 izgovorena je ista rečenica, ali tako da je riječ „može" naglašena. Na tim slikama plava linija označava kretanje tona, a žuta intonacije. Vidi se da je na slici 40, na početku gdje je izgovorena riječ „pitanje", plava linija vidljivo povišena u odnosu na ostale njene dijelove. To je zbog toga što je na tom mjestu ton povišen jer je riječ izgovorena naglašeno. Slično se vidi i na slici 41 na mjestu na kojem je izgovorena riječ „može" koja je na tom spektrogramu naglašena. Isto tako se vide i promjene intenziteta (žuta linija), ali one su manje istaknute.

Prema (Kroul, 2009) većina istraživanja u ovom području fokusirana je na detekciju promjene tona, što je s tehničke strane promjena osnovne frekvencije.



Pitanje ...

Slika 40: Spektrogram rečenice „Pitanje je da li se to može napraviti“ u kojem je naglašena riječ „pitanje“.



... može ...

Slika 41: Spektrogram rečenice „Pitanje je da li se to može napraviti“ u kojem je naglašena riječ „može“.

3. Osnovne tehnike analize i obrade govornog signala

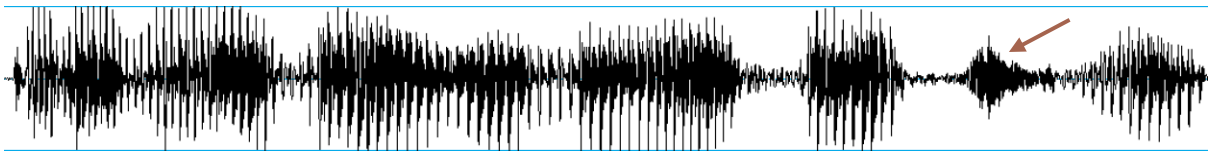
Da bi iz nekog zvučnog zapisa govora mogli dobiti korisne podatke taj je zapis potrebno na adekvatan način pripremiti, prikazati i analizirati. U ovom dijelu opisane su tehnike analize zvuka koje su relevantne za ovo istraživanje.

3.1.1 Oscilogram

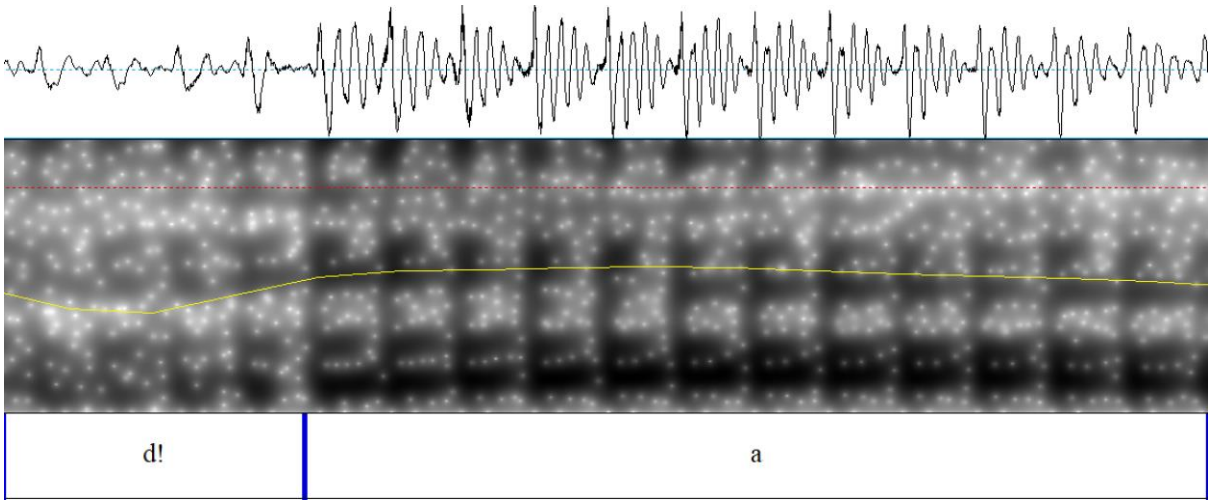
Oscilogram prikazuje zvuk u vremenskoj domeni, odnosno pokazuje promjene zvučnog tlaka u vremenu. Na ovakvom prikazu najbolje se vidi intenzitet (na vertikali), dok je frekvenciju teže odrediti - ona se očituje kao gustoća zubaca. Iako karakteristike pojedinačnih glasova općenito nisu uočljive na ovakvom prikazu, on može poslužiti kao dodatno sredstvo za procjenu početka i završetka glasova s višim frekvencijama. Na primjer, na slici 42 strelicom je označen dio gdje je izgovoren glas “č”. U usporedbi s okolnim zvukom primjećuje se veća gustoća zubaca, ali i manji intenzitet. To je zbog toga što je frekvencija glasa “č” viša od one okolnih glasova, pa se stoga i vidno razlikuje na oscilogramu, a isto tako ovaj je glas bezvučan (on je u stvaru šum) pa je zbog toga slabijeg intenziteta. Za ovo istraživanje oscilogram je bio od pomoći kod segmentacije govora jer na spektrogramu kod ovakvih glasova nije uvijek bilo lako odrediti gdje oni počinju i završavaju.

Oscilogram je također koristan kada treba prikazati periodičnost. Periodičnost je prisutna kod vokala, kao što se vidi na slici 43, na kojoj se glas „a“ nalazi odmah iz glasa „d“. Na toj se slici na oscilogramu vidi periodičnost zvuka glasa „a“ koji se sastoji od kratkih (sličnih) segmenata koji se ponavljaju.

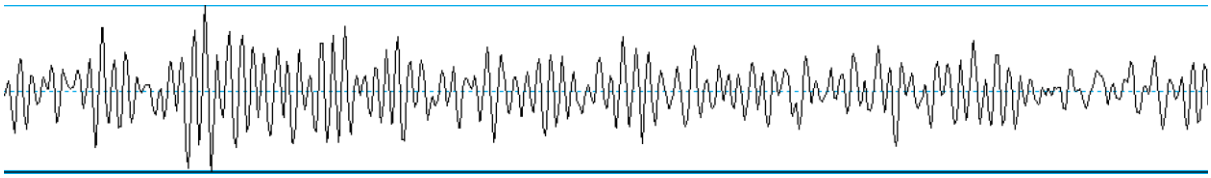
Titranje glasnica proizvodi periodične zvučne titraje, a nepravilni vrtlozi zračne struje aperiodične. Na slici 44 prikazan je (povećan) oscilogram glasa „č“ u riječi „oblačno“, gdje se primijećuje odsutnost periodičnosti titraja jer je taj glas samo vrtlog zračne struje.



Slika 42: Oscilogram govora u kojem je izrečeno „prevladavalo oblačno“. Strelica označava dio povišene frekvencije gdje je izgovoren glas „č“.



Slika 43: Oscilogram i spektar dijela riječi „prevladavalo“ s izdvojenim dijelom „da“. Uočljiva je periodičnost zvuka za glas „a“.



Slika 44: Oscilogram glasa „č“ u riječi „oblačno“.

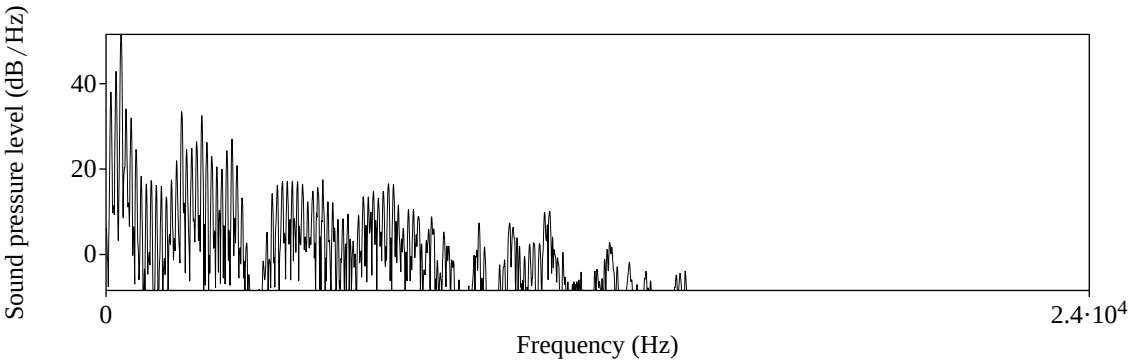
3.1.2 Spektralna analiza

Jedna od najvažnijih tehnika analize zvuka je spektralna analiza. Spektar nekog zvuka nam pokazuje kolika je zastupljenost pojedinih frekvencija u tom zvuku i s kojim intenzitetom. Rezultat ove analize može se prikazati pomoću spektrograma, kao onaj na slici 40 ili 41, na kojem se intenzitet frekvencija vidi kao zacrnjenje (što je ono veće to je intenzitet jači), ili se može prikazati kao spektar gdje je zastupljenost pojedine frekvencije prikazana stupcem čija visina odgovara intenzitetu, a pozicija na x-koordinati frekvenciji. Spektralna analiza se radi tako da se kompleksan zvuk rastavi na njegove sastavne komponente (sinusoide) matematičkom tehnikom koja se zove *Fourierova analiza* i koja daje informaciju o tome koliko je svaka sastavna frekvencija nekog kompleksnog periodičnog zvuka zastupljena. Spektralna analiza je važna za analizu govora jer se u spektrogramu vide formanti, a pomoću njih se mogu utvrditi svojstva glasa koji je njime prikazan. Na osnovu tih svojstava moguće je utvrditi o kojem se glasu radi. Ovo je posebno važno za strojno učenje upotrebom neuralne mreže jer upravo su specifične karakteristike glasova ono po čemu ih ona može razlikovati (učenjem) od ostalih glasova za potrebe klasifikacije.

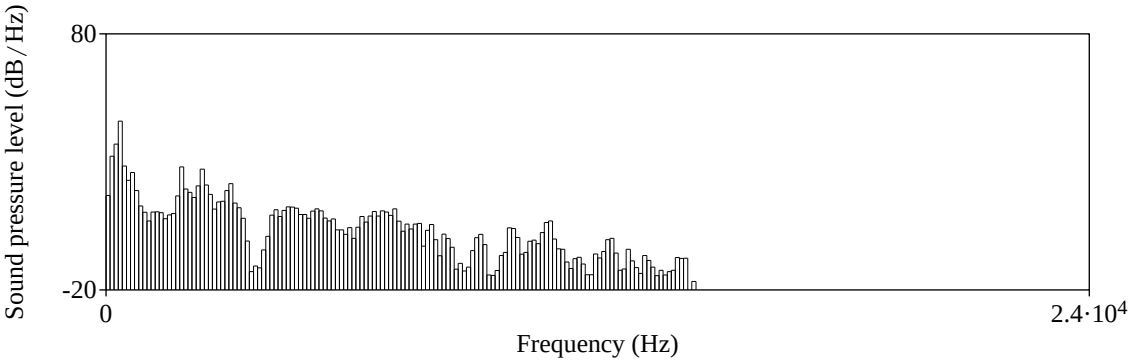
3.1.3 LTAS (Long Term Average Spectrum) ili usrednjeni spektar

Ovakav spektar je jedna varijanta spektralne analize koja nam omogućava da radimo s manjim brojem podataka. Na slici 45 prikazan je spektar glasa *i*. Frekvencijski raspon koji je tu obuhvaćen je 22.05 kHz. Taj je prikaz podijeljen na 16385 frekvencijskih intervala tako da svaki od njih prikazuje korak od 1.345 Hz.

Na slici 46 prikazan je usrednjeni spektar sa slike 45. Ovdje je cijeli frekvencijski raspon podijeljen na intervale od 100 Hz, tako da svaki stupić prikazuje prosječni spektar na frekvencijskom intervalu od 100 Hz. Može se vidjeti da je oblik spektra isti kao onaj na slici 45, samo što je usrednjeni spektar manje „detaljan“. Na ovaj se način dobije manja količina podataka s kojom je praktičnije raditi u svrhe treniranja neuralne mreže, što će kasnije biti opisano.



Slika 45: Spektar glasa „i“.



Slika 46: Usrednjeni spektar glasa „i“.

3.1.4 Računalni alati

Za ovo istraživanje korišteni su sljedeći softverski alati:

- Praat – program za analizu govora.
- Python – programski jezik (verzija 3).
- Biblioteka za strojno učenje *scikit-learn* za Python.

Sve su ovo besplatni „open-source“ alati koji se mogu instalirati na većini popularnih računalnih operacijskih sustava (kao što su Linux, Windows i MacOS). Iako Praat ima svoj programski jezik za ovo smo istraživanje upotrijebili Python zbog nekoliko razloga:

- Python je trenutno jedan od najpopularnijih programskih jezika, pa je stoga jako dobro dokumentiran, što uključuje opsežnu literaturu i veliku količinu informacija na internetu.
- Python je fleksibilniji programski jezik od onoga u Praatu jer je napravljen za potrebe programera u raznim područjima primjene.
- Za Python postoji velik broj biblioteka, ne računajući samo njegovu standardnu biblioteku, već i one koje su napravile druge organizacije i programeri za specifične potrebe.
- Biblioteka za Python *scikit-learn* bila je upotrebljena za ovo istraživanje zbog dobre podrške za neuralne mreže i drugih modela i funkcija potrebnih za strojno učenje.
- Python je open-source programski jezik dostupan na većini računalnih platformi kao što su Windows, Linux i MacOS.

4. Strojno učenje i neuralne mreže

4.1 Pregled područja automatskog prepoznavanja govora

Automatsko prepoznavanje govora je područje računarstva kojem je cilj izrada računalnog programa koji zvuk govora može konvertirati u tekst ili u neki drugi oblik iz kojeg se jasno mogu razlučiti pojedine izgovorene riječi. Tradicionalno, postoji nekoliko osnovnih pristupa automatskom prepoznavanju govora (Dhanashri & Dhonde, 2015):

- Predlošci: Ovdje se govorni signal uspoređuje s većim brojem unaprijed definiranih predložaka snimljenog govora i traži se onaj koji najbolje odgovara tom ulaznom govornom signalu. Ovdje se u stvari radi o prepoznavanju uzoraka. Osnovni problem kod ovog pristupa je taj da je za varijacije u govoru potrebno imati velik broj predložaka, što je često nepraktično.
- Baza-znanja: Za ovaj se pristup upotrebljava velik skup informacija o lingvističko-fonetskim aspektima govora, uključujući i spektrograme, ali i ovdje problem stvaraju varijacije u govoru.
- Statistički modeli: Ovi su modeli zasnovani na ideji automatskog učenja. Za ovaj pristup u upotrebi su dvije vrste računalnih modela: Skriveni markovljevi modeli ili HMM (engl. *Hidden Markov Models*) i neuralne mreže.

U ovom dijelu prikazan je kratki pregled ovog područja prema (Juang & Rabiner).

Područje automatskog prepoznavanja govora je krenulo s ozbiljnijim razvojem nakon što je ustanovljena važnost spektra govornog signala za identifikaciju fonetskih elemenata u govoru (Juang & Rabiner), (Fletcher, 1922). Većina modernih sustava za prepoznavanje govora zasnovana je na spektralnoj analizi govornog signala, a takva se analiza danas može izvoditi efikasno upotrebom modernih tehnika obrade digitalnog signala. Kod ranijih sustava kod kojih se upotrebljavala spektralna analiza postupak prepoznavanja uglavnom se temeljio na principima akustičke fonetike, gdje su glavne elemente tvorili fonemi. Prepoznavanje riječi zavisilo je o analizi formanata, gdje je cilj bio na osnovu njih prepoznati pojedine glasove, pa time i riječi. Neki takvi sustavi opisani su u (Davis, Biddulph, & Balashek, 1952), (Olson & Belar, 1956), (Forgie & Forgie, 1959). Drugi sustavi su bili fokusirani na prepoznavanje vokala (Suzuki & Nakata, 1961), prepoznavanje fonema (Sakai & Doshita, 1962), te prepoznavanje brojeva (Nagata, Kato, & Chiba, 1963). Ovaj zadnji rad bio je prvi u kojem je upotrebljeno segmentiranje govora. U (Fry & Denes, 1959) opisan je sustav koji prepoznaje četiri vokala i

devet konsonanata u kojem su upotrebljene statističke informacije o dozvoljenom nizu fonema, čime su poboljšali performanse sustava za prepoznavanje riječi koje se sastoje od dva ili više fonema. Tim je radom započela upotreba statističke sintakse za automatsko prepoznavanje govora.

U kasnim 60im godinama razvijena je tehnika koja se zove *Linear Predictive Coding (LPC)* (Atal & Hanauer, 1971), (Itakura & Saito, 1970), koja znatno pojednostavljuje analizu govornog signala koji proizvodi govorni trakt. Kasnije su osnovne ideje primjene tehnologije za prepoznavanje uzoraka u prepoznavanju govora, zasnovanima na LPC, upotrebljene u (Itakura, 1975) i (Rabiner, Levinson, Rosenberg, & Wilpon, 1979). Neki od sustava razvijenih u to vrijeme je „Harpy" (Lowerre, 1990) sa sveučilišta Carnegie Mellon University koji je mogao prepoznati govor upotrebom riječnika od 1.011 riječi. Jedan važan doprinos ovog sustava bio je princip pretraživanja grafa, gdje je jezik za prepoznavanje prikazan kao mreža izvedena iz leksičkog sadržaja riječi, zajedno sa sintaksnim pravilima i pravilima o granicama riječi. Ideja je bila da sustav radi tako da se govor prvo segmentira, a zatim da se ti segmenti uspoređuju s predlošcima koristeći takozvanu Itakura-distancu (Itakura, 1975). Neki drugi sustavi razvijeni u to vrijeme su Hearsay-II i HWIM (Klatt, 1977).

Tih su godina kompanije IBM i AT&T radile na razvoju komercijalnih sustava za prepoznavanje govora. IBM je radio na razvoju glasom-kontrolirane pisaće mašine čiji je cilj bio da govor prevodi u slova i riječi koje bi bile prikazane na zaslonu (Jelinek, Bahl, & Mercer, 1975). Nedostatak tog sustava bio je u tome da je mogao raditi samo sa govornicima za koje je bio prilagođen. AT&T je imao za cilj razviti sustav za automatske telekomunikacijske usluge. Ti su sustavi trebali raditi s velikim brojem govornika, s raznim naglascima i dijalektima, bez da se sustav mora konfigurirati ili trenirati za specifične govornike. S obzirom da su ovakve primjene obično podrazumijevale kratke govorne segmente istraživanje se usredotočilo na razvoj akustičkog modela (spektralni oblik riječi ili fonema) u kombinaciji s jezičnim modelom (prikaz sintakse tipičnih rečenica za ovakve potrebe), a također se radilo na konceptu uočavanja ključnih riječi (Wilpon, Rabiner, Lee, & Goldman, 1990). Na primjer, ako sustav korisniku nudi izbor da odabere plaćanje na rate ili u cijelosti onda ako korisnik spomene riječ „rate" sustav može pretpostaviti da je korisnik odabrao plaćanje na rate, bez obzira što možda nije prepoznao ostale riječi u rečenici, koja je mogla biti „Želim opciju na rate".

Kasnije, 80ih godina prošlog stoljeća istraživanje u ovom području bilo je više usmjereno ka statističkim metodama, gdje je matematički model *Hidden Markov Model (HMM)* poslužio kao

osnova takvim sustavima (Jelinek, Continuous Speech Recognition by Statistical Methods, 1976) (Levinson, Rabiner, & Sondhi, 1983). Nakon publikacije teorije HMM (Ferguson, 1980) ovaj je statistički model postao prvim izborom za sustave za automatsko prepoznavanje govora (Rabiner & Juang, Statistical Methods for the Recognition and Understanding of Speech, 2004). Princip rada sustava temeljenih na HMM je taj da se pripremi dovoljno velik broj testnih podataka kojima se kasnije odgovarajućim metodama za procjenu utvrde optimalni parametri koji odgovaraju zadanom modelu (Baum, 1972). Neke od tehnika razvijenih za rad s HMM opisane su u (Poritz, 1982) i (Liporace, 1982). Neki sustavi razvijeni na ovakvoj tehnologiji su Sphinx (Lee, 1988), BYBLOS (Schwartz, i dr., 1989) i DECIPHER (Murveit, i dr., 1989).

Još jedna tehnologija koja se vratila na scenu u ovom području su *umjetne neuralne mreže*. One su nastale 50ih godina prošlog stoljeća, ali u početku nisu davale dobre rezultate (McCullough & Pitts, 1943). Sa stajališta prepoznavanja govora, neuralne su mreže u početku bile korištene za prepoznavanje nekolicine riječi ili fonema, gdje su davale dobre rezultate (Lippmann, 1990).

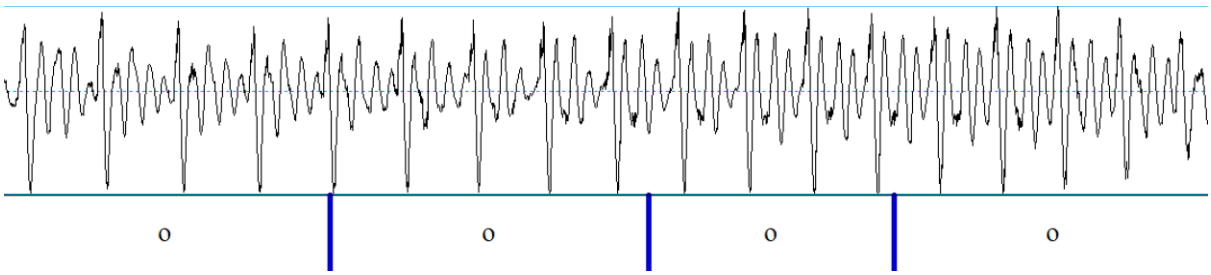
4.2 Osnovni principi i modeli za strojno učenje

Za ovo istraživanje bilo je potrebno implementirati sustav koji će moći (otprilike) odrediti vremensku poziciju riječi u zvučnom zapisu. Da bi takav sustav radio on mora biti u mogućnosti prepoznati barem jedan dio glasova koji su na snimci izgovoreni tako da se usporedbom tih glasova s tekstom podnatpisa može odrediti koje su riječi izgovorene na snimci. Takva vrsta problema – prepoznavanje glasova ili, općenito, prepoznavanje govora, pripada području *strojnog učenja*. Strojno učenje se primjenjuje za riješavanje problema kod kojih ne postoji egzaktni algoritam. Na primjer, za problem sortiranja vrijednosti u rastućem ili opadajućem poretku postoje mnogi egzaktni algoritmi koji rade s bilo kojim nizom vrijednosti među kojima postoji odnos manje/veće. Međutim, postoji velik broj problema za koje egzaktni algoritmi nisu poznati. Na primjer, prepoznavanje objekata na fotografijama ili u prostoru, prepoznavanje lica, rukopisa, govora ili glasa neki su od tih problema. Na slici 47 (stranica 58) prikazan je oscilogram glasa *o* podijeljen na četiri dijela. Ako se pažljivo pogleda grafički prikaz impulsa tog glasa primjećuje se da niti jedan od četiri dijela nije isti kao neki drugi. Oni jesu slični, ali nisu isti. Ovdje bi bilo teško definirati pravila za to kako izgleda glas *o*. Ono što bi taj zadatak otežalo je činjenica da glas *o*, kao i svi ostali glasovi, nema samo oblik koji je prikazan na slici 47, nego ih može imati gotovo beskonačno mnogo. To može ovisiti o okolnim glasovima, okolnoj buci, načinu izgovora, a može biti i djelomično izgovoren, pogrešno izgovoren, izgovoren glasnije ili tiše ili čak preskočen (taj slučaj stvara dodatne probleme kod identifikacije riječi).

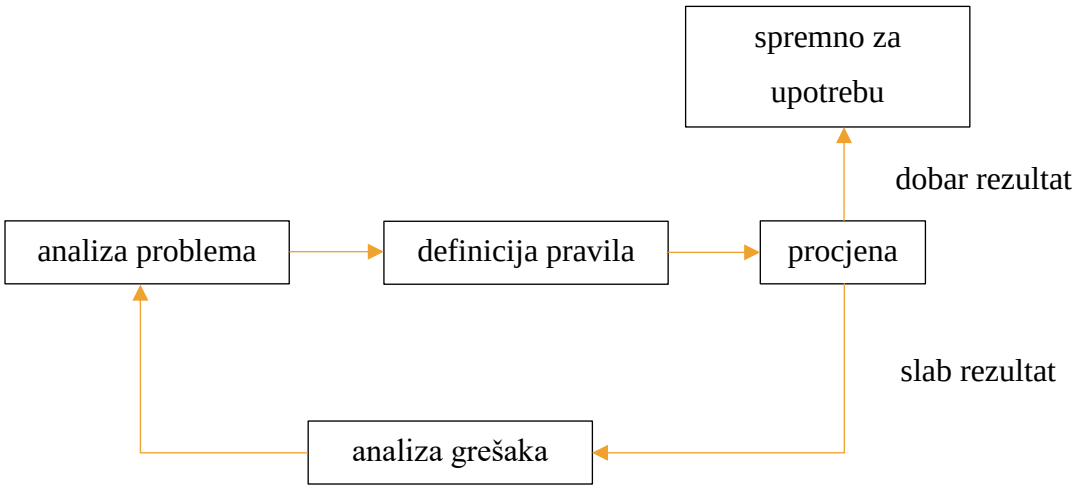
Zbog svega ovog u ovom istraživanju upotrebljeni su neki principi i tehnike strojnog učenja. Pristup strojnom učenju drugačiji je od tradicionalnog pristupa razvoju algoritama i programa. Na slikama 48 i 49 prikazana je razlika između tradicionalnog pristupa i onog strojnog učenja prema (Geron, 2017). U toj se knjizi navodi primjer izrade spam filtera, gdje bi u tradicionalnom pristupu slijedili sljedeće korake:

1. Prvo bi ustanovili kako tipični spam izgleda, kao što su česte riječi ili fraze, s nekim drugim pokazateljima (kao što je email adresa pošiljatelja i slično).
2. Za detekciju svih uočenih karakteristika u koraku 1 definirali bi algoritam i po njemu napisali program.
3. Testirali bi program i po potrebi ponavljali korake 1 i 2 sve dotle dok program ne bi davao zadovoljavajuće rezultate.

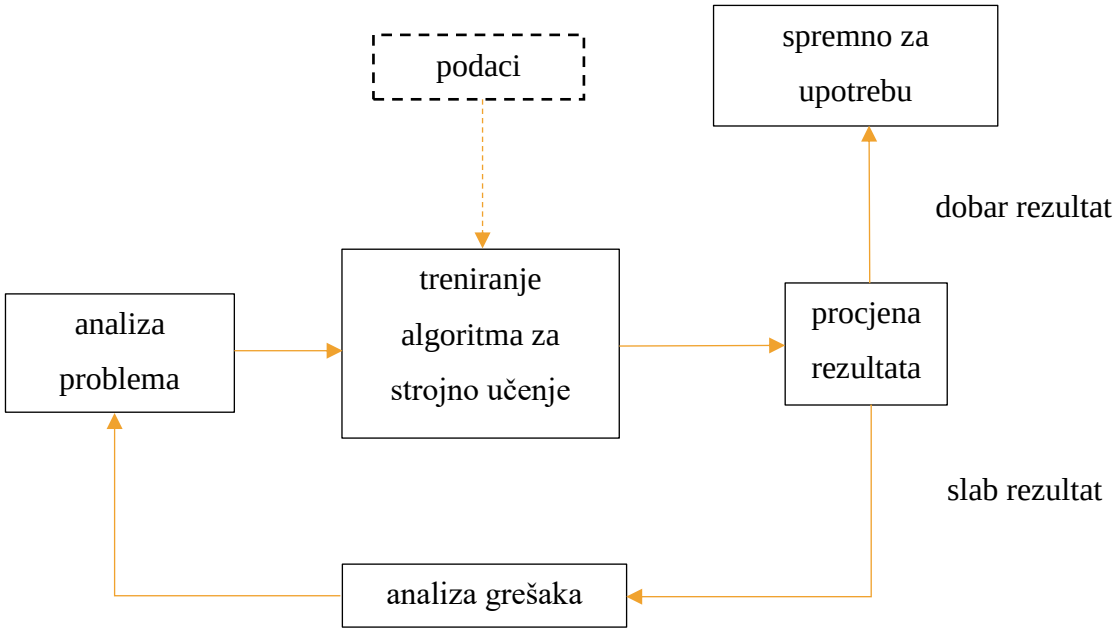
Ovako napisan program bi zahtijevao niz kompleksnih pravila, što bi bilo teško za održavanje. Ako se, međutim, odabere pristup strojnog učenja onda bi program automatski uočavao riječi i fraze koje se često ponavljaju u takvim porukama, bez da se algoritam za to eksplicitno mora definirat. Takav je program i jednostavniji za proširivanje jer je nove primjere potrebno samo dodati u skup primjera za treniranje. Na primjer, ako se u spam porukama počnu pojavljivati neke nove riječi i fraze, kao što je *niske kamate,* onda program za strojno učenje može detektirati nagli porast poruka koje su korisnici označili kao spam i koje sadrže ovu frazu. Ova je činjenica također relevantna za ovo istraživanje jer je važno da se sustav može lako proširiti novim primjerima glasova za treniranje neuralne mreže. Ova mogućnost prilagođavanja sustava strojnog učenja promjenama prikazana je na slici 50 (stranica 59).



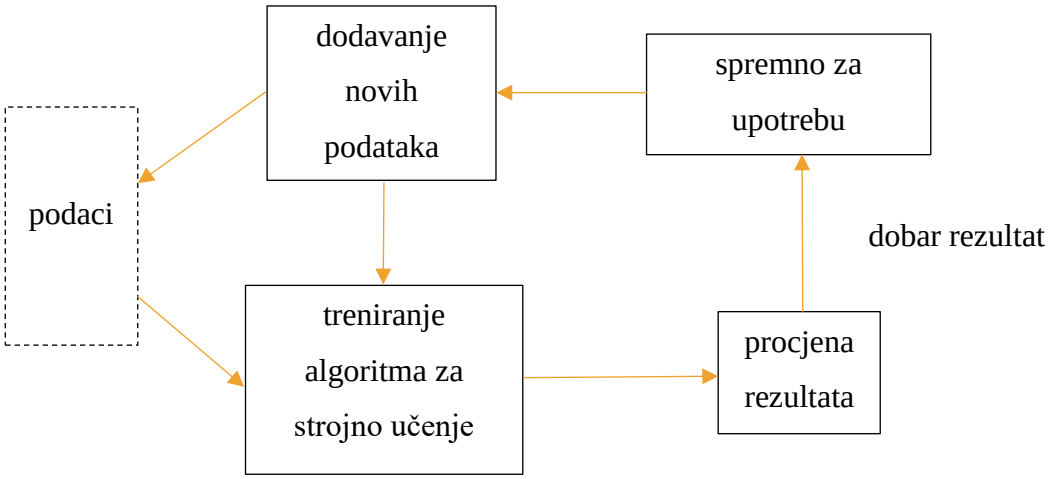
Slika 47: Impulsi glasa 'o'.



Slika 48: Tradicionalni pristup razvoju programa.



Slika 49: Razvoj programa za strojno učenje.



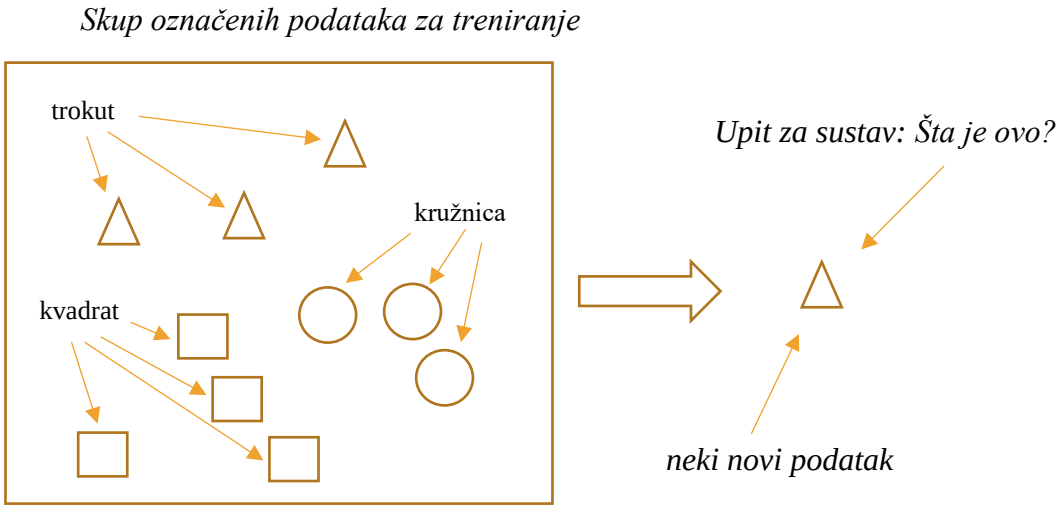
Slika 50: Automatsko prilagođavanje sustava za strojno učenje.

4.3 Vrste sustava za strojno učenje

Postoji velik broj vrsta sustava za strojno učenje. Oni se mogu općenito klasificirati u tri skupine:

- Da li je postupak učenja nadgledan od strane čovjeka?
- Da li mogu učiti u tijeku rada?
- Da li rade tako da uspoređuju nove podatke s postojećim ili imaju mogućnost ustanovljavanja uzoraka u podacima za treniranje i stvaranja prediktivnog modela?

Za ovo istraživanje korištena je metoda takozvanog *nadgledanog učenja*, pa će ona ovdje biti ukratko opisana. Kod nadgledanog učenja ulazni podaci sadrže i informaciju o kategoriji kojoj pripadaju. Takva se vrsta učenja često koristi za klasifikaciju. Spam filter je jedan primjer takvog sustava, gdje korisnik odredi je li poruka spam i onda ta poruka može poslužiti kao primjer spama. Još jedan primjer je klasifikacija fonema – ono što je upotrebljeno u ovom istraživanju. Korisnik iz neke snimke govora izdvoji pojedinačne foneme i označi ih pripadajućim slovom ili nekim odgovarajućim simbolom, a sustav za učenje to iskoristi kao primjer fonema. Ovaj način učenja prikazan je na slici 51.



Slika 51: Nadgledano strojno učenje.

4.4 Neuralne mreže

Jedan od ključnih računalnih modela upotrebljenih za ovo istraživanje je *neuralna mreža*. Ona se ovdje upotrebljava za prepoznavanje pojedinačnih glasova u zvučnom zapisu. Kada se radi o prepoznavanju govora najveća poteškoća je u tome da njegov zvučni signal može imati gotovo beskonačan broj oblika. U svim tim oblicima, međutim, postoje određene zakonitosti koje ljudski mozak lako prepozna. Problem je u tome što za utvrđivanje tih zakonitosti ne postoje egzaktni algoritmi. Jedan razlog je taj što naš mozak može nadoknaditi ono što nije dobro čuo time što zna jezik – gramatiku i leksikon. Ako govornik kaže *Nisam stigao na autobus*, a slušatelj nije čuo prvi glas *n* onda će slušatelj najvjerojatnije dobro razumijeti šta je govornik rekao jer „rupe“ u prepoznavanju može popuniti time što je razumio ostale riječi u toj rečenici. Drugi, još veći problem za automatsko prepoznavanje govora je kontekst. Ako se slušatelj gornje rečenice nalazi na autobusnoj stanici onda će mu biti još lakše nadoknaditi ono što nije dobro čuo jer će biti u stanju zaključiti neke glasove ili cijele riječi.

Sve je to vrlo teško ili čak nemoguće jasno definirati u obliku nekog računalnog algoritma jednostavno zato što je broj zvučnih varijanti nekog izgovorenog glasa prevelik i, što je još važnije, nepoznat. Zbog svega toga se za (dijelomično) riješenje ovakvih problema pribjegava heurističkim metodama, to jest metodama čiji rezultat ne mora biti optimalan, ali je dovoljno dobar.

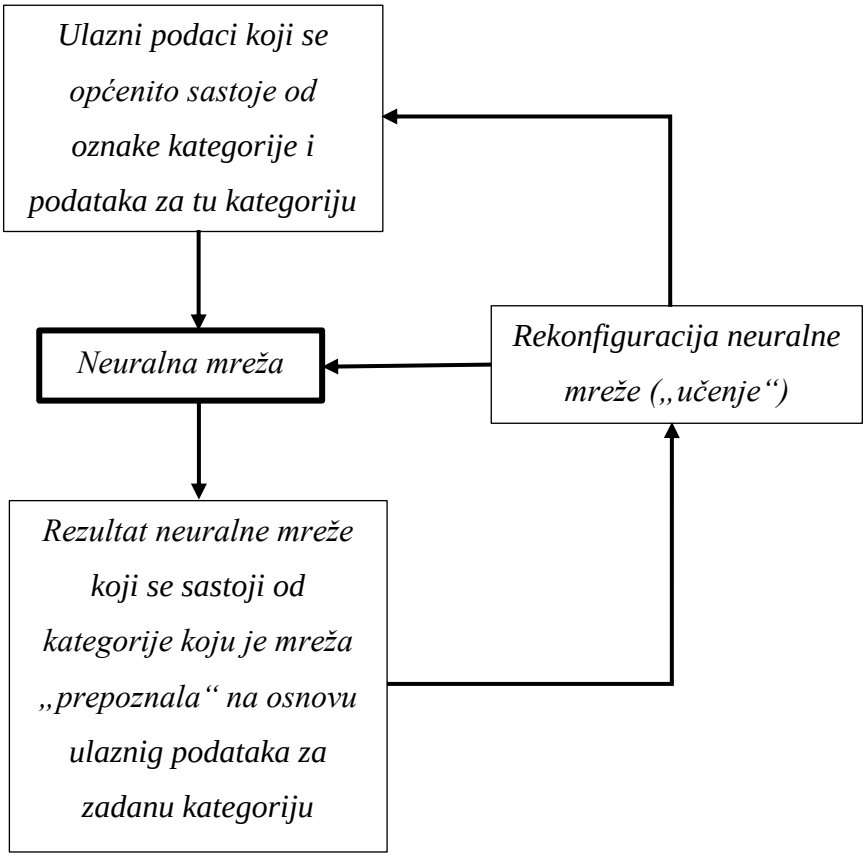
Neuralne mreže su jedan model koji je zasnovan na ideji *učenja* gdje se osigura dovoljno velik broj podataka koji predstavljaju *primjere* zajedno s poznatim rezultatom (takozvano *nadgledano učenje*), a postupkom učenja neuralne mreže ona se „konfigurira“ na način da za svaki taj primjer daje rezultat koji se za njega očekuje. Slika 52 pokazuje općeniti princip treniranja neuralne mreže. Cilj tog postupka učenja je da neuralna mreža eventualno bude u stanju korektno klasificirati i podatke koji nisu bili korišteni za njeno učenje. Primjerice, veći broj snimljenih glasova *a* može poslužiti kao skup primjera kojim se neuralna mreža uči prepoznati taj glas, s ciljem da ona eventualno bude u stanju prepoznati glas *a* koji nije bio u tom skupu primjera, recimo izgovor nekog drugog govornika ili čak istog govornika u drugačijim uvjetima. Prema (Geron, 2017) neuralne mreže zadnjih godina imaju dobru perspektivu iz nekoliko razloga:

- Postoje velike količine podataka za treniranje neuralnih mreža i one često daju bolje rezultate za vrlo velike i kompleksne probleme.

- Veliki porast računalne snage od 1990s omogućuje treniranje velikih neuralnih mreža u prihvatljivom vremenskom roku. Jednim je dijelom tome zaslužna i industrija računalnih igara koja je proizvela velik broj GPU (Graphics Processing Unit) kartica jer one imaju specijalne procesore za brzo računanje pa su pogodne i za grafičke simulacije.
- Algoritmi za treniranje su unaprijeđeni, iako se ne razlikuju puno od onih u 1990im godinama. razlike su dovoljne da je uočen veliki napredak.
- Neka teoretska ograničenja se nisu pokazala kao veliki problem u praksi. Na primjer, za neke algoritme za treniranje smatralo se da neće biti praktični zbog toga što nakon nekog vremena više ne poboljšavaju rezultate, ali se pokazalo da je to u praksi rijetkost ili da su i takvi rezultati blizu optimalnih.
- Istraživanje neuralnih mreža zadnjih godina dobro je financirano, pa zbog toga i dobro napreduje. Isto tako, puno je izuzetnih proizvoda napravljeno s njima u pozadini (kao što su samovozeći automobili), što još više potiče istraživanje i razvoj ovog područja.

Za treniranje neuralnih mreža od ključnog je značaja da postoji dovoljno velik skup podataka za treniranje jer u protivnom ona neće davati dobre rezultate. Drugi važan faktor pri upotrebi neuralnih mreža je način na koji su ti podaci pripremljeni za treniranje; ako se iz tih podataka ne očituju dobro svi relevantni elementi onda također rezultati neće biti dobri. Prema tome, rad s neuralnim mrežama zahtijeva iskustvo i eksperimentiranje. Specifično, potrebno je odgovoriti na pitanja od kojih su neka od najvažnijih sljedeća (Winston, 1993):

- Kako prikazati ulazne podatke tako da neuralna mreža može raditi s njima?
- Kako interpretirati izlazne podatke tako da dobijemo informaciju o onome što želimo znati?
- Koliko nam neurona treba za problem koji želimo riješiti?
- Kakva nam struktura neuralne mreže treba za problem koji želimo riješiti?
- Na koji način treba trenirati neuralnu mrežu?



Slika 52: Postupak „treniranja“ neuralne mreže.

4.4.1 Opis modela i princip treniranja neuralnih mreža

S obzirom da je neuralna mreža jedan od glavnih alata upotrebljenih za ovo istraživanje u ovom djelu opisan je osnovni princip po kojem taj računalni model radi.

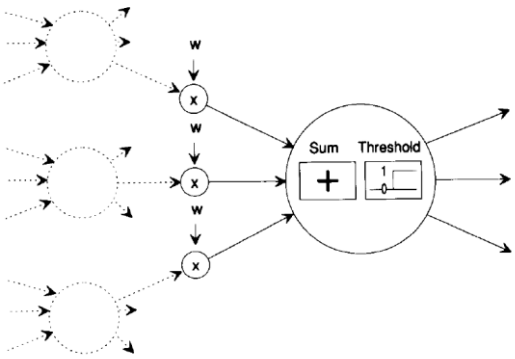
S obzirom da je neuralna mreža zamišljena tako da donekle simulira biološku neuralnu mrežu ona se sastoji od elemenata koji odgovaraju neuronima i vezama među njima. Neuralna mreža se sastoji od skupine *članova* ili *vrhova* koji su spojeni *poveznicama*. Ti članovi predstavljaju umjetne *neurone*. Svaka poveznica ima pridružen broj koji označava *težinu* te poveznice. Taj je broj glavni element za učenje neuralne mreže jer se njegovim ažuriranjem neuralna mreža konfigurira tako da bolje odgovara ulaznim podacima. Jedna grupa neurona sadrži ulazne podatke, pa time predstavlja ulaz neuralne mreže. Isto tako, postoji grupa neurona koja sadrži rezultat neuralne mreže. Cilj treniranja neuralne mreže je ažuriranje težine poveznica tako da se ulaz i izlaz dovedu u odnos koji je u skladu s očekivanim rezultatima.

Svaki neuron neuralne mreže sastoji se od skupa ulaznih poveznica koje dolaze iz drugih neurona, skupa izlaznih poveznica koje se vezuju za druge neurone, te trenutnu *aktivacijsku razinu*. Nadalje, svaki neuron ima formulu ili algoritam po kojem se izračunava sljedeća aktivacijska razina na osnovu ulaznih poveznica i njihovih težina. Na slici 53 prikazan je tipičan neuron umjetne neuralne mreže, gdje se vide tri ulazne poveznice, svaka sa svojom težinom (w), te tri izlazne poveznice.

4.4.2 Osnovni princip funkcioniranja umjetnog neurona

Na slici 53 vidi se da se neuron sastoji od dvije komponente: jedna zbraja skup vrijednosti (koje se dobiju iz ulaznih poveznica), a druga određuje aktivacijsku razinu. Taj se postupak može definirati ovako:

1. Vrijednost svake poveznice pomnoži s njenom težinom w i na kraju zbroji sve tako dobivene vrijednosti;
2. Taj zbroj usporedi s razinom aktivacije; ako je taj nivo iznad specificiranog praga za taj neuron onda je njegova izlazna vrijednost 1, inače je 0. Ta je izlazna vrijednost nova razina aktivacije neurona.



Slika 53: Prikaz funkcionalnosti umjetnog neurona (Winston, 1993).

Komponenta koja zbroj vrijednosti transformira u novu aktivacijsku razinu zove se *aktivacijska funkcija*. Jednostavna aktivacijska funkcija može zbroj vrijednosti usporediti s nekim unaprijed zadanim *pragom aktivacije*, te na osnovu njega utvrditi sljedeću razinu aktivacije. Općenito, rezultat aktivacijske funkcije je izlazna vrijednost neurona u datom trenutku. Ta vrijednost se „šalje“ na sve njegove izlazne poveznice. Pomoću te se vrijednosti računaju ulazne vrijednosti drugih neurona (koji su vezani s tom poveznicom), zajedno s vrijednošću težine koja je varijabilna i koja se mora modificirati u skladu s rezultatima dobijenim na izlaznim neuronima.

4.4.3 Neke vrste neuralnih mreža

Postoje dvije osnovne vrste neuralnih mreža koje se razlikuju po topologiji i načinu funkcioniranja. Neke od tih vrsta su *perceptron, feed-forward* i *povratne* (engl. *recurrent*) neuralne mreže. U ovom dijelu opisane su osnovne karakteristike ovih triju vrsta neuralnih mreža.

4.4.3.1 Perceptron

Perceptron je vrsta neuralnih mreža po strukturi i funkcionalnosti sličnih feed-forward mrežama, ali koje imaju samo jedan sloj. Iako po mogućnostima ispod onih kod feed-forward mreža s više slojeva, ove su mreže značajne zbog toga što su pomoću njih pronađena mnoga svojstva neuralnih mreža koja vrijede za kompleksnije strukture i bile su prva vrsta neuralnih mreža na kojima je započelo istraživanje iz ovog područja (Minsky & Papert, 1969). Na slici 54 (stranica 67) prikazana je jedna takva mreža. Tu se vidi da je svaki izazni element neovisan o drugim izlaznim elementima. To znači da se kompleksna perceptron-mreža može izgraditi od jednostavnijih takvih mreža. Na slici 55 (stranica 67) prikazan je takav jedan jednostavan

perceptron koji ima samo jedan izlazni element, odnosno samo jedan neuron. Perceptron je računalni model sa sljedećim svojstvima:

1. Postoji samo jedan neuron.
2. Ulazni podaci su binarni, to jest mogu biti samo 0 ili 1.
3. Kao što je prikazano na slici 56 (stranica 68) ulazni podaci se unose u takozvane *logičke kutije*. One daju rezultat 0 ili 1 na osnovu svih svojih ulaznih vrijednosti.
4. Izlazna vrijednost perceptrona je 0 ili 1 zavisno od toga je li zbroj izlaznih vrijednosti logičkih kutija kombiniran s težinom *w* prelazi neki zadani prag.

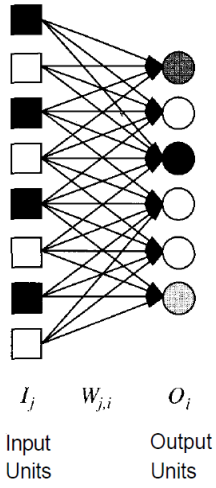
Ako označimo da je izlaz *i*-te logičke kutije l_i, *i*-ta težina w_i, a prag *T*, onda se rezultat perceptrona može opisati sljedećom formulom:

$$P = \begin{cases} 1 & \text{ako je } \sum_i (w_i * l_i) > T \\ 0 & \text{u suprotnom} \end{cases}$$

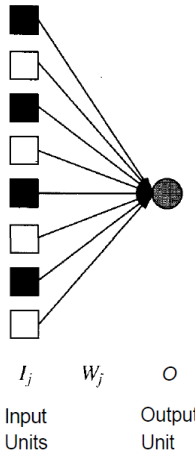
Uloga logičkih kutija je u tome da se ulazni podaci mogu raspodijeliti na segmente gdje svaka logička kutija određuje rezultat jednog takvog segmenta. Ti se rezultati kasnije zajedno s težinskim vrijednostima kombiniraju tijekom postupka učenja.

Jedna vrsta perceptrona je takozvani *dijametralno-ograničeni perceptron* (slika 57). Kod ove vrste perceptrona ulazni podaci su organizirani u matricu gdje se mogu upotrebljavati za učenje prepoznavanja dvodimenzionalnih oblika. Svaka logička kutija prima ulazne podatke iz jednog segmenta matrice koji je veličine nekog zadanog dijametra *d*, pa se u tom slučaju radi o dijametralno-ograničenom perceptronu dijametra *d*.

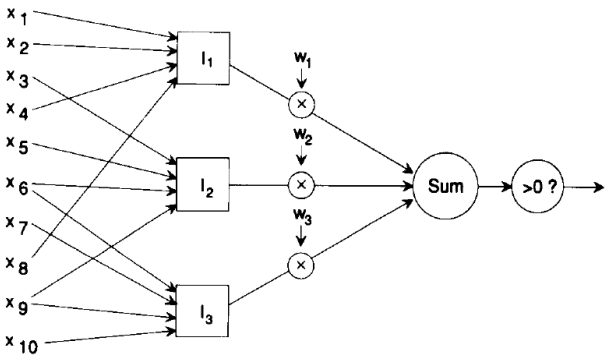
Još jedno pojednostavljenje perceptrona je takozvani *prolazni perceptron* (engl. *straight-through perceptron*). Ova vrsta perceptrona praktički nema logičke kutije, odnosno može se promatrati kao perceptron gdje svaka logička kutija ima jedan ulaz, a izlaz je isti kao i ulaz. Ovakav perceptron prikazan je na slici 58 (stranica 68).



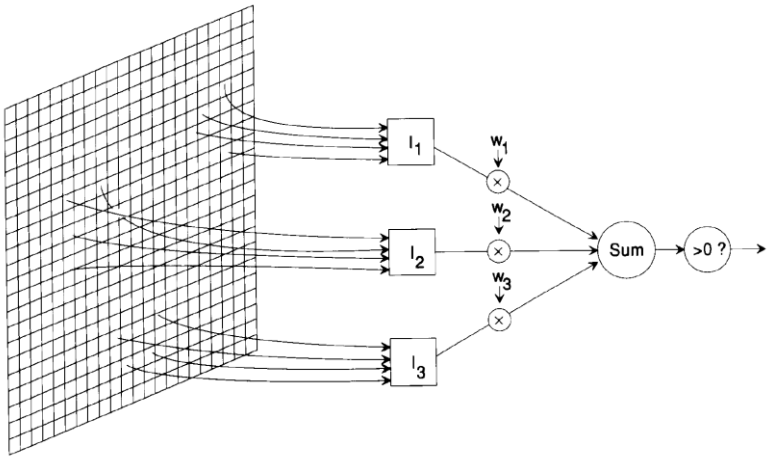
Slika 54: Mreža perceptrona (Russel & Norvig, 1995).



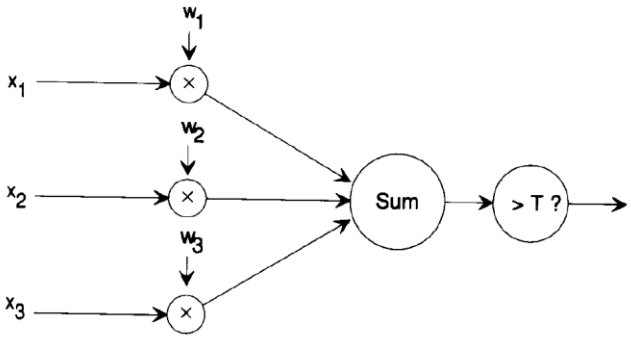
Slika 55: Jednostavni perceptron (Russel & Norvig, 1995).



Slika 56: Perceptron (Winston, 1993).



Slika 57: Dijametralno-ograničeni perceptron (Winston, 1993).



Slika 58: Prolazni perceptron (Winston, 1993).

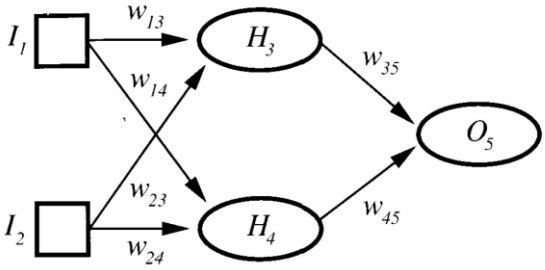
4.4.3.2 „Feed-forward“ neuralna mreža

Kako i sam naziv kaže, kod feed-forward mreža poveznice su jednosmjerne i ne postoje ciklusi, to jest nema poveznica koje se „vraćaju“ na isti element. S stajališta teorije grafova takve mreže imaju strukturu usmjerenog necikličkog grafa. Jedna takva mreža prikazana je na slici 59 i sastoji se od više slojeva, gdje se onaj u sredini naziva *skrivenim* slojem. Kod takve vrste mreže svaki element povezan je samo s elementima u sljedećem sloju. Jedna od prednosti ovakvih mreža je u tome da je računalni proces treniranja jednostavniji, s obzirom da se nove vrijednosti računaju jednosmjerno, od ulaznih elemenata prema izlaznim. Zbog toga ovakve mreže predstavljaju funkciju čiji su parametri vrijednosti težina poveznica. Mreža na slici 59 predstavlja funkciju (Russel & Norvig, 1995)

$$a_5 = g(w_{3,5}\, a_3 + w_{4,5}\, a_4) = g(w_{3,5}\, g(w_{1,3}\, a_1 + w_{2,3}\, a_2) + w_{4,5}\, g(w_{1,4}\, a_1 + w_{2,4}\, a_2))$$

gdje je g aktivacijska funkcija, a a_i je izlazna vrijednost elementa i.

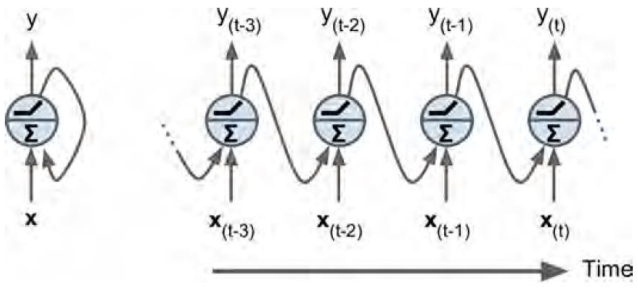
Iz ove ilustracije se vidi da je treniranje mreže ove vrste u stvari postupak kojim se “optimiziraju” vrijednosti parametara ovakve funkcije tako da njen rezultat što bolje odgovara onom očekivanom.



Slika 59: Model „feed-forward“ neuralne mreže s jednim skrivenim slojem (Russel & Norvig, 1995).

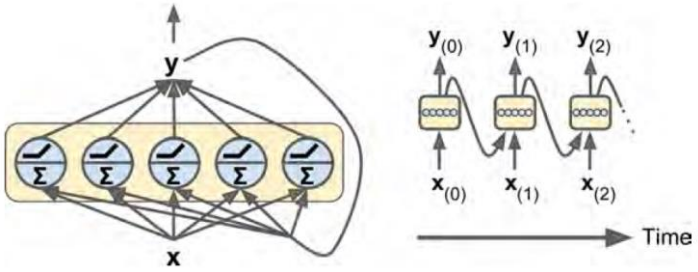
4.4.3.3 Povratne neuralne mreže

Ova vrsta neuralnih mreža po strukturi slična je feed-forward mrežama, s tim da je izlaz svakog neurona povezan s njegovim ulazom, kako je pokazano na slici 60. Za svaki vremenski korak t ovakav neuron primi ulazni podatak $x_{(t)}$ i svoj izlazni podatak iz prethodnog vremenskog koraka $y_{(t-1)}$.



Slika 60: Povratni neuron (lijevo), prikazan kroz vrijeme (desno) (Geron, 2017).

Sloj povratnih neurona prikazan je na slici 61. U ovom slučaju ulaz i izlaz su vektori: u svakom vremenskom koraku *t* svaki neuron prima ulazni vektor $\mathbf{x}_{(t)}$ i izlazni vektor prethodnog vremenskog koraka $\mathbf{y}_{(t-1)}$.



Slika 61: Sloj povratnih neurona (lijevo), prikazan kroz vrijeme (desno) (Geron, 2017).

Izlaz jednog sloja povratne mreže računa se po formuli

$$\mathbf{y}_{(t)} = \phi(\mathbf{W}_x^T \mathbf{x}_{(t)} + \mathbf{W}_y^T \mathbf{y}_{(t-1)} + \mathbf{b})$$

gdje su $\mathbf{W}_x$ i $\mathbf{W}_y$ tegovne matrice, **b** utjecajni vektor (engl. *bias vector*), a ϕ aktivacijska funkcija.

4.5 Alternativa neuralnim mrežama u prepoznavanju govora

Za prepoznavanje govora se osim neuralnih mreža upotrebljavaju i takozvani *skriveni Markovljevi modeli ili HMM* (engl. *Hidden Markov Models*). U (Vermeulen, Bernard, Yan, Fanty, & Cole, 1996) autori prikazuju rezultate usporedbe HMM s neuralnom mrežom u aplikaciji za upotrebu telefonskih servisa putem govora. Njihovi rezultati prikazani su u tablici 6.

Tablica 6: Relativne performanse sustava za prepoznavanje govora temeljen na HMM i neuralnoj mreži u prepoznavanju 58 riječi (Vermeulen, Bernard, Yan, Fanty, & Cole, 1996).

Sustav	Prepoznavanje
HMM	97.0%
Neuralna mreža	95.5%

Njihov je zaključak da iako HMM ima preciznije prepoznavanje rezultati su slični onima od neuralne mreže. Autori prednost daju neuralnoj mreži iz dva razloga:

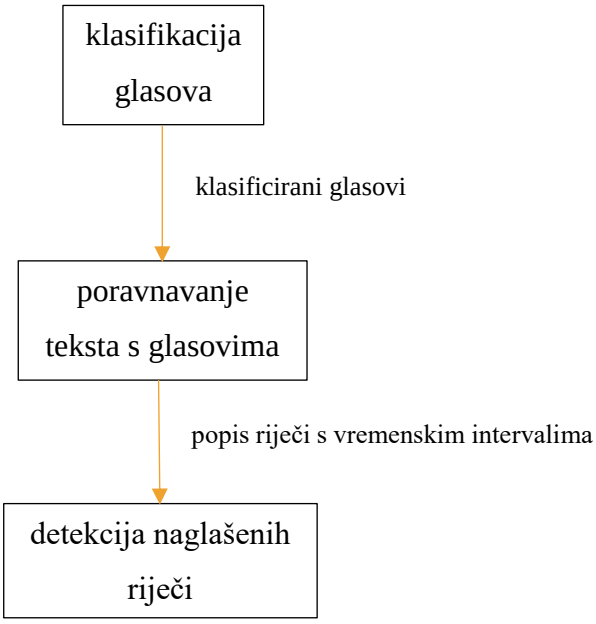
1. Sustav temeljen na neuralnoj mreži superioran je sa stajališta računalnih zahtjeva (računala i računalni programi). U ovom istraživanju autori su uglavnom imali u vidu hardver i softver za telefoniju.
2. Takav sustav omogućava dodavanje drugih funkcionalnosti kao što je identifikacija govornika i procjena pouzdanosti.

5. Metoda detekcije naglašenih riječi u zvučnom zapisu

Na temelju onoga što je prezentirano u prethodnim poglavljima, u ovom dijelu opisana je metoda pronalaženja naglašenih riječi u snimci govora s unaprijed napravljenim transkriptom. Ova se metoda općenito sastoji od pet koraka:

1. Segmentiranje govora.
2. Treniranje neuralne mreže.
3. Klasifikacija glasova (prepoznavanje govora).
4. Poravnavanje teksta s glasovima (da bi se odredile granice riječi).
5. Detekcija naglašenih riječi.

Prva dva koraka su priprema za ostale korake koji ovise o prepoznavanju fonema. Prvi je korak ujedno i vremenski najzahtjevniji jer je segmentiranje govora postupak koji se mora provoditi ručno i o čijoj preciznosti ovisi i kvaliteta treniranja neuralne mreže i njenog prepoznavanja fonema. Koraci 3, 4 i 5 sačinjavaju glavni dio postupka za detekciju naglašenih riječi. Općeniti postupak detekcije naglašenih riječi prikazan je na slici 62.

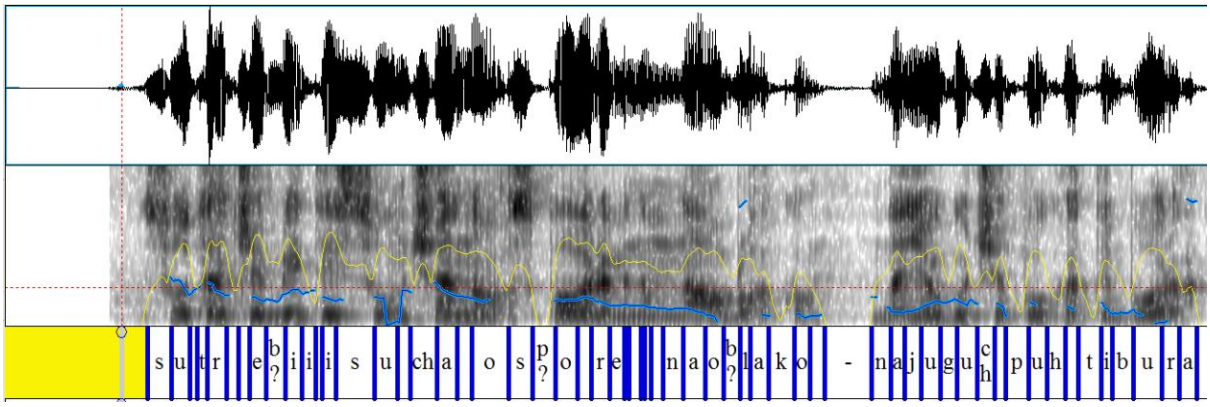


Slika 62: Općeniti postupak detekcije naglašenih riječi.

U sljedećem dijelu detaljno su opisani gornji koraci.

5.1 Segmentiranje govora

Da bi se iz nekog zvučnog zapisa govora mogli izdvojiti glasovi potrebno je taj zapis ručno podijeliti na segmente koji sačinjavaju pojedinačne glasove. Na slici 63 prikazan je jedan odsječak govora za koji su ručno označeni pojedinačni glasovi. Ovaj je proces vremenski zahtjevan jer, da bi se kvalitetno obavio, potrebno je pažljivo odrediti granice između glasova. Prema (Babić, i dr., 1991) prosječna brzina govora je 4 do 7 slogova u sekundi, dok je brzina govora na TV dnevnicima oko 6 slogova u sekundi. Ako se svaki slog u prosjeku sastoji od dva do tri glasa onda je u jednoj sekundi izgovoreno oko 15 glasova. To znači da segmentiranje govora koji traje 10 sekundi (bez stanki) zahtijeva ručno izdvajanje oko 150 glasova, a za jednu minutu oko 900 glasova.



Slika 63: Primjer segmentiranog govora.

Kod segmentiranja govora postoji par poteškoća. Jedna je *koartikulacija*, pojava kod koje glasovi u rečenici zbog utjecaja susjednih glasova zvuče drugačije od izolirano izgovorenih glasova. Taj je problem prisutan i kod automatskog prepoznavanja govora. Na primjer, glas *n* zvuči (i izgovara se) drugačije u riječi *onda* i u riječi *tanka* jer je u prvoj riječi sljedeći glas, *d*, dentalan, a drugoj je sljedeći glas, *k*, velaran (mekonepčani). Ovo je primjer *asimilacije* (Hardcastle & Hewlett, 2006). Zbog toga koartikulacija često stvara drugačiji spektralni oblik glasova koji su njome zahvaćeni, što otežava njihovo prepoznavanje na osnovu spektra. Ovdje se sada postavlja pitanje kako označiti neki glas koji ne zvuči isto kao njegov standardni oblik? U prethodnom primjeru, da li glas *n* treba u oba slučaja označiti kao *n*, jer se na tom mjestu u govoru stvarno nalazi glas *n*, ili ga treba označiti na način da se vidi da je na tom mjestu koartikulirani glas *n*? Sa stajališta treniranja neuralne mreže bolje je da se glasovi različitog

zvuka označavaju različito jer je tada manji broj varijanti istog glasa, što može olakšati klasifikaciju glasova. S druge strane, ako na mjestu gdje se nalazi glas *n* uvijek naznačimo da je to glas *n*, bez obzira na to da li on stvarno zvuči kao standardni *n*, onda neuralna mreža ima jednu veću kolekciju glasa *n* čiji se zvuk, a time i spektar, znatno razlikuje od jednog do drugog, pa je zbog toga potrebno imati veći skup takvih glasova za njeno treniranje.

Druga poteškoća segmentiranja govora je u tome što slušanjem nije uvijek lako utvrditi o kojem se glasu radi. S obzirom da je trajanje većine glasova ispod 100 milisekundi, slušanjem tako kratkih segmenata zvuka nije uvijek lako utvrditi koji je glas na tom mjestu izgovoren. Ovdje je često od pomoći spektrogram glasa na kojem se mogu vidjeti promjene pozicija formanata glasa, pa se u nekim slučajevima pomoću njih može utvrditi gdje neki glas počinje, a gdje završava. Na primjer, na slici 64 (stranica 76) označeni glas *e* (drugi *e* s lijeva) ima prilično jasno vidljive granice. Međutim, kod prvog glasa *e*, koji se nalazi odmah iza glasa *j*, početak je već teže uočiti zbog prijelaza iz *j* u *e*. Na slici 65 (stranica 76) prikazan je spektrogram riječi „aerodrom“ s izdvojenim diftongom „ae“ na kojem se vidi da prijelaz s „a“ na „e“ nije jasno određen, već je više postepen. Slično se vidi i na slici 66 (stranica 76) gdje je izgovorena riječ „rekreacija“, s postepenim prijelazom s „e“ diftonga „ea“ na „a“ istog diftonga „ea“.

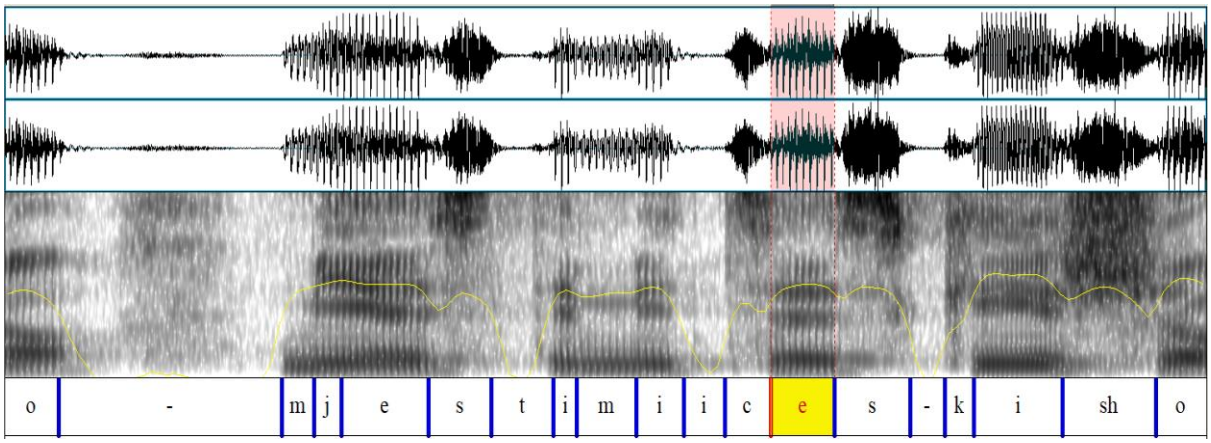
Općenito, kod nekih glasova je teže odrediti početak i kraj, ali kod nekih, kao što su tjesnačni i poluzatvorni frikativi, to je puno lakše jer je njihov spektar pomaknut na više frekvencije pa ih je zbog toga lakše uočiti i u frekvencijskoj i u vremenskoj domeni. Na slici 64 se lijevo i desno od označenog glasa *e* nalaze frikativi *c* i *s*, gdje se na spektrogramu jasno vidi pojačani intenzitet na višim frekvencijama, dok se na vremenskoj domeni prikazanoj iznad takvi glasovi pokazuju s većom gustoćom impulsa (opet zbog više frekvencije). Ovdje se jasno vidi i dio zvuka (eksplozija) za glas *k* koji se sastoji i od viših frekvencija, pa je lako vidljiv i u vremenskoj domeni.

S obzirom da u ovom istraživanju upotrebljavamo neuralnu mrežu za prepoznavanje fonema, segmentiranjem izdvajamo i označavamo foneme iz nekog skupa snimki koji smo izdvojili za treniranje neuralne mreže. Sljedeći dio sastoji se od izračunavanja usrednjenog spektra širine 100 Hz za svaki označeni segment, to jest fonem, upotrebom Praat programa. Tako dobiveni spektar sprema se u datoteku kao skup nizova spektralnih vrijednosti zajedno s oznakom glasa za svaki niz (ovaj format podataka je specifičan za Praat). Sljedeći isječak pokazuje jedan segment vrijednosti usrednjenog spektra za glas *d*:

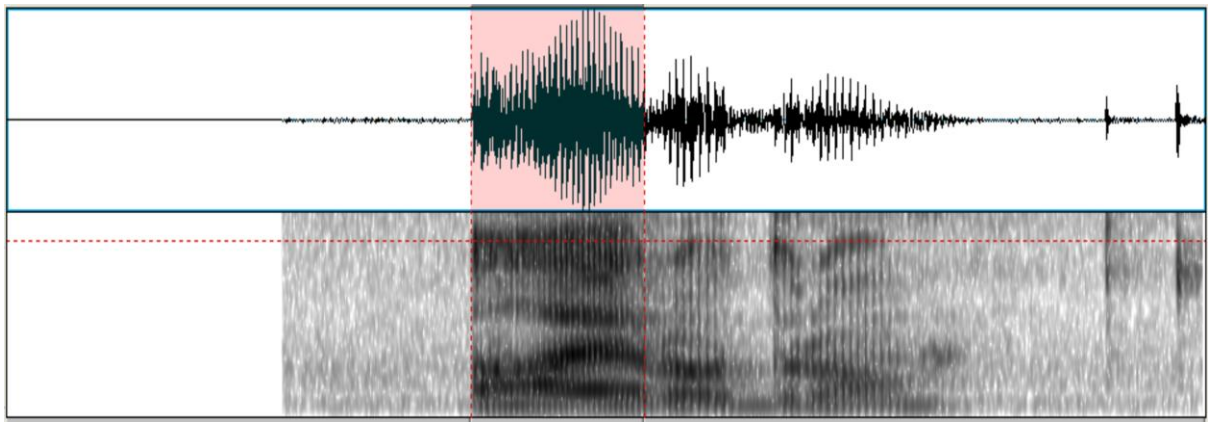
```
item [1]:
    class = "Ltas 2"
```

```
name = "d"
xmin = 0
xmax = 24000
nx = 240
dx = 100
x1 = 50
ymin = 1
ymax = 1
ny = 1
dy = 1
y1 = 1
z [] []:
    z [1]:
        z [1] [1] = 33.43887306745233
        z [1] [2] = 38.68299864713636
        z [1] [3] = 40.08951329682979
        z [1] [4] = 38.591847919544705
        z [1] [5] = 34.685218242860095
        z [1] [6] = 29.099934085175782
        z [1] [7] = 23.72663661647848
        z [1] [8] = 19.872319175298912
        z [1] [9] = 16.33655398488388
        z [1] [10] = 12.862920000393625
        z [1] [11] = 10.609115751237955
        z [1] [12] = 8.79262714473105
        z [1] [13] = 6.3514859830456905
        z [1] [14] = 3.734512867083653
        z [1] [15] = 2.161364040137169
        z [1] [16] = 1.6751215313612846
        z [1] [17] = 0.8208775999229656
        z [1] [18] = -1.495183274984121
        z [1] [19] = -4.486987352871267
        z [1] [20] = -1.4197395880879138
        z [1] [21] = 3.195387970254411
        z [1] [22] = 6.0904451032362985
        z [1] [23] = 6.979408369791833
        z [1] [24] = 5.444088562858926
        z [1] [25] = 1.5121595685181375
        z [1] [26] = -2.683734415328124
        z [1] [27] = -1.1288650675561065
        z [1] [28] = 1.003374426820192
          ...
        z [1] [235] = -61.4142337256387
        z [1] [236] = -63.96956649474305
        z [1] [237] = -63.31706132442125
        z [1] [238] = -62.6753219060982
        z [1] [239] = -64.45625779396737
        z [1] [240] = -69.50407949615783
```

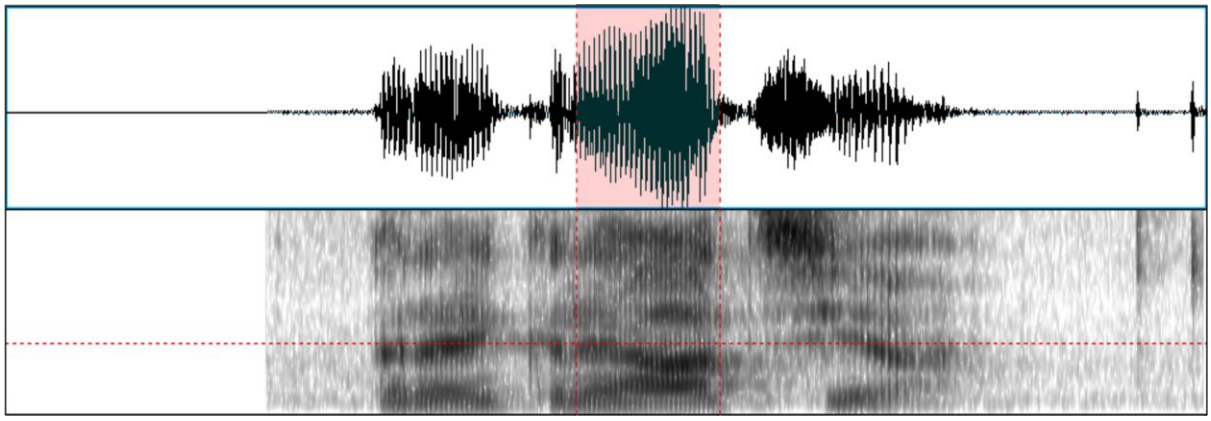
Niz vrijednosti spektra počinje nakon *z[1]* dijela i sastoji se od 240 vrijednosti, što odgovara frekvenciji od 0 – 24000 Hz (frekvencijski raspon spektralne analize). Pod oznakom *name* je postavljeno *d* – to je oznaka glasa koja je upisana za taj specifični dio zvuka prilikom segmentiranja govora. Primjer ovog postupka pokazan je na slici 67 (stranica 77). Ovako dobijeni podaci još nisu upotrebljivi za treniranje neuralne mreže, pa ih je potrebno transformirati. Taj je postupak opisan u sljedećem dijelu.



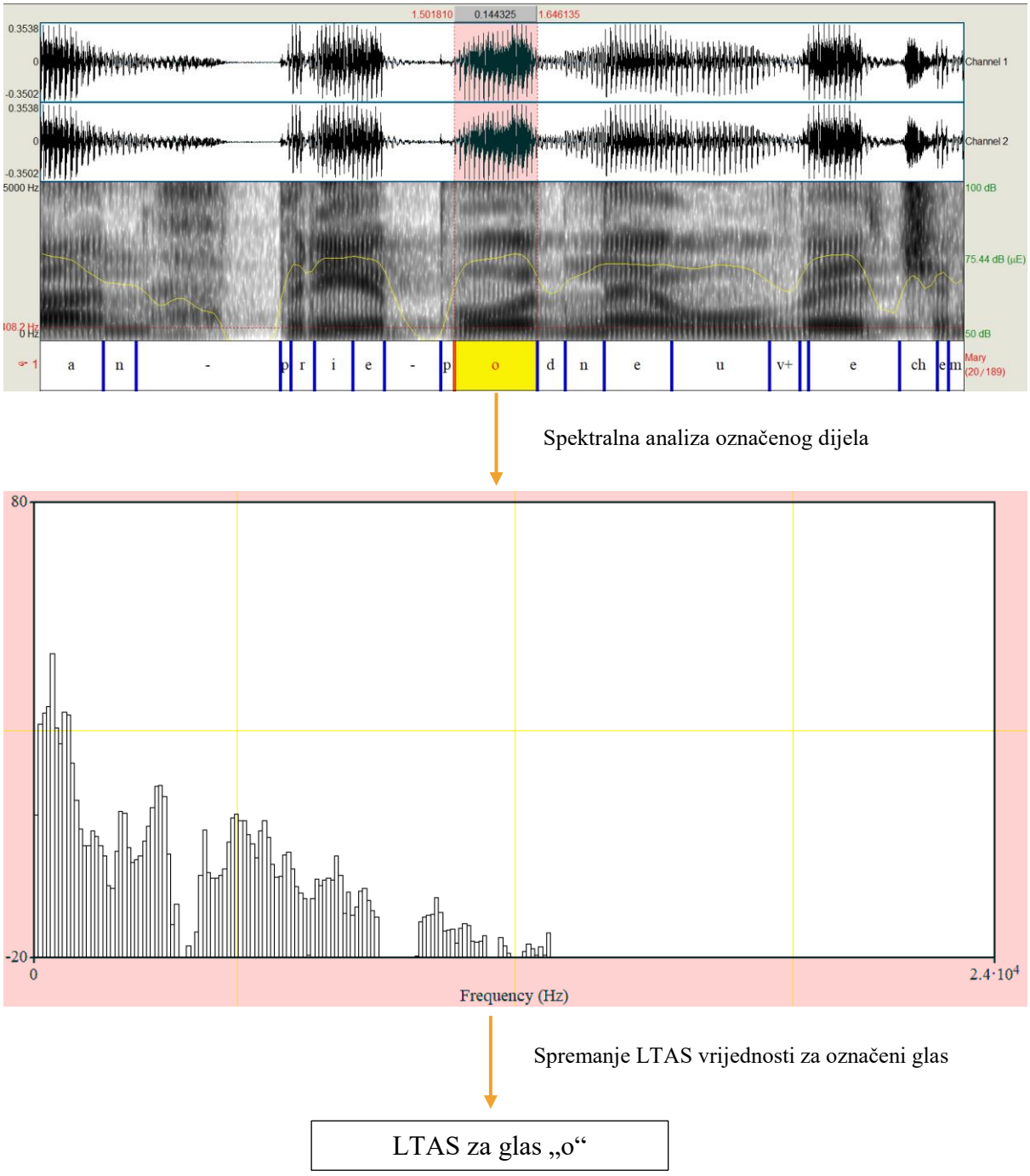
Slika 64: Segmentirani glasovi.



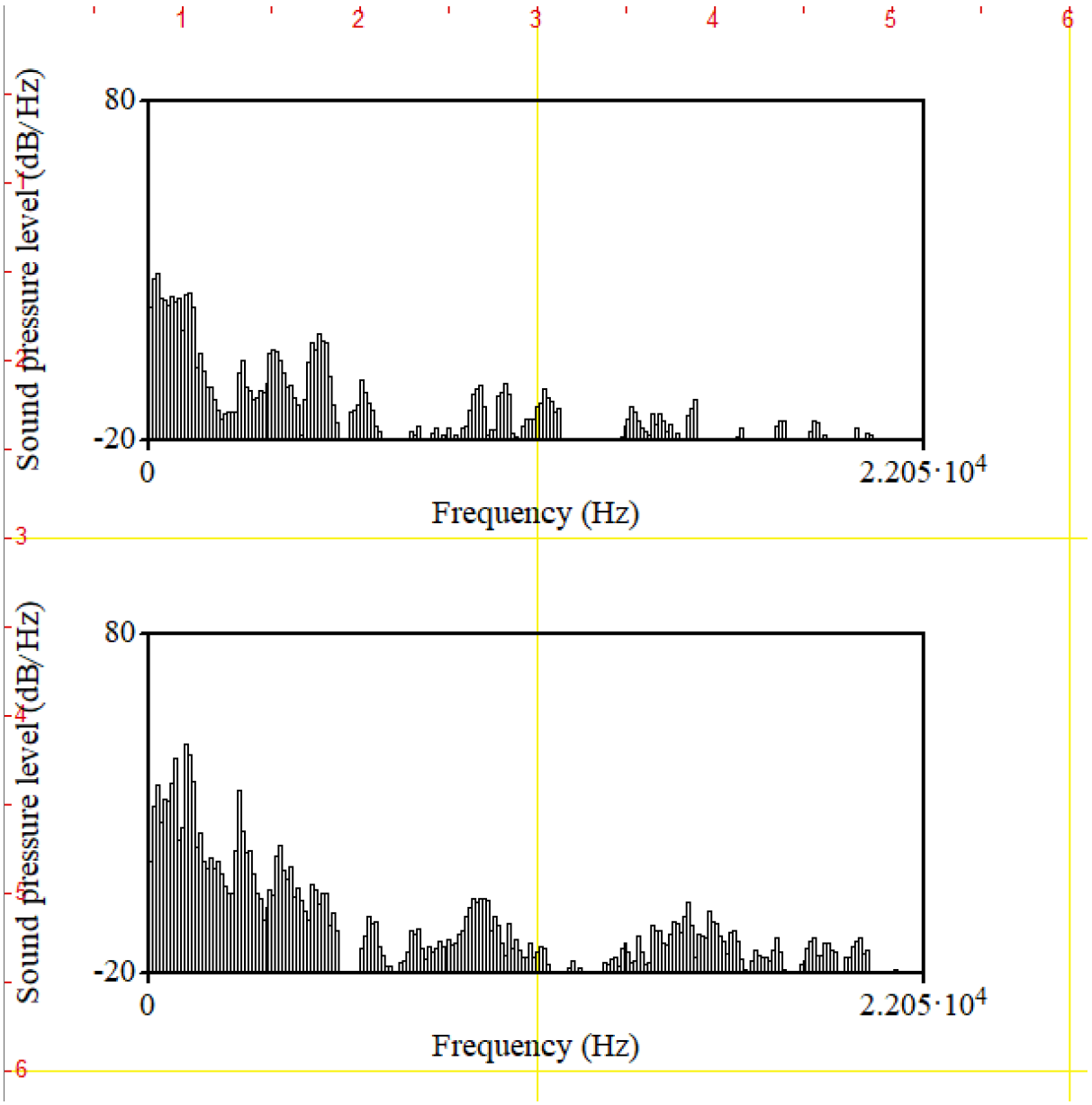
Slika 65: Oscilogram i spektrogram riječi „aerodrom“ (označeni dio obuhvaća dio riječi gdje je izgovoreno „ae“).



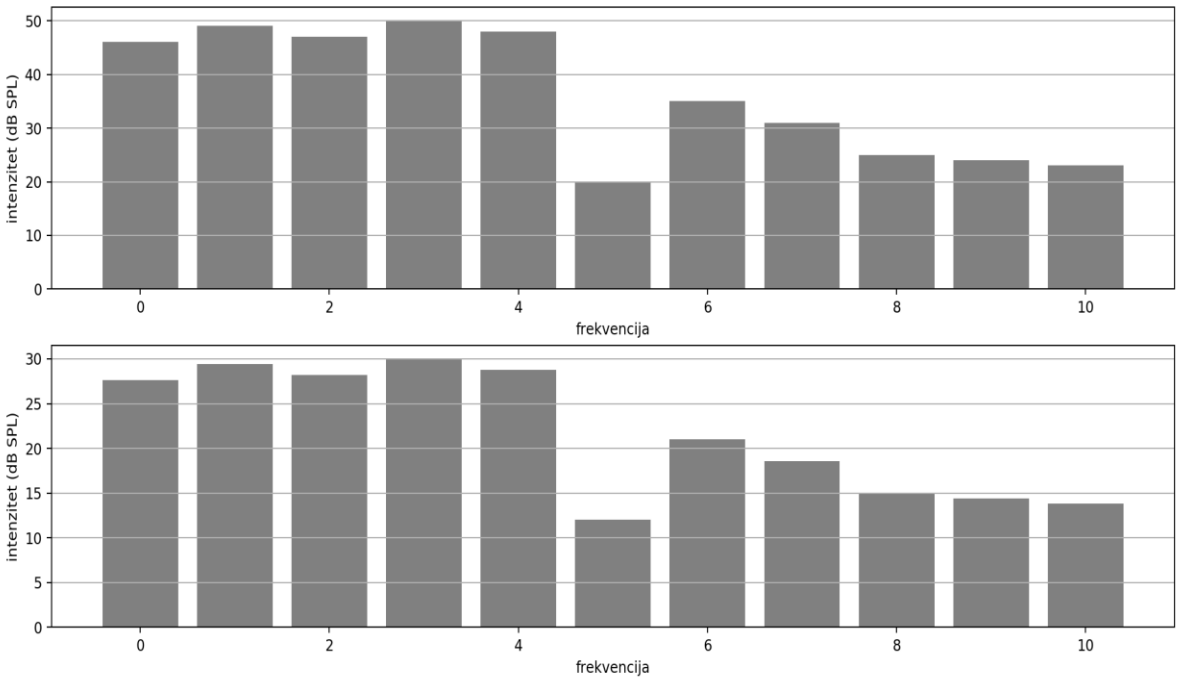
Slika 66: Oscilogram i spektrogram riječi „rekreacija“ (označeni dio obuhvaća dio riječi gdje je izgovoreno „ea“).



Slika 67: Postupak dobivanja vrijednosti usrednjenog spektra iz segmentiranog govora.



Slika 68: Spektar glasa „a“ izgovorenog slabijim intenzitetom (gornja slika) i jačim intenzitetom (donja slika).



Slika 69: Dio dvaju spektara s različitim maksimalnim intenzitetom. Vrijednosti gornjeg spektra kreću se do intenziteta 50 dB SPL, dok su vrijednosti donjeg spektra transformirane tako da se kreću do intenziteta 30 dB SPL, ali zadržavaju isti spektralni oblik.

5.2 Treniranje neuralne mreže

S podacima dobivenim u prethodnom koraku nije moguće trenirati neuralnu mrežu. Te je podatke potrebno prvo transformirati.

5.2.1 Priprema podataka

Prije treniranja neuralne mreže podatke je trebalo transformirati na dva načina:

1. Normalizacija, odnosno skaliranje vrijednosti na interval [0, 1].
2. Svođenje na isti intenzitet tako da svi glasovi izgledaju kao da su izgovoreni podjednakom glasnoćom.

Vrijednosti se moraju svesti na interval [0, 1] jer većina neuralnih mreža radi samo s takvim vrijednostima. Taj se postupak obično zove *normalizacija*. Svođenje na isti intenzitet radi se da zvučne karakteristike glasa ne bi ovisile o intenzitetu, nego samo o spektralnom obliku.

5.2.1.1 Normalizacija podataka

Normalizacija ili min-max skaliranje je postupak svođenja vrijednosti na interval [0, 1] koji se računa po formuli

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

gdje je x vrijednost u nizu kojeg skaliramo, X je niz vrijednosti s kojima radimo, a x' normalizirana vrijednost. Na primjer, za niz vrijednosti

[12, 4, 7, 20, 3, 18, 15, 10]

prvo odredimo najmanju i najveću vrijednost da bi dobili *min(x) = 3* i *max(x) = 20*. Ako sada za svaku vrijednost ovog niza primjenimo gornju formulu dobili bi novi niz u kojem su sve vrijednosti u intervalu [0, 1].

[0.52, 0.05, 0.23, 1, 0, 0.88, 0.7, 0.41]

Ovdje se vidi da je maksimalna vrijednost, 20, svedena na 1, a minimalna vrijednost, 3, na 0.

5.2.1.2 Normalizacija intenziteta

U normalnom govoru intenzitet stalno varira. S obzirom da je za treniranje neuralne mreže važan samo spektralni oblik glasova potrebno je eliminirati razlike u intenzitetu tako da se svi glasovi svedu na isti intenzitet. Na slici 68 (stranica 78) prikazana su dva spektra glasa „a“ istog govornika, gdje je na gornjoj slici taj glas izgovoren tiše, a na donjoj glasnije. Može se

primijetiti da su ova dva spektra po obliku slična (ako, na primjer, uzmemo u obzir pozicije prva četiri formanta). Međutim, spektar na donjoj slici izgleda „povišeno“ u odnosu na onaj na gornjoj slici, a razlog tome je jači intenzitet govora na donjoj slici. Ovo može negativno utjecati na treniranje neuralne mreže jer tada za isti glas ona mora uzeti u obzir ne samo spektralni oblik nego i intenzitet, što povećava raznolikost ulaznih vrijednosti za isti glas. Na slici 69 (stranica 79) prikazan je dio spektra. Na gornjoj slici maksimalni intenzitet je 50 dB. Na donjoj slici prikazan je isti spektar, ali s maksimalnim intenzitetom do 30 dB. Ovakva transformacija spektra dobijena je po formuli

$$x = \mathrm{x} - \mathrm{x} \cdot \frac{\max(s) - I_{max}}{\max(\mathrm{s})}$$

gdje je *x* vrijednost (u ovom slučaju vrijednost usrednjenog spektra) intenziteta, *s* je niz vrijednosti, a I_{max} maksimalni intenzitet niza *s*. Cilj ove transformacije je da se svaka vrijednost smanji ili poveća tako da međusobni odnos među vrijednostima ostane isti, čime se sačuva prvobitni spektralni oblik. Na slici 69 vidi se da je oblik dvaju spektara identičan, ali se razlikuju u maksimalnom intenzitetu.

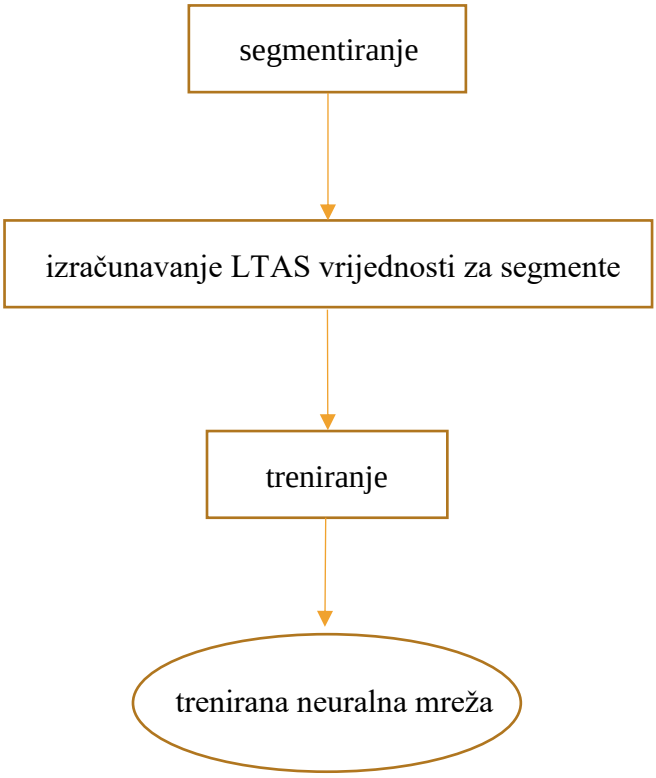
5.2.2 Postupak treniranja

Nakon transformacije podataka slijedi treniranje neuralne mreže. Pripremljene vrijednosti spektra, zajedno s oznakama glasova su ulazni podaci za mrežu, kako je ilustrirano na slici 71 (stranica 83). Općenito, svaki segment usrednjenog spektra je jedan ulazni parametar za neuralnu mrežu, tako da je sveukupno 120 ulaznih parametara. Postupak treniranja neuralne mreže sastoji se od sljedećih koraka:

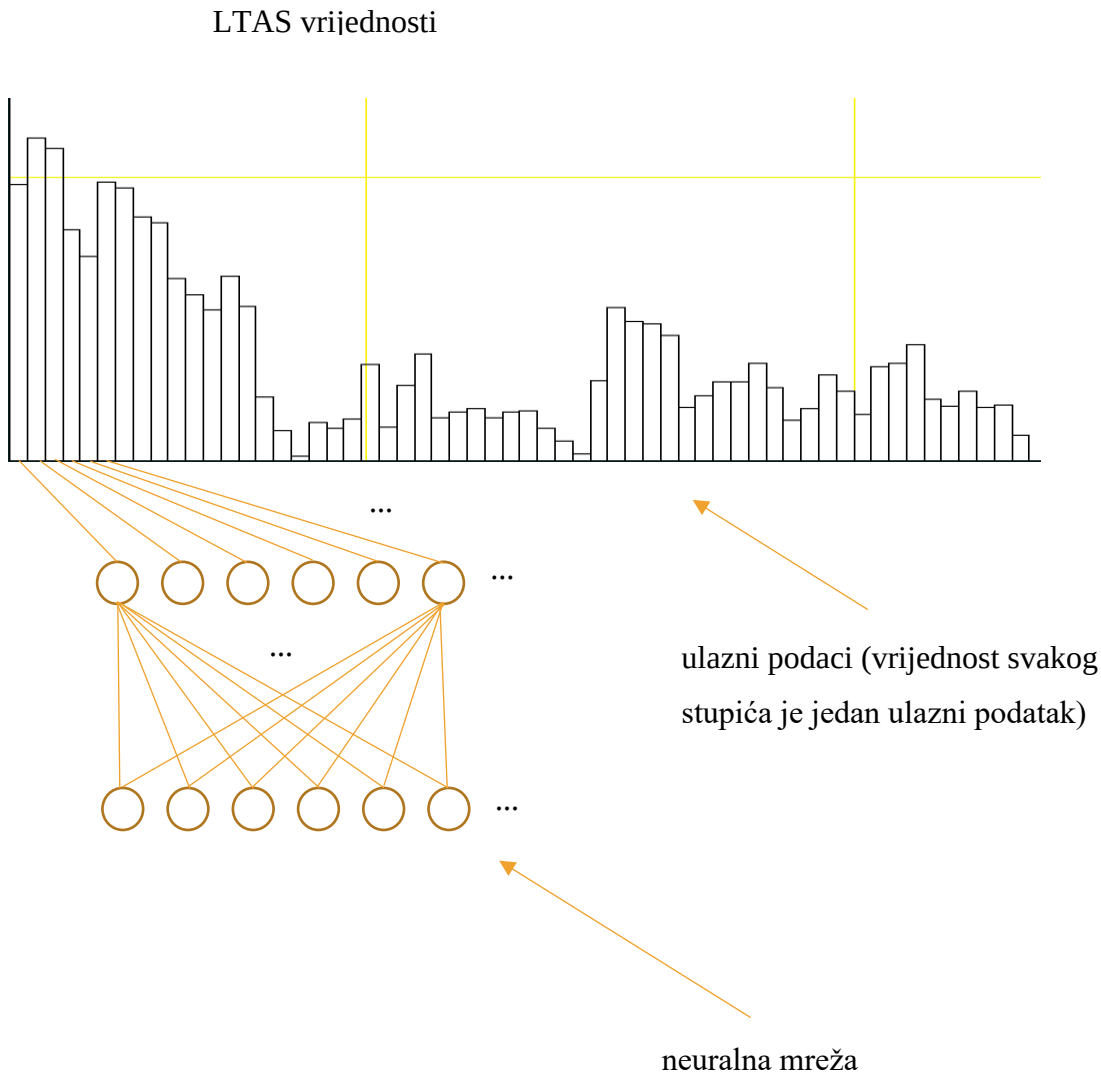
1. Za svaki segment snimke izračunaju se vrijednosti spektra i rezultat spremi u datoteku, zajedno s oznakom glasa za koji je taj spektar izračunat.
2. Transformiranje podataka.
3. Transformirane vrijednosti spektra za sve glasove iz segmentiranog zvuka učitaju se u listu koja predstavlja *skup za treniranje* (engl. *training set*). S tim se skupom vrijednosti pokrene treniranje neuralne mreže.

Rezultat koraka 3 je neuralna mreža trenirana s određenim brojem „primjera“ glasova. Treniranje neuralne mreže nije korak koji se izvodi samo jednom – ako je potrebno poboljšati priciznost prepoznavanja glasova ovaj se postupak ponavlja s novim, većim skupom vrijednosti za treniranje, što zahtijeva dodatno segmentiranje neke nove snimke ili podešavanje prepoznatog dijela zvuka nakon prolaska kroz neuralnu mrežu. Sam proces treniranja neuralne

mreže traje relativno kratko, zavisno od brzine računala i količine vrijednosti za treniranje. Cijelokupni postupak pripreme za treniranje i samog treniranja neuralne mreže prikazan je na slici 70.



Slika 70: Postupak treniranja neuralne mreže.



Slika 71: Princip treniranja neuralne mreže s vrijednostima usrednjenog spektra.

5.3 Prepoznavanje govora

Nakon što je neuralna mreža trenirana prepoznavanje se govora svodi na klasifikaciju glasova iz snimke. Ovaj se postupak sastoji od dva dijela. Prvi je klasifikacija glasova, a drugi grupiranje klasificiranih glasova prema određenim kategorijama.

5.3.1 Klasifikacija glasova

Klasifikacija glasova sastoji se od dva koraka:

1. Snimka se segmentira na dijelove od 10 milisekundi, gdje se za svaki dio napravi spektralna analiza.
2. Za snimku se odrede pozicije pulseva koji označavaju prisutnost glotalnog zvuka (ovu funkcionalnost Praat ima ugrađenu). Za svaku se od tih pozicija odredi segment od 10 milisekundi (5 milisekundi lijevo i desno od zadane pozicije). Nakon toga se za svaki taj dio napravi spektralna analiza.

Drugi korak se radi zbog toga što se u prvom koraku lako može „preskočiti“ neki bitan dio zvuka, primjerice, ako smo jedan glas „uhvatili“ pri kraju ili na prijelazu u sljedeći glas. Zbog toga nam drugi korak daje dodatne informacije o mjestima na kojima postoji zvuk. Isto tako, raditi samo drugi korak ne bi dalo dobre rezultate jer pulsevi nisu prisutni kod bezvučnih frikativa, što se vidi na slici 73 (stranica 86).

Rezultati klasifikacije glasova u ova dva koraka se kombiniraju prema vremenu pojavljivanja tako da se dva niza glasova spoje u jedan. Ova su dva koraka ilustrirana na slikama 72 i 73. Na primjer, rezultat jednog dijela klasifikacije na osnovu ulaznih segmenata od 10 milisekundi izgleda ovako (u zagradi je početno vrijeme u kojem se taj segment pojavljuje):

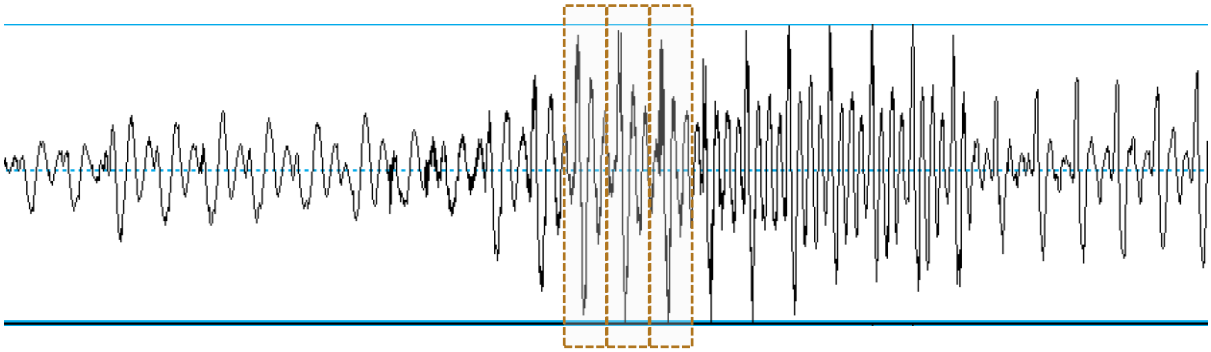
```
t (0_0450)
v (0_0550)
s (0_0650)
s (0_0750)
s (0_0850)
s (0_0950)
s (0_1050)
t (0_1150)
s (0_1250)
t (0_1350)
t (0_1450)
g (0_1550)
u (0_1650)
o (0_1750)
u (0_1850)
…
```

Isto tako, rezultat klasifikacije na osnovu pulseva izgleda ovako:

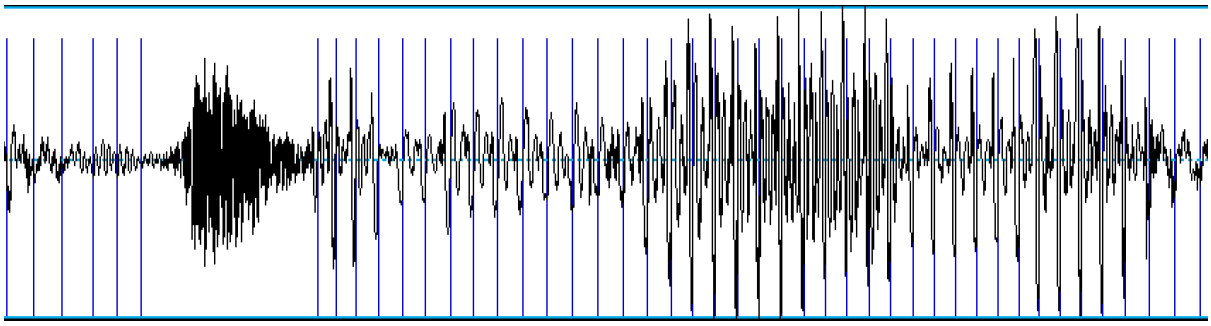
```
u (0_1681)
u (0_1785)
u (0_1890)
o (0_1997)
u (0_2105)
o (0_2211)
o (0_2317)
o (0_2423)
a (0_2529)
a (0_2638)
e (0_2750)
e (0_2867)
e (0_2989)
e (0_3111)
e (0_3227)
…
```

Spajanjem ova dva niza dodaju se elementi jednog u drugi tako da su sortirani prema vremenu. Gornja dva niza spojena na ovaj način izgledaju ovako:

```
t (0_0450)
v (0_0550)
s (0_0650)
s (0_0750)
s (0_0850)
s (0_0950)
s (0_1050)
t (0_1150)
s (0_1250)
t (0_1350)
t (0_1450)
g (0_1550)
u (0_1650)
u (0_1681)
o (0_1750)
u (0_1785)
u (0_1850)
u (0_1890)
o (0_1997)
u (0_2105)
o (0_2211)
o (0_2317)
o (0_2423)
a (0_2529)
a (0_2638)
e (0_2750)
e (0_2867)
e (0_2989)
e (0_3111)
e (0_3227)
…
```



Slika 72: Segmenti zvuka od 10 milisekundi.



Slika 73: Oznake pulseva (okomite crte).

5.3.2 Grupiranje klasificiranih glasova

Obzirom da neuralna mreža neće svaki glas ispravno klasificirati, glasove smo grupirali u kategorije, što je drugi dio postupka prepoznavanja govora. Ovo smo napravili zbog dva razloga:

1. Obzirom da smo zvuk podijelili na segmente od 10 milisekundi neuralna mreža je davala veliki broj rezultata klasifikacije, po jedan za svaki taj segment. To bi za potrebe poravnavanja govora s tekstom stvaralo poteškoće i sa stajališta performansi i sa stajališta preciznosti.
2. Taj velik broj glasova semantički ne odgovara onome što je izrečeno. Na primjer, jedan niz od 10 glasova klasificiranih kao glas *a* je u stvari jedan glas *a*, pa se zbog toga taj cijeli niz može interpretirati kao jedan glas.

U ovom dijelu opisan je postupak grupiranja glasova. Ispod je prikazan ispis rezultata klasifikacije glasova gdje je izgovoreno *„u ime tih važnijih stvari“*.

```
 0:  7.31      7.3826    72.6    uuuiiiiiiiiiiiii
 1:  7.4024    7.4541    51.7    *mmmmn
 2:  7.46      7.5237    63.7    eieeeeeeeeiiii
 3:  7.5343    7.5444    10.1    *nm
 4:  7.59      7.6367    46.7    iiiiiiiiii
 5:  7.7387    7.7488    10.1    *mm
 6:  7.7596    7.8737    114.1   oouuouoooooaaaaaaaaaaaaa
 7:  7.8966    7.9115    14.9    aaaau
 8:  7.93      7.9434    13.4    žžžž
 9:  7.9514    7.98      28.6    čžšžšžž
10:  7.9939    8.0293    35.4    *nmnnn
11:  8.0397    8.12      80.3    eiiieiiiiii
12:  8.15      8.21      60.0    sssssss
13:  8.2595    8.2595    0.0     *n
14:  8.2709    8.43      159.1   oooaaaaaaaaaaaaaau
15:  8.45      8.57      120.0   eeiiiiiieieieiiiiiiiii
16:  8.573     8.5846    11.6    *nmnn
```

Prvi stupac je redni broj segmenta, drugi i treći početak i kraj, četvrti ukupno trajanje segmenta (razlika vrijednosti drugog i trećeg stupca), a peti je grupa glasova kao rezultat klasifikacije. Glasovi su grupirani prema kategorijama glasova koje je neuralna mreža najbolje prepoznavala:

- Vokali (*i, e, a, o, u*)
- Frikativi *š* i *č*
- Frikativi *c* i *s*
- Frikativi *đ, ž* i *z*

Također se grupiraju svi ostali glasovi koji se vremenski nalaze između glasova gornje tri kategorije. Na primjer, u redu 10 grupirani su glasovi *m* i *n*. U redovima 0 do 2 koji označavaju segment govora u vremenskom intervalu 7.31 do 7.52 izgovorena je riječ „ime“, u redovim 3 do 5 (vremenski segment 7.53 do 7.74) riječ „tih“, u redovima 6 do 11 (vremenski segment 7.75 do 8.12) riječ „važnijih“, a u ostatku ispisa riječ „stvari“. U prvom slučaju, za riječ „ime“ vidi se da je neuralna mreža uglavnom korektno klasificirala većinu glasova, s tim da je kod glasa *m* u jednom dijelu tog cijelog segmenta pogrešno prepoznala glas *n*. U drugom slučaju, za riječ „tih“ neuralna mreža je korektno klasificirala samo glas *i*, ali je „pogriješila“ kod ostalih glasova. U trećem slučaju, za riječ „važnijih“ nekih glasova uopće nema, kao što su *v*, *j* i *h*. Razloga za to može biti više – od toga da su neki od tih glasova bili previše tiho izgovoreni pa ih neuralna mreža nije prepoznala do toga da zbog koartikulacije ili prijelaza iz jednog glasa u drugi nisu bili jasno izgovoreni ili nisu bili uopće izgovoreni. Takvi su slučajevi u govoru česti. Isti je slučaj s riječi „stvari“ u zadnjem dijelu ispisa. Nadalje, razlike između *č* i *ć*, te *dž* i *đ* nisu uzete u obzir iz tri razloga:

1. One se po zvuku slabo razlikuju pa bi process prepoznavanja time bio otežan.
2. Obzirom da je na raspolaganju tekst onoga što je izgovoreno lako je ustanoviti o kojem glasu se radi nakon što je riječ prepoznata.
3. Većina govornika hrvatskog jezika ionako ne pravi razliku između ovih glasova (Babić, i dr., 1991).

Nakon što je dobijen popis grupa glasova kao u gornjem ispisu slijedi korak poravnavanja teksta (podnatpisa) s grupama glasova iz takvog popisa.

5.4 Poravnavanje teksta s govorom

Ova je metoda zamišljena tako da radi s ulaznim podacima koji se mogu podijeliti u dvije skupine:

- Snimka govora (podaci o zvuku)
- Popratni tekst onoga što je izgovoreno (podnatpisi)

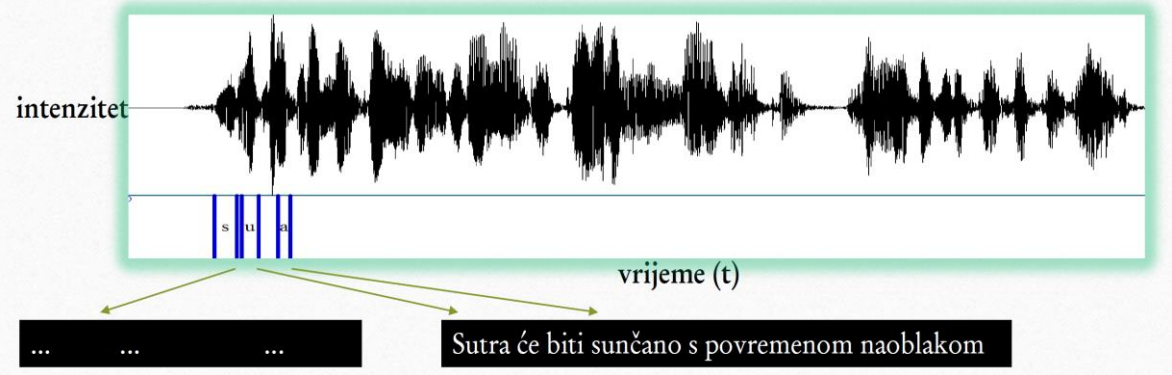
Pretpostavka je da tekst onoga što je izgovoreno neće biti savršeno poravnat s govorom. Iako je osnovna ideja ta da je tekst uključen u obliku podnatpisa, to općenito nije uvjet, odnosno tekst može biti zadan i zasebno, u kojem slučaju je potrebno ručno otprilike naznačiti početak i kraj tog teksta u zvuku da bi se kasnije unutar tog zvučnog intervala mogle odrediti granice pojedinačnih riječi, odnosno poravnati riječi teksta s govorom. To je poravnanje nužno da bi

ova metoda dobro radila zato jer se ona zasniva na dijelomično prepoznatim riječima, pa je prisutnost teksta u okolini onoga što je izgovoreno važna. Slika 74 prikazuje izgovorene rečenice *„Sutra će biti sunčano s povremenom naoblakom. Na jugu će puhati bura“*. Na početku te snimke označeni su neki glasovi riječi „sutra“. Ako na osnovu tih glasova želimo zaključiti kojoj riječi bi oni mogli pripadati moramo uzeti u obzir tekst koji se vremenski nalazi u okolini tih glasova, primjerice, jednu ili dvije sekunde prije ili poslije trenutka kada su oni izgovoreni.

Poravnanje (engl. *alignment*) je process u kojem je cilj odrediti početak i završetak nekog dijela govorne snimke koji odgovara tekstu čiju poziciju na snimci želimo odrediti. Poravnanje je ključni korak ovog istraživanja jer je preduvjet bilo kakve zvučne analize neke izgovorene riječi taj da znamo gdje se ona pojavljuje na snimci. Preciznost poravnanja ovisi o informacijama koje možemo dobiti iz analize snimke, a te informacije ovise o točnosti prepoznavanja pojedinačnih glasova. S obzirom da je cilj ovog istraživanja upotreba *djelomično* trenirane neuralne mreže, odnosno djelomično prepoznavanje govora, takva mreža će dobar dio glasova prepoznati pogrešno. Ovdje je jasno da je značenje pojma *djelomično* relativan: Što je mreža trenirana s više glasova to će ih bolje prepoznavati, a time će i poravnanje teksta s govorom davati bolje rezultate. Međutim, opširnije treniranje neuralne mreže zahtijeva segmentiranje veće količine snimljenog govora, što je za većinu jezika vremenski zahtjevan posao. Jedan od ciljeva ovog istraživanja je vidjeti koliko se može dobiti iz neuralne mreže koja je trenirana s relativno malim brojem segmentiranog teksta, zbog čega je neophodno na što bolji način iz djelomično prepoznatih glasova zaključiti o kojem dijelu snimke se radi, odnosno koja riječ teksta pripada kojem segmentu snimke. Upravo to je cilj postupka poravnanja teksta s govorom koji je ovdje opisan.

Prema tome, osnovni postupak ovog istraživanja može se definirati ovako:

1. Naći koji dio govornog signala (snimke) odgovara kojem nizu riječi popratnog teksta;
2. Analizom tog dijela govornog signala utvrditi je li neka riječ naglašena.



Slika 74: Govorna snimka s podnatpisima (ili tekstom).

Postupak poravnanja možemo općenito definirati kao postupak kojem je cilj za dva stringa, s_1 i s_2, gdje je s_1 duži od s_2, pronaći pozicije slova u s_2 (ali tako da njihov poredak ostane isti) tako da se podudaraju sa slovima na istoj poziciji u s_1. Na primjer, neka je s_1 string „jadran“, a s_2 string „ada“. Poravnjanjem ova dva stringa dobije se sljedeće:

„j**ad**r**a**n“
„_**ad_a**“

Crtice između slova u donjem stringu označavaju prazan prostor. Ovdje se prvi *a* u donjem stringu nalazi na poziciji 0, pa je potrebno sadržaj tog stringa od te pozicije pomaknuti za jedno mjesto u desno da bi prvi *a* došao na poziciju 1 jer se na toj pozicij u gornjem stringu nalazi *a*. Ovim je i slovo *d* došlo na poziciju 2, gdje se već poklapa s istim slovom gornjeg stringa. Drugo slovo *a* donjeg stringa, međutim, ostalo bi ispod slova *r* gornjeg stringa (na poziciji 3), pa se ono moralo pomaknuti jedno mjesto u desno, na poziciju 4 na kojoj se nalazi drugo slovo *a* gornjeg stringa, odnosno sadržaj tog stringa od pozicije 3 morao se pomaknuti za jedno mjesto u desno da bi se poklapalo drugo slovo *a* jednog i drugog stringa. Općenito, ovim se postupkom slova kraćeg stringa razmiču da bi time došla na odgovarajuću poziciju.

Prethodno opisani postupak u ovom radu služi kao osnova za poravnavanje teksta s govorom, međutim on u tom osnovnom obliku nije prihvatljiv iz nekoliko razloga:

- Ako pretpostavimo da je dulji string onaj koji je rezultat klasifikacije glasova (jer glasova obično ima više od slova), ti glasovi će biti u grupama (kako je prethodno opisano), pa zbog toga više ne radimo s dva stringa. U tom slučaju jedno slovo moramo usporediti s jednom grupom glasova, a ne s drugim slovom. Nadalje, s obzirom da neuralna mreža neke kategorije glasova prepoznaje bolje od drugih postupak

poravnavanja treba na neki način to uzeti u obzir. U ovom radu postupak poravnavanja upotrebljava sustav bodovanja koji nekim kombinacijama pozicija slova daje više bodova nego drugim.

- Popis glasova ne mora sadržavati sva slova koja se nalaze u tekstu. Primjerice, popis grupa glasova može biti *[szs][ao][mn][aaao]*, a tekst *soba*. Iako u popisu glasova ne postoji *b*, poravnanje bi trebalo (ili moglo) biti *s-[szs]*, *o-[ao]*, *b-?* i *a-[aaao]*. Općenito, ovakvi slučajevi su česti, pa zbog toga postupak poravnavanja treba naći kombinaciju koja ispravno poravnava najveći dio teksta.
- Prosječan podnatpis može sadržavati desetak riječi, a to može značiti 40, 50 ili više grupa glasova. Isprobavanje svih mogućih kombinacija ne bi bilo efikasno jer kod duljih stringova taj broj može biti jako velik (u stvari može biti toliko velik da algoritam postane neupotrebljiv). Zbog toga algoritam treba biti dovoljno efikasan da može raditi s tekstovima duljine desetak riječi.

Naivan algoritam mogao bi raditi tako da generira i analizira sve pozicije kraćeg stringa dok ne poravna sva slova ili dok ne iscrpi sve mogućnosti. U sljedećem primjeru prikazan je ovaj postupak s tekstom „soba“ i sedam grupa glasova. Početno poravnanje slova s grupama je

[zsz] [mn] [ao] [rrr][eaaa][rvrv] [eae]
s o b a

U tablici 7 (stranica 108) prikazane su sve moguće kombinacije pozicija slova donjeg stringa koje bi ovakav algoritam generirao. U ovom primjeru može se uočiti nekoliko problema koji algoritam implementiran na takav način čine nepraktičnim:

- Efikasnost: Za sedam grupa glasova i četiri slova generirano je 35 kombinacija. Za niz od tridesetak grupa i dvadesetak slova broj kombinacija može narasti na desetke miliona zato jer se velik broj kombinacija ponavlja. Primjerice, u tablici 7 kombinacija s *ba* počevši od stupca *[eaaa]* ponavlja se šest puta. Kod dužih nizova ta ponavljanja generiraju eksploziju kombinacija.
- Neke kombinacije su manje vjerojatne od drugih. U tablici 7 u retku 4 slovo *a* je poravnato s posljednjom, sedmom grupom, dok je prethodno slovo, *b*, poravnato s trećom grupom. S obzirom da glasovi tih grupa dolaze iz govornih segmenata od 10 milisekundi malo je vjerojatno da postoji „rupa“ od tri grupe glasova između *b* i *a*. To znači da se na ovaj način generiraju kombinacije koje nisu moguće, čime se dodatno usporava postupak poravnavanja.

Iako algoritam prikazan u tablici 7 generira sve moguće kombinacije pozicija slova teksta, u boljoj implementaciji mogu se dobiti sve te kombinacije, ali istovremeno izbjeći to da ih se nanovo generira.

Za postupak poravnanja zvuka s tekstom postoji nekoliko relevantnih radova. U (Moreno & Alberti, 2009) opisana je metoda poravnanja teksta s dugačkim snimkama govora upotrebom faktorskih automata. U (Stan, Bell, & King, 2012) opisana je metoda za isti problem, ali gdje u tekstu ima grešaka i bez prethodne upotrebe akustičkog modela jezika. U (Moreno, Joerg, Van Thong, & Glickman, 1998) opisana je metoda poravnanja snimke govora s tekstom upotrebom rekurzivne tehnike koja radi i sa signalom u kojem ima šuma ili buke. U (Bordel, Nieto, Penagarikano, Rodriguez-Fuentes, & Varona, 2012) opisana je metoda poravnanja teksta sa snimkama govora u trajanju od oko 3 sata, bez upotrebe jezičnog ili akustičkog modela. U (Anguera, Luque, & Gracia, 2014) i (Huang, 2003) opisana je metoda poravnjanja upotrebom ograničenih resursa, bez upotrebe akustičkih modela ili baza podataka s pripremljenim rezultatima treniranja. U ovom se radu upotrebljava metoda prepoznavanja govora zasnovana na grafemima. U (Hazen, 2006) opisana je metoda poravnanja i korekcija transkripta za dugačke snimke govora. U (Caseiro, Meinedo, Serralheiro, Trancoso, & Neto, 2002) poravnanje se upotrebljava za potrebe indeksiranja sadržaja knjiga, a u (Hoffmann & Pfister, 2013) opisan je pristup poravnanju upotrebom skrivenih markovljevih modela (HMM).

5.4.1 Pregled algoritama za približno poravnavanje stringova

Postupak poravnavanja teksta s govorom može se svesti na postupak tzv. *približnog poravnavanja stringova* (engl. *approximate string matching*). U ovom dijelu dat je pregled nekih osnovnih algoritama za približno poravnavanje stringova prema (Navarro, 2001).

Približno poravnanje stringova je poravnanje koje dozvoljava greške. Cilj ovih algoritama je naći dio teksta u kojem se pojavljuje zadani uzorak (koji je i sam tekst), s tim da se dozvoli određen broj grešaka. Model greške je način na koji se utvrđuje koliko se dva stringa razlikuju. Ovakvi se algoritmi često koriste u područjima kao što su

- Pretraživanje teksta
- Obrada signala
- Bioinformatika
- Prepoznavanje govora
- ... i mnoga druga područja.

Ključni pojam kod ovakvih algoritama je pojam *udaljenosti* ili distance koji se precizno može definirati na sljedeći način: „Udaljenost $d(x, y)$ između dva stringa x i y je minimalna cijena niza *operacija* koje transformiraju x u y. Cijena niza operacija je zbroj cijena individualnih operacija u tom nizu. Operacije su konačni skup pravila oblika $\delta(z, w) = t$, gdje su z i w različiti stringovi, a t je pozitivan realni broj. Jednom kad operacija konvertira podstring z u w, niti jedna druga operacija više nije dozvoljena nad w“.

Pod pojmom operacije podrazumijeva se jedno ili više sljedećeg:

- Umetanje (slova na neko mjesto u stringu)
- Brisanje
- Supstitucija (zamjena)
- Preokretanje ili transpozicija (na primjer, *ab* u *ba*)

Funkcije ili mjere udaljenosti koje se često upotrebljavaju su:

- *Levenshteinova* ili *edit-distanca*. Ova funkcija udaljenosti dozvoljava umetanja, brisanja i supstitucije. U općem obliku sve ove operacije imaju cijenu 1. Općenito, ova se distanca može definirati kao minimalni broj umetanja, brisanja i supstitucija koji bi dva stringa učinili jednakima. Na primjer, stringovi *promet* i *proba* imaju edit-distancu 3 jer je potrebno jedno slovo obrisati ili umetnuti i dva slova zamijeniti, što

znači jedno umetanje i dvije supstitucije. Kada se radi o pretraživanju teksta onda se ova udaljenost zove „podudaranje stringova s *k* razlika“.

- *Hammingova distanca*. Ova funkcija udaljenosti dozvoljava samo supstitucije, koje u općem obliku imaju cijenu 1. Ova funkcija podrazumijeva da su oba stringa iste duljine. Hammingova distanca je tada broj pozicija u kojima se dva string razlikuju. Na primjer, stringovi *proba* i *torba* imaju hammingovu distancu 3 jer se razlikuju u svim pozicijama prva tri slova. Kada se radi o pretraživanju teksta ova se funkcija naziva „podudaranje stringova s *k* nepodudaranja“.
- *Epizodna distanca*. Ova funkcija dozvoljava samo umetanja, s cijenom 1. Za tekst T duljine *n* i epizodu P duljine *m* ova funkcija pronalazi sve najkraće podstringove teksta T koji sadrže P kao podniz.
- *Distanca najdužeg zajedničkog podniza (engl*. Longest Common Subsequence, LCS). Ova funkcija dozvoljava samo umetanja i brisanja, s cijenom 1. Zove se tako zbog činjenice da ona označava duljinu najdužih podnizova u oba stringa koji se poklapaju. LCS distanca je broj znakova koji se ne poklapaju na ovaj način. Ova se funkcija često upotrebljava kod programa koji uspoređuju dva tekstualna sadržaja da bi odredili gdje su razlike, a ima i primjenu u bioinformatici.

Od svih ovih funkcija udaljenosti za ovo istraživanje je najrelevantnija edit-distanca zato jer ona uzima u obzir i umetanja i brisanja i supstitucije, što je potrebno za rad s podacima koje dobivamo od neuralne mreže kao rezultat klasifikacije glasova. Ova funkcija je jedna od više proučavanih i može se podijeliti na *opću edit-distancu* ako svaka operacija (umetanje, brisanje i supstitucija) ima različitu cijenu ili *jednostavnu edit-distancu* ili samo *edit-distancu* ako svaka operacija ima cijenu 1. U ovom slučaju jednostavno tražimo minimalni broj operacija potrebnih da bi dva stringa postala jednaka. Edit-distanca je jedna od više proučavanih funkcija za distancu jer je primjenjiva u velikom broju problema. Većina algoritama za ovu mjeru udaljenosti fokusirana je na jednostavnu edit-distancu, ali se oni lako mogu adaptirati na opći oblik. Nadalje, mnogi algoritmi napravljeni za izračunavanje edit-distance mogu se specijalizirati za izračunavanje nekih drugih mjera udaljenosti. Na primjer, ako za algoritam koji izračunava edit-distancu dozvolimo samo umetanja i brisanja po cijeni 1 dobit ćemo LCS za dva stringa. Isto tako, ako dozvolimo samo supstitucije dobit ćemo Hammingovu distancu. Algoritmi za edit-distancu mogu se proširiti tako da se doda operacija transpozicije, što je korisno za aplikacije za pretraživanje teksta.

Kao što postoje mnogi algoritmi za sortiranje, tako postoje i mnogi algoritmi za određivanje edit-distance između dva ili više stringova. U ovom dijelu prikazani su neki takvi algoritmi koji su podijeljeni u sljedeće skupine:

- *Algoritmi dinamičkog programiranja* – Ovi su algoritmi opisani u 5.4.1.1.
- Algoritmi s konačnim automatima – Ovi su algoritmi opisani u 5.4.1.2.
- *Bit-paralelni algoritmi* – Ovi su algoritmi zasnovani na iskorištavanju paralelizma računala u radu s bitovima. Na ovaj način broj operacija koji algoritam mora izvršiti može se smanjiti najviše *w* puta, gdje je *w* broj bitova u jednoj riječi (word) računala. S obzirom da su današnja računala uglavnom 64-bitna ovakvo je poboljšanje signifikantno.
- *Filtrirajući algoritmi* – Ova vrsta algoritama zasnovana je na činjenici da može biti jednostavnije odrediti da se neka pozicija u tekstu ne podudara, nego da se podudara s nekim uzorkom ili dijelom uzorka. Na primjer, ako se stringovi „pro“ i „blem“ nijedan ne nalaze u nekom dijelu teksta, onda se niti riječ „problem“ ne može tamo naći sa samo jednom operacijom jer samo jedna operacija ne može modificirati oba dijela ovog uzorka. Većina ovakvih algoritama rade na ovom principu tako da traže dijelove uzoraka kod kojih nema greške (to jest, ne moraju se modificirati jednom od operacija).

S obzirom da za ovaj rad paralelni i filtrirajući algoritmi nisu relevantni u sljedeća dva dijela ukratko su opisani i ilustrirani po jedan algoritam iz prvih dviju skupina za izračunavanje edit-distance.

5.4.1.1 Algoritmi dinamičkog programiranja

Dinamičko programiranje je tehnika definiranja algoritma kojom se problem rastavlja na više manjih, jednostavnijih problema koji se onda dalje rastavljaju na manje probleme, sve dok te manje probleme ne može riješiti direktno. Riješenja tih manjih problema se onda kombiniraju dok se ne dobije riješenje početnog problema. To je tipičan način riješavanja problema po principu *divide-and-conquer* (podijeli-pa-vladaj). Dinamičko programiranje opisano je detaljno u mnogim knjigama koje se bave algoritmima, kao što je (Cormen, Leiserson, Rivest, & Stein, 2009). U ovom dijelu prikazaj je jedan algoritam za pronalaženje edit-distance zasnovan na principu dinamičkog programiranja.

Pretpostavimo da želimo odredit edit-distancu *ed(x, y)*. Nadalje, neka postoji matrica $C_{0..|x|, 0..|y|}$ (|*x*| je broj koji označava duljinu stringa *x*) koja je popunjena s vrijednostima gdje polje $C_{i, j}$ sadrži najmanji broj operacija potrebnih da se poravna $x_{1..i}$ sa $y_{1..j}$. Tada se edit-distanca može izračunati prema sljedećem:

$C_{i, 0} = i$

$C_{0, j} = j$

$C_{i,j} = ako\ (x_i = y_j)\ onda\ C_{i-1, j-1},\ inače\ 1 + min(C_{i-1, j}, C_{i, j-1}, C_{i-1, j-1})$

Na kraju ovog postupka $C_{|x|, |y|} = ed(x, y)$.

$C_{i, 0}$ i $C_{0, j}$ predstavljaju edit-distancu između stringa duljine *i* ili *j* i praznog stringa. Ovdje je očito da je potrebno *i* i *j* brisanja da bi se sa nepraznog došlo do praznog stringa. Za dva neprazna stringa duljine *i* i *j* može se pretpostaviti da su sve edit-distance kraćih stringova bile izračunate, pa se sada konvertira $x_{1..i}$ u $y_{1..j}$.

Ako su posljednji znakovi x_i i y_j jednaki tada ih nema potrebe uzeti u obzir, pa se ide na konverziju $x_{1..i-1}$ i $y_{1..j-1}$. Međutim, ako nisu jednaki onda možemo obrisati x_i i konvertirati $x_{1..i-1}$ u $y_{1..j}$, umetnuti y_j na kraj $x_{1..i}$ i konvertirati $x_{1..i}$ u $y_{1..j-1}$ ili zamijeniti x_i sa y_j i konvertirati $x_{1..i-1}$ i $y_{1..j-1}$.

Ovakav algoritam mora popunjavati matricu na način da u gornje, lijevo i gornje-lijevo susjedi nekog polja izračunati prije samog tog polja. Na slici 75 (stranica 99) ilustriran je ovaj algoritam na računanju *ed("survey", "surgery")*.

Ovakav se algoritam može lako prilagoditi i za pretraživanje teksta gdje se traži neki kratki uzorak unutar većeg teksta.

5.4.1.2 Algoritmi s konačnim automatima

Još jedan način za pronalaženje edit-distance je upotrebom nedeterminističkog konačnog automata (*Nondeterministic Finite Automaton, NFA*). Teorija automata i formalnih jezika opisana je u (Hopcroft & Ullman, 1979).

Na slici 76 (stranica 99) prikazan je NFA za $k = 2$ greške (odstupanja). Svaki redak označava broj grešaka ustanovljenih do određenog trenutka, a svaki stupac predstavlja prefik uzorka koji se podudara s tekstom. Horizontalne strelice predstavljaju podudaranje znakova – ako se znakovi podudaraju onda se ide na sljedeći znak uzorka i teksta. Svaki put kada dođe do

nepodudaranja ide se na sljedeći redak, to jest tada je došlo do greške. U tom slučaju vertikalne strelice označavaju umetanje znaka u uzorak, u kojem slučaju idemo na sljedeći znak u tekstu, ali ne i u uzorku. Pune dijagonalne strelice označavaju supstituciju znaka, gdje se ide na sljedeći znak i u tekstu i u uzorku. Isprekidane dijagonalne strelice označavaju brisanje znaka iz uzorka, što znači da se ide na sljedeći znak u uzorku, ali ne i u tekstu. Ovakav automat završava postupak podudaranja kada se nađe u jednom od krajnje desnih stanja (označenih dvostrukim krugom).

Iako je za ovo istraživanje inicijalno napravljen pokušaj upotrebe edit-distance s algoritmom koji je dio jedne od biblioteka za Python, od njene se upotrebe u osnovnom obliku odustalo iz sljedećih razloga:

- Edit-distanca po svojoj definiciji ne uključuje mogućnost ograničenja razmaka između slova, što bi se svodilo na ograničavanje broja umetanja ili brisanja na određenoj poziciji u tekstu ili uzorku. Ako je, na primjer, tekst *soba* poravnat s glasovima tako da između slova *s* i *o* postoji prevelik razmak (čije je trajanje duže od jednog do dva prosječna glasa) onda takvo poravnanje ne bi bilo realno jer u povezanom govoru nije vjerojatno da će postojati razmaci takvog trajanja.
- Poravnavanjem teksta s glasovima upotrebom edit-distance teško je dobiti informaciju gdje je početak, a gdje završetak tog poravnanja. U primjeru s tekstom *soba* i nizom glasova *msaovaaa*, neka je tekst poravnat na sljedeći način:

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*(pozicija glasa)*
m	*s*	*a*	*o*	*v*	*a*	*a*	*a*	
	s		***o***	***b***	***a***			

  Ovdje je važna informacija ta da se ovaj tekst poklapa s odgovarajućim nizom glasova na intervalu pozicija [1, 5] jer se pomoću te informacije može utvrditi u kojem intervalu zvuka je izgovorena ta riječ.

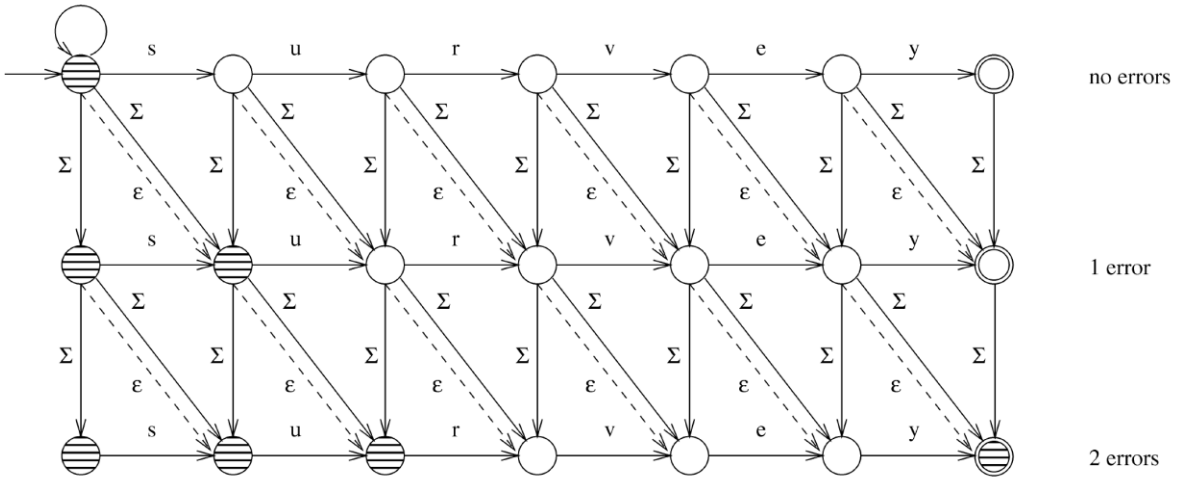
- S obzirom da neuralna mreža neke vrste glasova prepoznaje bolje od drugih potrebno je veću vrijednost dati poravnanjima u kojem su ti glasovi poravnati s odgovarajućim slovima teksta nego onima u kojima to nije slučaj. Primjerice, slovo *s* poravnato s

glasom *c* vrijedi više nego gdje je *b* poravnato s *m* jer neuralna mreža bolje prepoznaje frikative od nazala i okluziva.

Zbog toga je za ovo istraživanje napravljen algoritam pravnavanja koji uzima u obzir točnost poravnanja, kao i edit-distanca, ali se također može parametrizirati tako da se izbjegnu gore navedeni problemi.

		s	u	r	g	e	r	y
	0	1	2	3	4	5	6	7
s	1	**0**	1	2	3	4	5	6
u	2	1	**0**	1	2	3	4	5
r	3	2	1	**0**	1	2	3	4
v	4	3	2	1	**1**	2	3	4
e	5	4	3	2	2	**1**	**2**	3
y	6	5	4	3	3	2	2	**2**

Slika 75: Primjer matrice za računanje edit-distance između riječi „survey“ i „surgery“ algoritmom dinamičkog programiranja (Navarro, 2001).



Slika 76: NFA za približno podudaranje uzorka „survey“ s dvije greške. Zatamnjena stanja su ona koja su aktivna nakon prolaska kroz tekst „surgery“ (Navarro, 2001).

5.4.2 Algoritam

U ovom dijelu opisan je algoritam upotrebljen za ovo istraživanje. On ne izračunava edit-distancu nego traži podudaranje koje donosi maksimalan broj bodova, uzimajući u obzir kategorije glasova koje neuralna mreža prepoznaje najbolje. Takvim grupama glasova dodijeljeni su bodovi gdje više bodova dobije grupa glasova koju mreža bolje prepoznaje, a manje ona grupa kod koje je obično više grešaka u prepoznavanju. U sljedećem primjeru ilustriran je rad ovog algoritma. Neka je definirana sljedeća grupa glasova i tekst koji treba poravnati:

Tekst: **soba**
Grupe glasova:
0. [zsz]
1. [mn]
2. [ao]
3. [rrr]
4. [eaaa]
5. [rvrv]
6. [eae]

Algoritam prvo razvije stablo kombinacija tako da je širina stabla određena brojem grupa, a dubina brojem slova teksta. Na slici 79 (stranica 107) prikazan je dio stabla kombinacija za pronalaženje podudaranja gornjeg teksta sa zadanim grupama glasova. Stablo je prikazano vertikalno tako da je širina na vertikali, a dubina na horizontali. Dubina stabla je 4 jer su četiri slova u riječi *soba*, a širina 7 jer je obuhvaćeno sedam grupa glasova (indeksi počinju od 0). Svaki čvor stabla predstavlja grupu glasova i označen je indeksom koji označava poziciju slova u tekstu s kojim je ta grupa poravnata (ali se ne mora podudarati s tim slovom). Na toj slici potcrtani su čvorovi koji su dio *najboljeg puta* u tom stablu. U ovom primjeru niz indeksa najboljeg puta je <0, 2, 3, 4>, a prema gornjem popisu grupe s tim indeksima su *zsz*, *ao*, *rrr* i *eaaa*. To znači da grupu s indeksom 0 (*zsz*) treba poravnati sa slovom na indeksu 0 (s), grupu s indeksom 2 (*ao*) sa slovom na indeksu 1 (o), grupu s indeksom 3 (*rrr*) sa slovom na indeksu 2 (b) i grupu s indeksom 4 (*eaaa*) sa slovom na indeksu 3 (a). Poravnanje je, prema tome

0	*1*	*2*	*3*	*4*	*5*	*6*	*(indeks grupe)*
zsz	*mn*	*ao*	*rrr*	*eaaa*	*rvrv*	*eae*	
s		***o***	***b***	***a***			
0		*1*	*2*	*3*			*(indeks slova)*

Kao što je prethodno rečeno, ovom se algoritmu može postaviti parametar koji ne dozvoljava razmake veće od neke zadane vrijednosti. U ovom primjeru, između slova *s* i *o* postoji razmak od jedne grupe. To može biti prihvatljivo jer ta jedna grupa vremenski ne zauzima više od jednog glasa. Međutim, ako bi se slovo *a* poravnalo s grupom na indeksu 6 (*eae*) onda bi razmak između slova *b* i *a* bio dvije grupe što je manje realno jer bi to značilo da se na tom mjestu vremenski možda nalaze dva ili više glasova koji su ovim postupkom preskočeni, odnosno ignorirani. Zbog toga je ovaj postupak praktičniji od jednostavnog izračunavanja edit-distance, iako je moguće da bi se i neki algoritam za edit-distancu mogao prilagoditi za ovu svrhu tako da se ograniči broj umetanja praznih mjesta u tekst ili uzorak, iako takvo istraživanje ovdje nije napravljeno.

Na slici 78 (stranica 106) u cijelosti je ilustriran rad ovog algoritma kroz isti primjer. Glasovi u tom primjeru su bodovani prema sljedećoj tablici:

Kategorija glasa	**Bodovi**
i, e, a, o, u	0.3
č, š	3.5
c, s	3.5
đ, ž, z	3.5

S lijeve strane stabla nalazi se broj koji pokazuje dubinu svakog čvora, a ta dubina označava indeks slova teksta koje uspoređujemo s grupom na indeksu pokazanom u samom čvoru.

Svako slovo koje se nalazi u grupi glasova s kojom ga se uspoređuje inicijalno dobije 0.5 bodova. Ako se glas nalazi u gornjoj tablici onda mu se pribroji i broj bodova koji je za tu vrstu glasa dodijeljen. Na slici 78 s desne strane nalazi se podstablo cijelog stabla s lijeve strane. To je podstablo poravnanje koje je algoritam pronašao kao najbolje. Svaki čvor u zagradi sadrži dva broja:

- Broj bodova za poklapanje trenutnog slova i odgovarajuće grupe glasova.

- Ukupan broj bodova za odgovarajuće podstablo.

Ovdje se vidi da korijen ima ukupno 5.6 bodova za zadani tekst i grupu glasova, a na kraju ispisa s desne strane nalazi se popis parova, gdje prva vrijednost svakog para označava indeks slova koje je poravnato s grupom glasova sadržanoj u drugoj vrijednosti para. Iz toga se vidi da je najbolje poklapanje

[(0, ['z', 's', 'z']), (2, ['a', 'o']), (3, ['r', 'r', 'r']), (4, ['e', 'a', 'a', 'a'])]

što odgovara poklapanju kako je prethodno pokazano. U podstablu s desne strane na slici 78 ova putanja je podebljana i potcrtana. Čvor s indeksom grupe 4 ima 0.8 bodova jer se na tom indeksu nalazi grupa glasova *eaaa* koja je poravnata sa slovom *a* (dubina tog čvora je 3, što je indeks slova *a* u tekstu). Ti su bodovi dobijeni tako da se zbroji inicijalnih 0.5 bodova (što se daje svakom slovu koje se nalazi u odgovarajućoj grupi glasova) s 0.3 boda, što je broj bodova za vokale prema gornjoj tablici. Sada čvor iznad, s indeksom grupe 3, pokazuje 0/0.8. Prvi broj pokazuje da na tom mjestu poklapanje nosi 0 bodova, što je logično jer se slovo *b* ne poklapa s glasom *r*, odnosno ne nalazi se u grupi glasova *rrr*. Drugi broj pokazuje koliko ovo podstablo ukupno nosi bodova. S obzirom da se samo slovo *a* poklapa sa svojom grupom ukupna je vrijednost tog podstabla 0.8 bodova. Sada se opet ide na sljedeći čvor, onaj čiji je indeks grupe 2. Ovdje se slovo *o* poklapa s grupom na tom indeksu (odnosno nalazi se u toj grupi glasova), pa je broj bodova za to poklapanje 0.8 (prvi broj), a ukupna vrijednost tog dijela podstabla je 1.6, što je zbroj 0.8 bodova za podstablo od indeksa 3 i 0.8 bodova za trenutni čvor s indeksom grupe 2. Sada se dolazi do čvora s indeksom grupe 0. Slovo *s* nalazi se u grupi glasova s indeksom 0, a to prema gornjoj tablici nosi 3.5 bodova. Ako se tome pribroji inicijalnih 0.5 bodova dobije se 4 boda za podudaranje slova *s* s grupm *zsz*. Tih 4 boda sada se zbroje s ukupnim brojem bodova podstabla grupe s indeksom 2, što je 1.6, pa se dobije konačni broj bodova 5.6 za ovo poravnanje.

Na slici 77 (stranica 105) neki su brojevi (na početku reda) koji označvaju dubinu čvora u tom redu označeni s M. Time su označeni čvorovi koje je algoritam već analizirao i koje ne treba opet analizirati. Primjerice, u sljedećem dijelu stabla označeno podstablo se pojavljuje dva puta: Prvi put u prvom zatamnjenom dijelu i drugi put u drugom. Oznake *#n* su identifikator čvora, gdje se vidi da su čvorovi u drugom zatamnjenom dijelu u stvari isti čvorovi iz prvog dijela. Ovdje je važno to da se u oba dijela radi o podstablu *na istoj dubini*, što pokazuju i brojevi dubine sa strane (2, 3, 3). S obzirom da dubina označava indeks slova u tekstu ovdje se radi u analizi poravnanja slova na indeksima 2 i 3 s grupama na indeksima 3 i 4, te slova na

indeksima 2 i 3 s grupama na indeksima 3 i 5. Nakon što je analiza tih poravnanja napravljena jednom, nema je potrebe ponavljati (jer se radi o istim slovima i grupama), pa se na ovaj način uštedi značajno vrijeme izvršavanja programa kojim je ovaj algoritam implementiran. Ova se tehnika zove *memoizacija* (od riječi *memo*) jer se dobiveni rezultati spremaju da bi se kasnije izbjeglo ponovno izračunavanje istih u kojem slučaju se te rezultate samo pročita iz tablice u kojoj su spremljeni. Na toj se slici može vidjeti da je velik dio cijelog stabla u stvari memoiziran na ovaj način, što ukazuje na efikasnost ovog algoritma. Bez ove tehnike dobili bi velik broj kombinacija s kojima iznova analiziramo dio podudaranja, kao što je ilustrirano u tablici 7.

Jedan problem kod ovog algoritma je u tome što u slučajevima gdje je glasova manje nego teksta neće napraviti dobro poravnanje. U sljedećem primjeru dan je popis grupa glasova gdje je izgovoreno „istodobno su nakon ...“ i slova riječi „istodobno“ poravnata s tim grupama:

```
['v', 'm', 'n']
['i'].....................    i
['s'].....................    s
['a', 'o'].................    t
['n', 'd', 'h']
['a', 'o'].................    o
['m', 'n'].................    d
['a', 'o'].................    o
['s'].....................    b
['o', 'u'].................    n
['o'].....................    o
['a', 'o']
['r']
['a']
['a', 'o']
['a', 'o']
['m', 'l']
```

Zatamnjeni dio pokazuje niz grupa glasova gdje se lako vidi da je na tom mjestu izrečeno „istodobno“, ali nedostaju glasovi *t* i *b*. S obzirom da algoritam pokušava poravnati svako slovo teksta s nekim glasom (to jest, grupom) u ovom je slučaju prešao na iduću riječ. U ovom primjeru ispravno poravnanje bilo bi ovako:

```
['v', 'm', 'n']
['i'].....................    i
['s'].....................    s
['a', 'o'].................    o
['n', 'd', 'h'].....      d
['a', 'o'].................    o
['m', 'n'].................    n
['a', 'o'].................    o
['s']
['o', 'u']
['o']
['a', 'o']
```

```
['r']
['a']
['a', 'o']
['a', 'o']
['m', 'l']
```

To, međutim, znači da bi algoritam trebao isprobati i poravnanja u kojima su neka slova isključena. Drugim riječima, algoritam bi tada trebao isprobati sve kombinacije teksta bez jednog slova, pa one bez dva slova, bez tri itd. To bi znatno povećalo broj kombinacija (jer postoji puno načina da se iz nekog teksta ukloni *n* slova), a time i vrijeme izvršavanja. Iako ta modifikacija algoritma nije napravljena za ovo istraživanje, ona je ostavljena kao jedna od tema budućeg rada na ovom problemu.

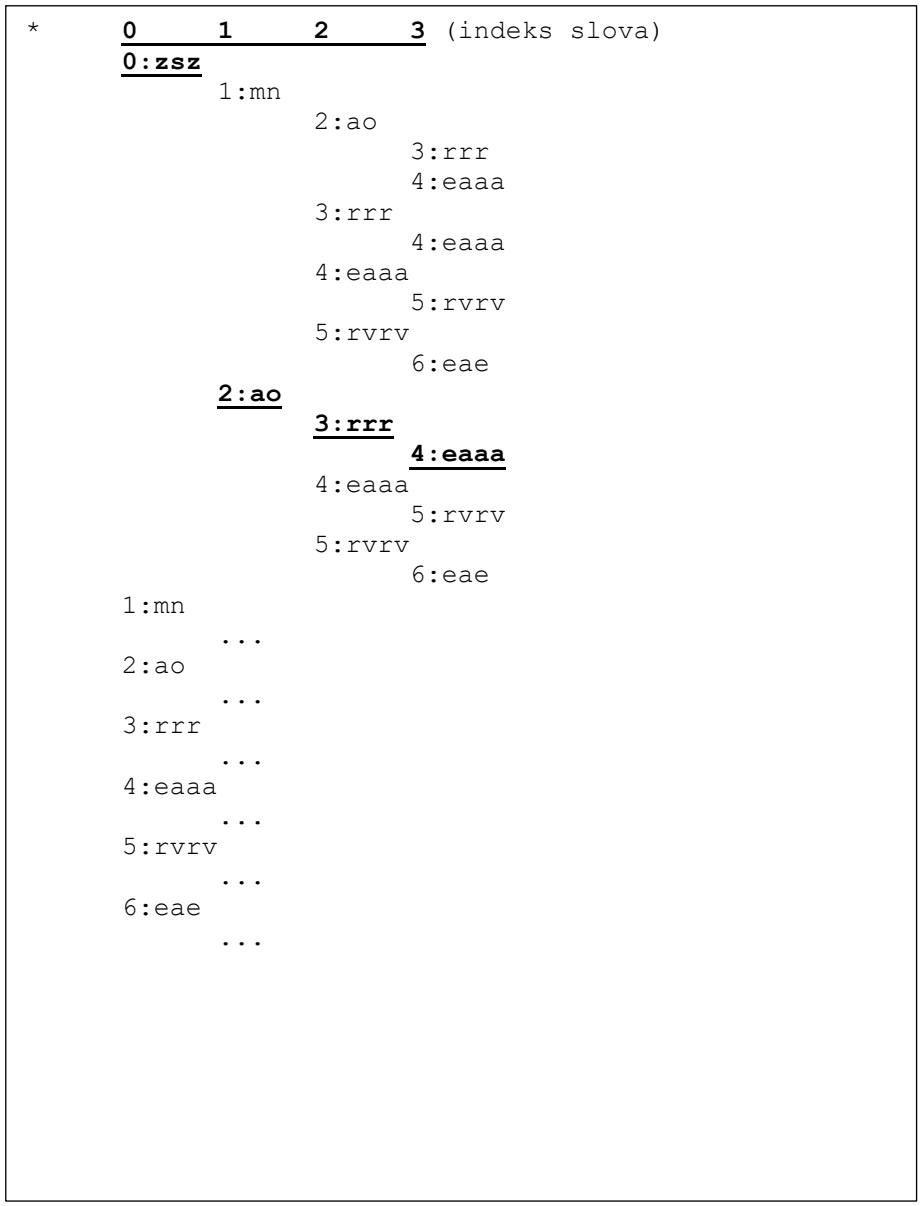
```
-1:  -1 (0/0) [] #0
0:  0 (4.0/4.0) [] #1
1:    1 (0/0) [] #2
2:      2 (0/0) [] #3
3:        3 (0/0) [] #4
3:        4 (0.8/0.8) [] #5
2:      3 (0/0) [] #6
3:        4 (0.8/0.8) [] #7
3:        5 (0/0) [] #8
1:    2 (0.8/0.8) [] #9
2M      3 (0/0.8) [7, <Cvor>] #6
3M        4 (0.8/0.8) [4] #7
3M        5 (0/0) [5] #8
2:      4 (0/0) [] #11
3:        5 (0/0) [] #12
3:        6 (0.8/0.8) [] #13
0:  1 (0/0) [] #14
1M    2 (0.8/1.6) [6, <Cvor>] #9
2M      3 (0/0.8) [7, <Cvor>] #6
3M        4 (0.8/0.8) [4] #7
3M        5 (0/0) [5] #8
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
1:    3 (0/0) [] #16
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
2:      5 (0/0) [] #18
3:        6 (0.8/0.8) [] #19
0:  2 (0/0) [] #20
1M    3 (0/0.8) [11, <Cvor>] #16
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
2M      5 (0/0.8) [19, <Cvor>] #18
3M        6 (0.8/0.8) [6] #19
1:    4 (0.3/0.3) [] #22
2M      5 (0/0.8) [19, <Cvor>] #18
3M        6 (0.8/0.8) [6] #19
2:      6 (0/0) [] #24
0:  3 (0/0) [] #25
...
```

Slika 77: Dio stabla pretraživanja (zatamnjeni dio se pojavljuje dva puta).

```
-1:  -1 (0/0) [] #0
0:  0 (4.0/4.0) [] #1
1:    1 (0/0) [] #2
2:      2 (0/0) [] #3
3:        3 (0/0) [] #4
3:        4 (0.8/0.8) [] #5
2:      3 (0/0) [] #6
3:        4 (0.8/0.8) [] #7
3:        5 (0/0) [] #8
1:    2 (0.8/0.8) [] #9
2M      3 (0/0.8) [7, <Cvor>] #6
3M        4 (0.8/0.8) [4] #7
3M        5 (0/0) [5] #8
2:      4 (0/0) [] #11
3:        5 (0/0) [] #12
3:        6 (0.8/0.8) [] #13
0:  1 (0/0) [] #14
1M    2 (0.8/1.6) [6, <Cvor>] #9
2M      3 (0/0.8) [7, <Cvor>] #6
3M        4 (0.8/0.8) [4] #7
3M        5 (0/0) [5] #8
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
1:    3 (0/0) [] #16
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
2:      5 (0/0) [] #18
3:        6 (0.8/0.8) [] #19
0:  2 (0/0) [] #20
1M    3 (0/0.8) [11, <Cvor>] #16
2M      4 (0/0.8) [13, <Cvor>] #11
3M        5 (0/0) [5] #12
3M        6 (0.8/0.8) [6] #13
2M      5 (0/0.8) [19, <Cvor>] #18
3M        6 (0.8/0.8) [6] #19
1:    4 (0.3/0.3) [] #22
2M      5 (0/0.8) [19, <Cvor>] #18
3M        6 (0.8/0.8) [6] #19
2:      6 (0/0) [] #24
0:  3 (0/0) [] #25
1M    4 (0.3/1.1) [18, <Cvor>] #22
2M      5 (0/0.8) [19, <Cvor>] #18
3M        6 (0.8/0.8) [6] #19
2M      6 (0/0) [] #24
1:    5 (0/0) [] #27
2M      6 (0/0) [] #24
0:  4 (0/0) [] #29
1M    5 (0/0) [] #27
```

```
0>  0 (4.0/5.6) [9, <Cvor>] #1
1>    1 (0/0.8) [3, <Cvor>] #2
2>      2 (0/0.8) [5, <Cvor>] #3
3>        3 (0/0) [3] #4
3>        4 (0.8/0.8) [4] #5
2>      3 (0/0.8) [7, <Cvor>] #6
3>        4 (0.8/0.8) [4] #7
3>        5 (0/0) [5] #8
1>    2 (0.8/1.6) [6, <Cvor>] #9
2>      3 (0/0.8) [7, <Cvor>] #6
3>        4 (0.8/0.8) [4] #7
3>        5 (0/0) [5] #8
2>      4 (0/0.8) [13, <Cvor>] #11
3>        5 (0/0) [5] #12
3>        6 (0.8/0.8) [6] #13
[(0, ['z', 's', 'z']), (2, ['a',
'o']), (3, ['r', 'r', 'r']), (4,
['e', 'a', 'a', 'a'])]
BODOVI: 5.6
```

Slika 78: Primjer postupka poravnavanja. Lijevi okvir sadrži cijelo stablo pretraživanja, a desni podstablo s najboljim rezultatom.



Slika 79: Primjer generiranja kombinacija za pronalaženje odgovarajućeg podudaranja.

Tablica 7: Primjer poravnavanja generiranjem svih mogućih kombinacija (podebljan je redak s najboljim poravnanjem).

Kombinacija	**[zsz]**	**[mn]**	**[ao]**	**[rrr]**	**[eaaa]**	**[rvrv]**	**[eae]**
1	s	o	b	a			
2	s	o	b		a		
3	s	o	b			a	
4	s	o	b				a
5	s	o		b	a		
6	s	o		b		a	
7	s	o		b			a
8	s	o			b	a	
9	s	o			b		a
10	s	o				b	a
11	**s**		**o**	**b**	**a**		
12	s		o	b		a	
13	s		o	b			a
14	s		o		b	a	
15	s		o		b		a
16	s		o			b	a
17	s			o	b	a	
18	s			o	b		a
19	s			o		b	a
20	s				o	b	a
21		s	o	b	a		
22		s	o	b		a	
23		s	o	b			a
24		s	o		b	a	
25		s	o		b		a
26		s	o			b	a
27		s		o	b	a	
28		s		o	b		a
29		s		o		b	a
30		s			o	b	a
31			s	o	b	a	
32			s	o	b		a
33			s	o		b	a
34			s		o	b	a
35				s	o	b	a

5.4.3 Poravnavanje cijelih podnatpisa s govorom

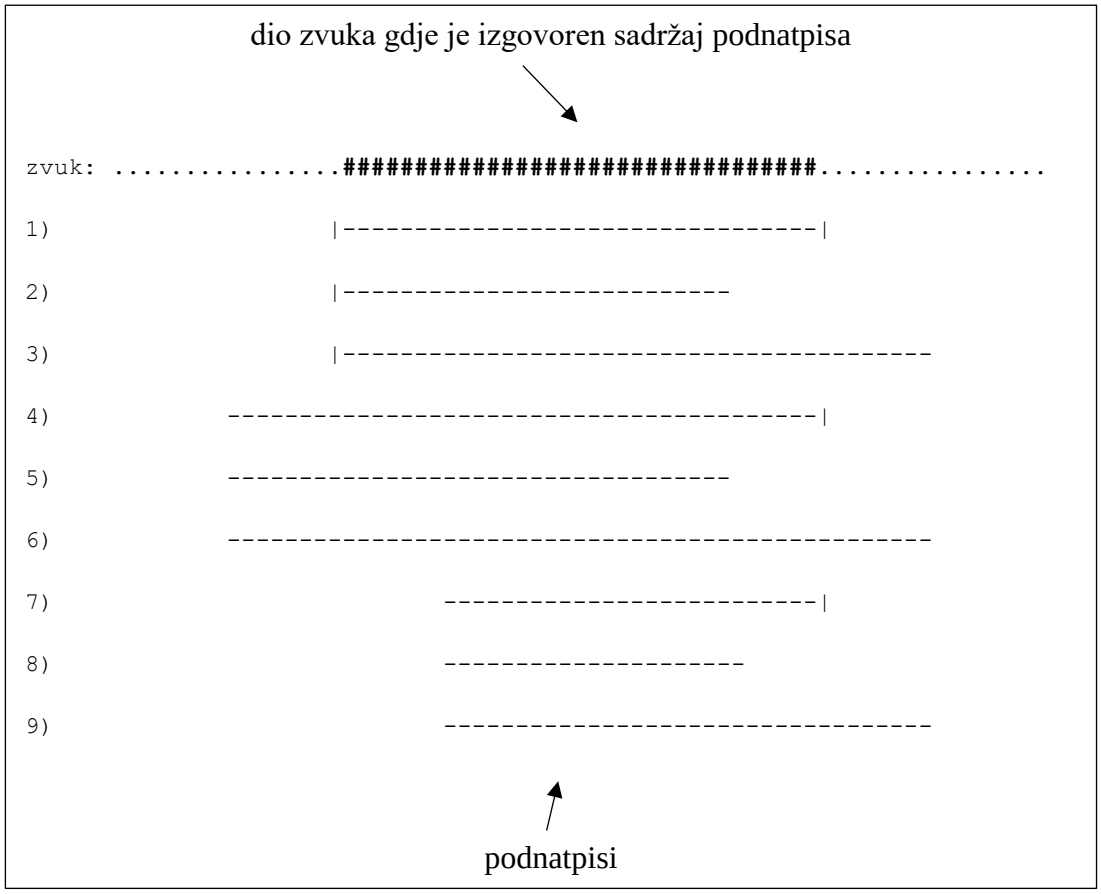
U prethodnom dijelu opisan je algoritam koji se u ovom radu upotrebljava za poravnavanje pojedinačnih riječi s tekstom koji se može nalaziti u podnatpisu ili zadati zasebno. U ovom dijelu opisan je postupak poravnavanja cijelog podnatpisa s govorom onoga što se u njemu nalazi.

Podnatpisi se pojavljuju u okolini vremenskog intervala unutar kojeg je izgovoren tekst onoga što se nalazi u podnatpisu. Idealna pozicija podnatpisa je ona gdje se on pojavljuje otprilike (jer neće biti savršeno poravnat s govorom) na početku govornog dijela, a nestaje na kraju. Međutim, odstupanja su puno češća. Općenito, pojavljivanje i nestajanje podnatpisaa može se podijeliti u devet slučajeva:

1. Podnatpis se pojavljuje i nestaje točno ili dovoljno blizu početka odnosno završetka govornog dijela.
2. Podnatpis se pojavljuje na početku govornog dijela, a nestaje prije završetka.
3. Podnatpis se pojavljuje na početku govornog dijela, a nestaje nakon završetka.
4. Podnatpis se pojavljuje prije početka govornog dijela i nestaje točno na završetku.
5. Podnatpis se pojavljuje prije početka govornog dijela i nestaje prije završetka.
6. Podnatpis se pojavljuje prije početka govornog dijela, a nestaje nakon završetka
7. Podnatpis se pojavljuje nakon početka govornog dijela, a nestaje točno na završetku.
8. Podnatpis se pojavljuje nakon početka govornog dijela, a nestaje prije završetka.
9. Podnatpis se pojavljuje nakon početka govornog dijela, a nestaje nakon završetka.

Ovi su slučajevi ilustrirani na slici 80. Dio označen s „#“ je govor čiji se tekst nalazi u podnatpisu, a dio označen s „-„ je podnatpis. Vertikalne crte („|“) na nekim počecima i krajevima podnatpisa označavaju da se podnatpis pojavljuje ili nestaje tamo gdje govor počinje odnosno završava. Odstupanja od točne pozicije (početak ili završetak segmenta govora) su uglavnom od 0.5 do 1 sekunde, pa je za potrebe ovog istraživanja upotrebljeno odstupanje od 1 sekunde.

Za poravnavanje cijelog podnatpisa s govorom u ovom se radu upotrebljava isti algoritam kao i za poravnavanje pojedinačnih riječi u podnatpisu. I ovdje je broj bodova važan za odabir najboljeg poravnanja, iako više poravnanja može rezultirati istim brojem bodova. U poglavlju o rezultatima prikazani su detalji.



Slika 80: Devet slučajeva pojavljivanja podnatpisa.

5.5 Detekcija naglašenih riječi

Nakon što je tekst poravnat s govorom na snimci detekcija naglašenih riječi može se podijeliti u tri koraka:

1. Analiza intenziteta
2. Analiza tona (intonacije)
3. Analiza trajanja vokala (tempo govora)

Ovi se koraci ne moraju obavezno izvršavati tim redoslijedom. U ovom istraživanju rezultat ovih koraka sagledava se zasebno tako da se na osnovu takve skupine informacija mogu dobiti naznake je li neka riječ naglašena. Na primjer, ako se unutar jednog segmenta govora ističe samo povišeni ton onda je moguće da je ta riječ bila izgovorena naglašeno, bez obzira na to što na tom mjestu nije istovremeno bio pojačan intenzitet. Osnovna poteškoća u automatskoj detekciji naglašenih riječi je ta da ne postoji nekakva egzaktna mjera kojom se može utvrditi je li neka riječ naglašena ili ne. Čak i u slušanju snimke nečijeg govora dvoje ljudi se neće uvijek složiti oko toga je li govornik neku riječ naglasio ili nije. U mnogim slučajevima naglašavanje riječi je jedva primjetno, nešto što slušatelj može primijetiti, ali nije jasno uočljivo u analizi zvuka. U nastavku je detaljno opisan način analize ovih triju prozodijskih obilježja.

5.5.1 Analiza intenziteta

Na slici 81 (stranica 114) u donjem dijelu prikazana je varijacija intenziteta u normalnom govoru jednog dijela snimke. Na prikazanoj snimci govor je uglavnom ujednačen i nema riječi koje su jasno istaknute, to jest naglašene. Na toj se slici vidi da niti jedan od vrhova krivulje intenziteta ne odskače znatno od ostalih. Općenito, razlike u intenzitetu između naglašenih i nenaglašenih riječi su male. Na slici 82 (stranica 114) prikazana je snimka govora gdje su unutar označenog dijela dvije riječi izgovorene naglašeno. Na tom se dijelu vidi da su razlike skokova u intenzitetu u odnosu na ostale dijelove snimke praktički neprimjetne.

Za analizu intenziteta upotrebljavamo njegov numerički prikaz generiran iz Praat softvera. Popis vrijednosti sastoji se od vremena i intenziteta (u decibelima) u formatu *vrijeme:intenzitet*:

```
...
0.20:75.69069933932057
0.21:74.0922823308485
0.22:71.29459609881668
0.23:79.64224075592689
0.24:64.50737756591333
0.25:64.5570390381121
...
```

Za ovo istraživanje upotrebljavamo dva ovakva niza vrijednosti: Jedan dobiven izdvajanjem vrijednosti intenziteta u segmentima od 10 ms, a drugi na mjestima gdje su prisutni glotalni pulsevi. Ovako dobiveni nizovi su kasnije spojeni u jedan na osnovu vremenskih pozicija vrijednosti intenziteta. Nakon što je dobiven jedan cijelokupni niz vrijednosti intenziteta potrebno je u nekom zadanom segmentu tog niza naći skokove u intenzitetu koji prelaze neki zadani prag od *P*% u odnosu na prethodnu i sljedeću vrijednost. Takvi skokovi u intenzitetu predstavljaju jedan pokazatelj mogućeg naglašavanja riječi. Za ovo istraživanje *P* je inicijalno bio postavljen na 5, ali su isprobane i druge vrijednosti. U gornjem popisu vrijednosti intenziteta, na primjer, bio bi istaknut intenzitet 79.64 (zaokruženo na dvije decimale) jer je on za više od 5% veći od prethodne vrijednosti, 71.29 i sljedeće vrijednosti, 64.5.

Nakon što su metodom opisanom u 5.4 aproksimirane granice riječi, na ovaj način se za svaku riječ može dobiti informacija o tome da li je u intervalu zvuka te riječi bilo znatnih skokova u intenzitetu. U primjeru gornjeg popisa vrijednosti intenziteta, ako je neka riječ bila izgovorena u intervalu 0.20 – 0.25 onda bi za tu riječ bio označen skok u intenzitetu (na vremenu 0.23) kao jedan od pokazatelja naglašenosti.

5.5.2 Analiza tona

Slično analizi intenziteta, analiza tona u ovom istraživanju podrazumijeva pronalaženje skokova u promjenama tona tijekom govora. Na slici 83 (stranica 115) prikazan je isti govor kao na slici 82, ali s dodatnom krivuljom koja pokazuje varijacije u tonu. S obzirom da su na označenom mjestu dvije riječi izgovorene naglašeno jasno se primijećuju dva vrha tona na kojima je povišena frekvencija jer su vokali na tom mjestu izgovoreni povišenim tonom. Na slici 84 (stranica 115) prikazan je isti govor kao onaj na slici 81 koji je uglavnom jednoličan, s gotovo neprimjetnim promjenama u intonaciji, što se jasno vidi na krivulji tona na kojoj nema istaknutih promjena. Ovdje se jasno vidi razlika između govora na slici 84 i onoga na slici 81 koji je puno dinamičniji sa stajališta intonacije.

Kao za analizu intenziteta, i za analizu tona izdvojene su numeričke vrijednosti pomoću Praat softvera u formatu *vrijeme:ton*:

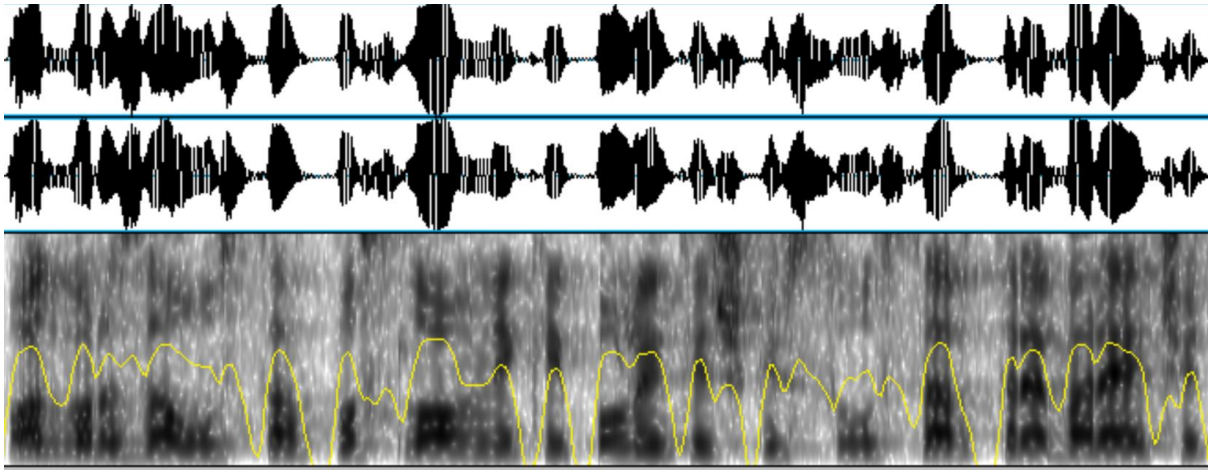
```
...
0.40000000000000024:115.12787313056477
0.41000000000000025:113.67333178308625
0.42000000000000026:112.25458867510922
0.43000000000000027:112.49184156258997
0.4400000000000003:113.28262816703239
0.4500000000000003:114.37689023786692
0.4600000000000003:115.86225095028728
0.4700000000000003:116.23608504823541
```

```
0.4800000000000003:116.65219187740094
0.4900000000000003:116.37855480472525
0.5000000000000003:116.2806751878383
...
```

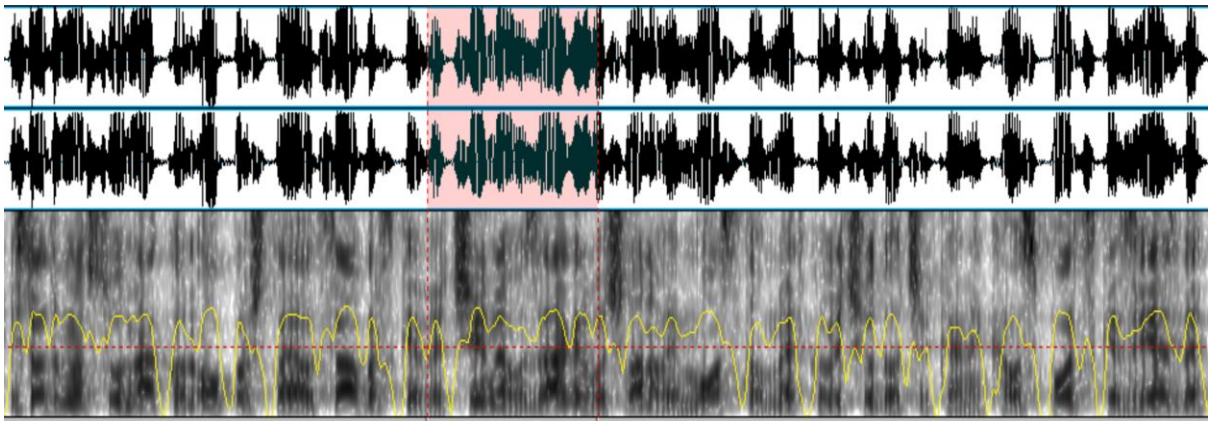
Sada se analiziraju vrijednosti tona na isti način kao i kod intenziteta, s tim da se u ovom slučaju uzimaju razlike od 20% u odnosu na okolne vrijednosti (zato jer su varijacije u tonu više izražene nego kod intenziteta).

5.5.3 Analiza trajanja vokala

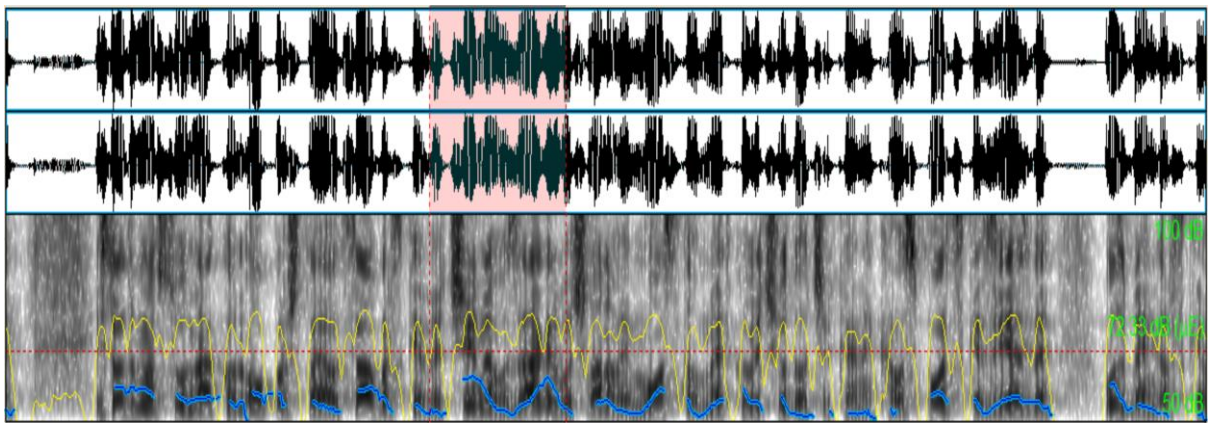
Varijacije u tempu govora također su dio prozodijskih obilježja (2.8). S tehničke strane, tempo govora određen je trajanjem vokala. Na slici 85 (stranica 116) prikazan je dio popisa klasificiranih glasova snimke govora. U redu 94 vidljiv je znatno duži niz vokala od ostalih, što ukazuje na to da je riječ na tom mjestu izgovorena sporije. Ovo samo po sebi ne mora značit da je riječ bila jasno naglašena, nego samo pruža jednu dodatnu informaciju u kombinaciji s informacijama o intenzitetu i tonu koja može poslužiti kao jedan od mogućih indikatora naglašenosti. Da bi se ustanovila produljenost vokala potrebno je uzeti u obzir njihovo trajanje u okolini, odnosno nekoliko riječi prije i nekoliko riječi poslije riječi koju analiziramo. Na slici 86 (stranica 117) prikazan je popis glasova za šest riječi. Da bi se ustanovilo da li unutar intervala neke riječi postoji znatno odstupanje u duljini vokala, odredi se segment od N riječi koji se zove *prozor* unutar kojeg uspoređujemo duljine vokala. To se u ovom istraživanju radi tako da se prvo odredi najdulji niz vokala riječi (ako ih ima) i taj niz usporedimo s najduljim nizom vokala riječi unutar trenutnog prozora. Ako je niz vokala riječi koju analiziramo dulji od P% od najduljeg niza vokala okolnih riječi onda se ta riječ označi kao produljena. Ovaj se postupak ponavlja kroz cijeli niz riječi zadanog teksta. U ovom radu veličina prozora postavljena je na šest riječi, a prag odnosa duljina nizova na 70%, što znači da se svako odstupanje veće od 70% u odnosu na ostale duljine nizova vokala unutar prozora smatra produljenim izgovorom.



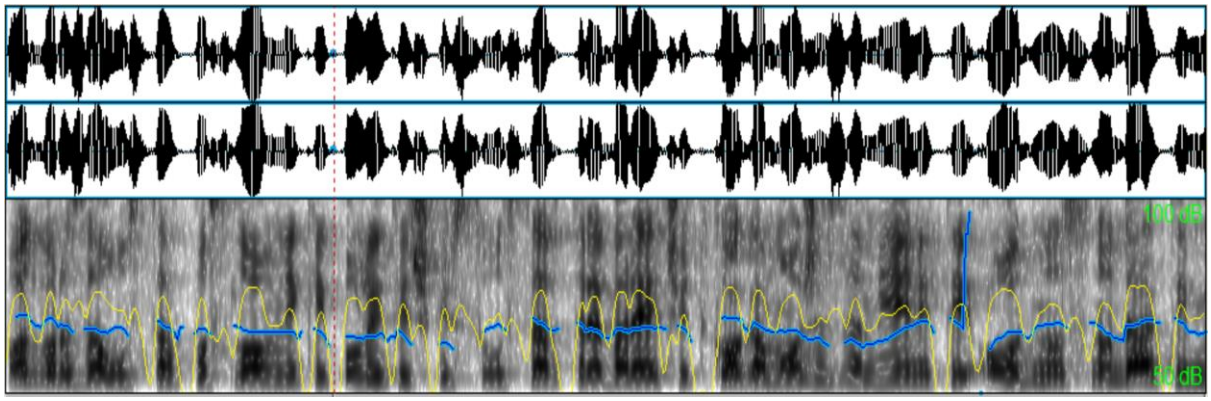
Slika 81: Prikaz varijacije intenziteta (donja slika, žuta linija).



Slika 82: Segment zvuka s označenim dijelom gdje su (lagano) naglašene dvije riječi. Varijacije u intenzitetu označene su žutom linijom.



Slika 83: Segment zvuka s označenim dijelom gdje su (lagano) naglašene dvije riječi. Varijacije u intenzitetu označene su žutom linijom, a varijacije u tonu plavom.



Slika 84: Prikaz varijacije tona (donja slika, plava crta).

R.BR.	OD	DO	DULJINA	GLASOVI
...				
57:	24.55	24.59	41.6	aaeeeeeie
58:	24.62	24.7	76.5	*mmvmnlmm
59:	24.7	24.72	19.0	oooo
60:	24.73	24.82	86.6	ooooaooooooooooaooaa
61:	24.83	24.88	50.0	ssssss
62:	24.97	25.06	90.0	aouooooooooooooou
63:	25.07	25.11	41.8	*nnnnm
64:	25.12	25.18	62.1	aaeaeaaaeeeeaaa
65:	25.28	25.34	56.8	aooaoaoaaao
66:	25.34	25.34	0.0	*n
67:	25.37	25.42	44.6	aaaeaeeeii
68:	25.42	25.44	20.0	zzzz
69:	25.45	25.51	60.0	čččščšč
70:	25.53	25.53	0.0	*n
71:	25.54	25.57	25.2	aaaaoa
72:	25.6	25.6	0.0	*n
73:	25.6	25.61	10.9	uuu
74:	25.62	25.67	45.4	oaaaaoooa
75:	25.69	25.7	12.1	aaa
76:	25.71	25.73	11.6	*rm
77:	25.74	25.79	50.0	šššščž
78:	25.84	25.84	0.0	*n
79:	25.84	25.86	20.0	iiii
80:	25.86	25.86	0.0	*h
81:	25.94	25.99	50.5	aaaaeaaaeee
82:	26.01	26.01	0.0	*v
83:	26.03	26.06	21.8	aaeee
84:	26.08	26.09	11.9	eiie
85:	26.1	26.11	11.8	*gr
86:	26.18	26.29	105.4	eaaaaaaaaoaoooooaaaa
87:	26.29	26.35	60.0	zzssssss
88:	26.45	26.55	100.0	oooaooooooooaaaaaaaaaa
89:	26.55	26.56	9.0	*lr
90:	26.59	26.63	40.0	zssss
91:	26.72	26.76	38.6	ieieeeeee
92:	26.78	26.78	0.0	*d
93:	26.78	26.83	50.0	zzzzzzzzzz
94:	26.83	27.0	170.8	eieieieeeeieieeeieieeeeeeaaeaeaaeaaaaa
95:	27.02	27.05	27.7	zzzzzzz
96:	27.06	27.08	26.5	*vrr
97:	27.09	27.15	54.4	aaaaeaeaeaaa
98:	27.16	27.16	0.0	*l
99:	27.16	27.17	10.0	eee
100:	27.18	27.19	11.1	*nn
101:	27.23	27.25	20.0	uea
102:	27.33	27.38	50.0	aeiaaa
103:	27.41	27.44	30.0	aaea
...				

Slika 85: Popis dijela prepoznatih glasova sa snimke govora.

```
28:     5.2         5.26        62.1        eeeeoeeaeee

29:     5.37        5.39        20.0        eie
30:     5.39        5.39        0.0         *n
31:     5.4         5.47        77.8        eeeeeeeeee
32:     5.49        5.49        0.0         *v
33:     5.51        5.56        50.0        sscscs

34:     5.58        5.58        0.0         *j
35:     5.58        5.62        37.0        eeeeeeeeee

36:     5.63        5.64        13.5        *nvv
37:     5.65        5.66        13.9        ououo
38:     5.67        5.7         36.2        *vnnnt
39:     5.77        5.82        54.4        aaaaaaaaa
40:     5.91        6.0         91.1        aaaaaaaaaaaaaaaaaaaeeeeeaaaaaaaaaaa
41:     6.01        6.01        0.0         *n
42:     6.02        6.02        6.6         ooo
43:     6.03        6.04        13.1        *vvv
44:     6.05        6.09        40.7        ioeeeei
45:     6.1         6.1         7.2         *vv
46:     6.11        6.2         88.5        zzzzssssszz
47:     6.2         6.21        5.7         *vv

48:     6.21        6.28        67.5        eeeeeeeeeee
49:     6.28        6.3         11.9        eaae
50:     6.31        6.31        0.0         *k
51:     6.31        6.35        43.8        aaaaaaaaaaa
52:     6.36        6.4         34.6        aaaaaaaa
53:     6.47        6.6         126.2       aaaaaaaaaoaaa
54:     6.6         6.6         0.0         *r

55:     6.61        6.71        100.5       eeeeeeeeeeeeaaaaa
56:     6.72        6.76        44.1        aaaaaaaaaaaaaaaa
57:     6.87        6.98        107.8       aaaaaaeeeeeee
58:     6.99        6.99        5.0         *nn
59:     7.0         7.01        10.0        uuou
60:     7.01        7.08        64.8        *mrnrn
61:     7.08        7.14        60.1        ooooaooaooaoaoaaaooa
```

Slika 86: Popis glasova za šest riječi (odvojenih razmacima) s prozorom koji obuhvaća tri riječi. Iscrtkani dio prikazuje pomicanje prozora za jednu riječ niže koji tada obuhvaća sljedeći segment od tri riječi. U ovoj ilustraciji mogao bi se isticati niz vokala u redu 40 (zavisno od postavljenog praga omjera duljina).

6. Rezultati istraživanja

U ovom dijelu prikazani su rezultati istraživanja podijeljeni u tri dijela:

1. Klasifikacija glasova
2. Poravnavanje teksta s govorom
3. Detekcija naglašenih riječi

Svaki od ovih dijelova ovisi o prethodnom – preciznost poravnavanja teksta s govorom ovisi o preciznosti klasifikacije glasova, a preciznost detekcije naglašenih riječi ovisi o preciznosti poravnanja teksta s govorom da bi se dobile granice riječi.

6.1 Klasifikacija glasova

U ovom su dijelu prikazani rezultati klasifikacije glasova. Neuralna mreža je trenirana s oko 150 sekundi govora, što je prilično mali skup za treniranje u odnosu na općenite sustave za automatsko prepoznavanje govora, gdje skup za treniranje sadrži minimalno nekoliko sati govora.

Neuralna mreža nije sve vrste glasova prepoznavala s podjednakom preciznošću. Ispod su opisani detalji prepoznavanja pojedinih grupa glasova.

6.1.1 Vokali

Vokali spadaju u jednu od grupa glasova koje je neuralna mreža najtočnije klasificirala. S obzirom da vokali traju duže od ostalih glasova i sastoje se od jednog, uglavnom periodičnog zvuka, njihova klasifikacija je uglavnom bila dovoljno točna za kasnije poravnjanje teksta s glasovima. U većini slučajeva klasifikacija vokala uključivala je više različitih vokala, što je vjerojatno zavisilo o glasu govornika, kvaliteti snimke, okruženju u kojem se govornik nalazio (studio ili negdje drugdje) i načinu izgovora. Na primjer, segment govora gdje je izgovoren vokal može sadržavati niz vokala

aaaaoooaaaauu

Iako je u ovom slučaju glas *a* dominantan (prema broju pojavljivanja), ne znači da je on stvarno bio izgovoren. Ovdje se kao rezultat klasifikacije jednostavno uzima ovakav čitav niz koji znači da je na tom mjestu bio izgovoren jedan od vokala *a, o* ili *u*.

6.1.2 Frikativi

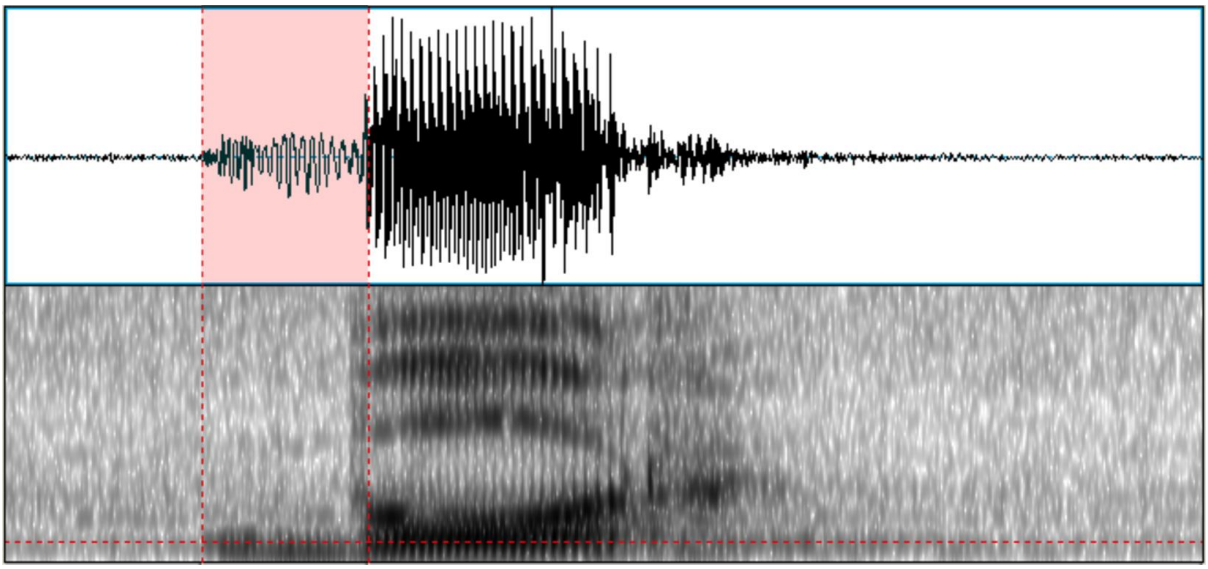
Frikativi *s, c, š, č/ć, z, ž* su također grupa glasova koji su bili prilično točno klasificirani. Iako kod njih nema periodičnosti (jer oni su šum), njihova je karakteristika ta da su visokofrekventni

i zbog toga lako prepoznatljivi. Bezvučni frikativi bili su bolje prepoznati nego zvučni, ali to nije predstavljalo problem jer su bezvučni frikativi brojniji i puno češći.

Frikativi *f* i *v* slabije su bili prepoznati, vjerojatno zbog slabijeg intenziteta i nižeg frekvencijskog raspona.

6.1.3 Okluzivi

Zvučni okluzivi, kao *b, d* i *g* su više problematični za segmentiranje jer se oni u stvari sastoje od dva dijela: okluzije i eksplozije. Na slici 87 prikazana je riječ *bor* gdje je označen glas *b*. Na toj se slici vidi eksplozija kojoj prethodi zvučna okluzija. Ovo predstavlja poteškoće kod klasifikacije ovakvih glasova jer segment zvuka od 10 milisekundi neće obuhvatiti cijeli glas nego samo jedan njegov dio (zvučnu okluziju ili eksploziju), pa će taj dio biti klasificiran kao zaseban glas. To je slučaj i kod mnogih drugih glasova, međutim ne i problem. Kao što je prethodno opisano, vokali mogu biti izgovoreni duže ili kraće, ali svaki je njihov segment uglavnom isti pa zbog toga nije problem ako je iz jednog vokala izdvojen segment od 10 milisekundi – taj segment i dalje ima karakteristike izgovorenog vokala.



Slika 87: Oscilogram i spektrogram riječi „bor“.

Bezvučni okluzivi su nešto manje problematični od zvučnih, ali zbog eksplozije moguće je da su u nekim slučajevima preskočeni ili samo djelomično obuhvaćeni, pa zbog toga nisu uvijek bili ispravno klasificirani.

6.1.4 Nazali

Nazali kao *m* i *n* sastoje se od tona na početku i tona na kraju, pa s toga kod njih postoji sličan problem kao kod zvučnih okluziva, iako u manjoj mjeri.

6.2 Rezultati prepoznavanja govora sa snimki

Prikazi klasificiranih glasova koji će ovdje biti prezentirani podijeljeni su u dvije skupine:

- Tri muška glasa – slike 88, 89 i 90
- Tri ženska glasa – slike 91, 92 i 93

Na slici 88 (stranica 123) prikazan je popis glasova sa snimke 1 u kojoj je izgovoreno „*a meni ostaje da pozovem još jednom sve da ovu akciju podrže i pozivima na broj ali i uplatama na taj račun*“. Potcrtani redovi označavaju gdje su otprilike granice riječi (utvrđene slušanjem). Govor na snimci nije snimljen u studiju i nije ga izgovorio profesionalni govornik, pa je zvuk i izgovor malo lošije kvalitete, iako to ne mora obavezno znatno utjecati na rezultate.

U prvom segmentu, 0..1, vidi se da je riječ „a“ točno prepoznata. Glas *a* se pojavljuje u dva reda zato jer je negdje između ta dva vremena neuralna mreža klasificirala manji dio zvuka kao neki drugi glas ili glasove. S obzirom da je to međuvrijeme prekratko da bi predstavljalo cijeli glas (oko 20 milisekundi) ti se glasovi ne pojavljuju u popisu.

U segmentu 2..8 izgovorena je riječ „meni“. U redu 2 nalazi se konsonant *m* zajedno s nekim drugim konsonantima. S obzirom da se jedan dio klasifikacije provodi nad segmentima zvuka od 10 milisekundi neuralna mreža svaki od tih segmenata klasificira kao neki glas, što ne mora u svakom takvom segmentu biti isti glas. Glas *n* je prepoznat u redu 7, također s nekim drugim konsonantima. Nadalje, u ovom segmentu nije prepoznat vokal *i*, nego umjesto njega vokal *e*. Iako je u ovom segmentu manje očito o kojoj se riječi radi, često je dovoljno ispravno prepoznatih glasova da se kasnijim poravnanjem glasova s tekstom može dobiti dovoljno dobra aproksimacija pozicije riječi.

U segmentu 23..25 izgovorena je riječ “još”. U redu 25 vidljivo je da je neuralna mreža *š* klasificirala kao *č/ć*, što je razumljivo s obzirom da su ti glasovi gotovo isti. Glas *j* nije bio prepoznat.

U segmentu 93..97 izgovorena je riječ “račun”. Ovdje nedostaju glasovi *r* i *u*, dok su glasovi *a*, *č* i *n* ispravno prepoznati.

Na slikama 89 i 90 prikazan je popis klasificiranih glasova za druga dva muška glasa (zatamnjeni dijelovi predstavljaju zvuk koji je dio kratke pauze). Na slikama 91, 92 i 93 prikazan je popis klasificiranih glasova govora tri ženska glasa. Neuralna mreža nije bila trenirana sa ženskim glasovima. Ženski glas se u prosjeku razlikuje od muškog po spektru koji je lagano pomaknut ka višim frekvencijama i oblik spektra je nešto više zaobljen. Rezultati pokazuju da na upotrebljenom uzorku nema velikih razlika u preciznosti klasifikacije između muških i ženskih glasova. Klasifikacija na slici 92 (ženski glas) je najpreciznija od svih šest ovdje prikazanih, što će se vidjeti kasnije u prikazu rezultata poravnavanja teksta s glasovima. U dijelu 6.3 prikazani su rezultati poravnavanja teksta s glasovima, što je korak koji ovisi o preciznosti klasifikacije glasova.

R.BR.	OD	DO	DULJINA	GLASOVI
0:	10.39	10.41	16.2	aaaa
1:	10.43	10.44	10.6	aaa
2:	10.45	10.53	72.2	*vvnmmvmr
3:	10.55	10.58	37.8	eeeeeeee
4:	10.59	10.61	12.3	eeee
5:	10.61	10.61	0.0	*r
6:	10.62	10.62	10.0	eee
7:	10.63	10.66	29.9	*vnrv
8:	10.66	10.72	57.2	eeeeeeeeeeee
9:	10.74	10.76	20.3	ooooo
10:	10.77	10.78	10.0	ooo
11:	10.8	10.81	9.9	oao
12:	10.82	10.83	10.3	ooa
13:	10.85	10.89	50.0	ssssss
14:	11.0	11.08	84.7	aeeaaaaeeeeeeeeee
15:	11.12	11.16	37.9	*rmrr
16:	11.16	11.18	15.9	ooaa
17:	11.19	11.25	59.3	aaaaaaoaaaaa
18:	11.4	11.46	58.3	ooooooooooo
19:	11.46	11.52	50.0	szszzz
20:	11.52	11.64	123.0	uoooooaoaooauaooaoaooaoao
21:	11.66	11.68	10.0	aoo
22:	11.68	11.76	87.6	*rrrmrn
23:	11.77	11.81	50.0	oeeeeeeeea
24:	11.83	11.84	13.1	aea
25:	11.88	11.95	70.0	čččččččč
26:	11.97	11.97	0.0	*r
27:	11.97	11.99	20.0	eeee
28:	12.0	12.0	0.0	*r
29:	12.01	12.04	30.8	eeeeeee
30:	12.05	12.11	63.0	*mrmmmm
31:	12.12	12.15	30.0	ooooaoo
32:	12.18	12.21	30.0	uooo
33:	12.22	12.3	80.0	sssssssc
34:	12.34	12.34	0.0	*r
35:	12.35	12.43	86.7	eeeeaeeeeeeeeeee
36:	12.44	12.48	38.9	*rrmr
37:	12.5	12.52	24.5	oaaao
38:	12.53	12.63	97.4	ooaooooooaoooooooo
39:	12.64	12.68	36.7	uuuooooo
40:	12.69	12.8	113.6	ooooooooooooooaoaaaaa
41:	12.88	12.96	70.0	csssssss
42:	13.0	13.01	12.2	*vm
43:	13.13	13.2	60.0	ooooaooouuo
44:	13.2	13.23	26.2	*mmm
45:	13.27	13.3	39.4	eeaoouuo
46:	13.31	13.33	22.5	*mr
47:	13.34	13.37	32.3	oeueeee
48:	13.38	13.4	24.2	aeaie
49:	13.42	13.42	0.0	*r
50:	13.43	13.47	48.5	iieeaaaeae
51:	13.48	13.56	80.1	eeeeeeaeeeaea
52:	13.58	13.62	39.5	*rrrrrkrrrk
53:	13.63	13.65	20.0	eee
54:	13.9	13.96	57.6	iiiiieiiiiiiei
55:	13.97	14.03	53.0	iiiiiiiiiiiii
56:	14.03	14.03	7.1	*mm
57:	14.04	14.05	13.5	iiii
58:	14.06	14.06	0.0	*m

59:	14.07	14.11	42.5	eoooouoooo
60:	14.12	14.2	74.7	ooooooooooooooo
61:	14.21	14.28	60.0	zzssssszszszs
62:	14.3	14.31	10.4	eie
63:	14.32	14.33	10.8	eie
64:	14.34	14.52	179.5	*rgrrvvnnmmm
65:	14.53	14.54	16.6	uuao
66:	14.55	14.57	11.9	oaaa
67:	14.59	14.64	48.6	*nnnn
68:	14.64	14.71	66.2	ooaoaaaaaoaao
69:	14.72	14.72	0.0	*v
70:	14.72	14.75	25.7	ouuuo
71:	14.76	14.76	0.0	*r
72:	14.78	14.81	36.6	ouoooooo
73:	14.85	14.87	22.0	*vv
74:	14.88	14.95	71.6	oaaaaaaaaaaaeae
75:	14.96	14.98	22.5	eeeee
76:	14.99	15.05	59.6	eaeaaeaaaaoa
77:	15.06	15.15	81.9	aaaaaaoaooaooouoo
78:	15.16	15.34	183.6	ueeeeieieiieieieieeiiiiiiiiiiiiieei
79:	15.35	15.37	12.7	uuuu
80:	15.38	15.43	56.3	uuuuuuuuuuo
81:	15.45	15.5	44.9	*mvrm
82:	15.54	15.6	50.9	oooooaaaaa
83:	15.62	15.63	20.0	aaa
84:	15.73	15.75	14.8	aaaa
85:	15.76	15.79	32.4	aaoaooo
86:	15.8	15.8	0.0	*n
87:	15.82	15.84	17.6	oooa
88:	15.84	15.85	12.9	*nn
89:	15.87	15.88	16.1	ooaa
90:	15.97	15.99	23.5	*nnr
91:	16.02	16.03	14.6	aaaa
92:	16.16	16.26	99.6	aaaaaaaaaaaaaaaaao
93:	16.27	16.38	100.9	aaaaaaaaaaaaaaaaaaaoa
94:	16.43	16.45	20.0	ččč
95:	16.5	16.5	0.0	*n
96:	16.5	16.52	16.8	oaooo
97:	16.53	16.58	47.7	*mnnd

Slika 88: Rezultat klasifikacije glasova rečenice „a meni ostaje da pozovem još jednom sve da ovu akciju podrže i pozivima na broj ali i uplatama na taj račun“ (muški glas).

R.BR.	OD	DO	DULJINA	GLASOVI
0:	21.35	21.37	20.0	iii
1:	21.38	21.45	70.0	ssssssss
2:	21.46	21.53	67.9	aaaaaaaaaaaaaaao
3:	21.54	21.56	21.9	*ndh
4:	21.56	21.59	25.4	ooaaaoo
5:	21.6	21.63	28.6	*mnnmm
6:	21.66	21.71	45.9	oaooaaaoaaao
7:	21.72	21.83	110.0	sssssssssss
8:	21.84	21.91	69.0	ooooouooooooooo
9:	21.92	21.94	17.3	ooooo
10:	21.95	22.06	107.7	oaoaaaaaaaaaaaaaaaaaaaa
11:	22.07	22.07	0.0	*r
12:	22.07	22.09	16.0	aaaa
13:	22.09	22.11	15.9	oaaaa
14:	22.15	22.21	60.0	aoooooaaaaaao
15:	22.22	22.23	17.8	*mml
16:	22.24	22.25	11.8	aaao
17:	22.26	22.36	102.2	ooooooooooooooooooooo
18:	22.38	22.38	0.0	*n
19:	22.39	22.43	40.0	ouuouooooou
20:	22.44	22.44	0.0	*v
21:	22.45	22.49	39.9	uoooaaaaaa
22:	22.5	22.52	19.0	aaaeo
23:	22.53	22.56	30.0	ščšč
24:	22.57	22.59	20.0	sss
25:	22.66	22.78	117.6	aaaaaaaaaaaaaaaaaaaaa
26:	22.79	22.81	20.2	aaaoa
27:	22.82	22.83	10.7	aoa
28:	22.84	22.85	10.7	*tv
29:	22.85	22.94	90.0	zcszsssscs
30:	22.95	23.08	134.1	aaaaaaaaaaaaaaaaaaaaooooooooooou
31:	23.1	23.1	0.0	*m
32:	23.11	23.13	16.2	aaaaa
33:	23.13	23.15	19.5	*rrr
34:	23.16	23.21	50.0	zzsssss
35:	23.3	23.36	57.2	ieieeaeaeaeaa
36:	23.38	23.39	11.1	*rn
37:	23.4	23.48	78.7	aeeeeeeieieoeiii
38:	23.49	23.51	21.6	*nnm
39:	23.51	23.59	80.0	ččššššččč
40:	23.6	23.67	66.5	eeeeeeeeeeeeeeeaee
41:	23.69	23.72	31.7	*nlnl
42:	23.72	23.86	137.0	aaaaaaaaaaaaaaooooooooooooaaoaa
43:	23.88	23.93	50.0	zzzzzzzzzss
44:	23.94	24.06	120.2	aaaaaaaaaaaaaaaaaaaaaaaaaa
45:	24.07	24.1	33.0	*rrrmr
46:	24.11	24.15	37.7	aeeaeeaaaaa
47:	24.15	24.15	0.0	*r
48:	24.16	24.17	10.0	oaoa
49:	24.18	24.19	17.5	*lvm
50:	24.2	24.23	30.0	uouuooo
51:	24.23	24.23	0.0	*l
52:	24.24	24.25	10.8	aaa
53:	24.27	24.35	80.0	čččščšččč
54:	24.36	24.47	110.0	uooaaaaaaooooooa
55:	24.47	24.47	0.0	*l
56:	24.48	24.54	57.9	aoaaaaoaoaaaa
57:	24.55	24.59	41.6	aaeeeeeie
58:	24.62	24.7	76.5	*mmvmnlmm

59:	24.7	24.72	19.0	oooo
60:	24.73	24.82	86.6	ooooaooooooooooaooaa
61:	24.83	24.88	50.0	ssssss
62:	24.97	25.06	90.0	aouooooooooooooou
63:	25.07	25.11	41.8	*nnnnm
64:	25.12	25.18	62.1	aaeaeaaaeeeeaaa
65:	25.28	25.34	56.8	aooaoaoaaao
66:	25.34	25.34	0.0	*n
67:	25.37	25.42	44.6	aaaeaeeeii
68:	25.42	25.44	20.0	zzzz
69:	25.45	25.51	60.0	čččščšč
70:	25.53	25.53	0.0	*n
71:	25.54	25.57	25.2	aaaaoa
72:	25.6	25.6	0.0	*n
73:	25.6	25.61	10.9	uuu
74:	25.62	25.67	45.4	oaaaaoooa
75:	25.69	25.7	12.1	aaa
76:	25.71	25.73	11.6	*rm
77:	25.74	25.79	50.0	šššščž
78:	25.84	25.84	0.0	*n
79:	25.84	25.86	20.0	iiii
80:	25.86	25.86	0.0	*h
81:	25.94	25.99	50.5	aaaaeaaaeee
82:	26.01	26.01	0.0	*v
83:	26.03	26.06	21.8	aaeee
84:	26.08	26.09	11.9	eiie
85:	26.1	26.11	11.8	*gr
86:	26.18	26.29	105.4	eaaaaaaaaoaoooooaaaa
87:	26.29	26.35	60.0	zzssssss
88:	26.45	26.55	100.0	oooaooooooooaaaaaaaaaa
89:	26.55	26.56	9.0	*lr
90:	26.59	26.63	40.0	zssss
91:	26.72	26.76	38.6	ieieeeeee
92:	26.78	26.78	0.0	*d
93:	26.78	26.83	50.0	zzzzzzzzzz
94:	26.83	27.0	170.8	eieieieeeeieieeeieieeeeeeaaeaeaaeaaaaa
95:	27.02	27.05	27.7	zzzzzzz
96:	27.06	27.08	26.5	*vrr
97:	27.09	27.15	54.4	aaaaeaeaeaaa
98:	27.16	27.16	0.0	*l
99:	27.16	27.17	10.0	eee
100:	27.18	27.19	11.1	*nn
101:	27.23	27.25	20.0	uea
102:	27.33	27.38	50.0	aeiaaa
103:	27.41	27.44	30.0	aaea

Slika 89: Rezultat klasifikacije glasova rečenice „istodobno su nakon oproštaja od saborske većine u zagrebu članovi mosta u metkoviću dočekani kao sportske zvijezde“ (muški glas).

R.BR.	OD	DO	DULJINA	GLASOVI
0:	7.31	7.38	72.6	uuuiiiiiiiiiiiii
1:	7.4	7.45	51.7	*mmmmn
2:	7.46	7.52	63.7	eieeeeeeeeiiii
3:	7.53	7.54	10.1	*nm
4:	7.59	7.64	46.7	iiiiiiiiii
5:	7.74	7.75	10.1	*mm
6:	7.76	7.87	114.1	oouuouooooaaaaaaaaaaaaa
7:	7.9	7.91	14.9	aaaau
8:	7.93	7.94	13.4	žžžž
9:	7.95	7.98	28.6	čžšžšžž
10:	7.99	8.03	35.4	*nmnnn
11:	8.04	8.12	80.3	eiiieiiiiii
12:	8.15	8.21	60.0	sssssss
13:	8.26	8.26	0.0	*n
14:	8.27	8.43	159.1	oooaaaaaaaaaaaaau
15:	8.45	8.57	120.0	eeiiiiiieieieiiiiiiiiii
16:	*8.57*	*8.58*	*11.6*	**nmnn*
17:	*8.59*	*8.62*	*32.2*	*iieie*
18:	*8.67*	*8.74*	*70.0*	*eeaoaaoo*
19:	8.79	8.85	63.7	uuooooooou
20:	8.86	8.87	10.3	uii
21:	8.88	8.9	16.7	iiii
22:	8.9	8.9	0.0	*m
23:	8.92	8.93	10.4	uiu
24:	8.94	8.94	0.0	*m
25:	8.95	8.97	19.2	uuuuu
26:	8.98	8.99	9.6	uuo
27:	9.0	9.02	25.5	*vvm
28:	9.03	9.12	92.8	ouuuouooooooooooooooooo
29:	9.14	9.14	6.4	*rr
30:	9.15	9.16	13.2	iaei
31:	9.17	9.19	21.9	iiiiii
32:	9.2	9.2	0.0	*h
33:	9.2	9.22	20.0	zzzz
34:	9.23	9.29	60.0	ččččččč
35:	9.31	9.38	67.5	iiiiieiiiiiiii
36:	9.39	9.46	75.5	*mmmmmm
37:	9.47	9.49	20.0	iiii
38:	9.54	9.69	153.6	iiiiiieieeeaaeeaaaaoaaaaaoooou
39:	9.7	9.7	0.0	*n
40:	9.71	9.72	9.3	ooo
41:	9.73	9.85	123.2	*rnnnnnmmmmmh
42:	9.86	9.95	90.0	šššščšššššššš
43:	9.98	10.0	18.8	uuuuu
44:	10.02	10.07	50.0	uuuuuouuuuu
45:	10.08	10.1	20.9	*nnh
46:	10.11	10.12	9.6	iii
47:	10.13	10.16	30.4	*nnk
48:	10.16	10.28	120.0	ueieeeeeeiaaoooo
49:	10.38	10.4	20.0	ouo
50:	10.57	10.59	20.0	ščž
51:	10.6	10.65	50.0	ieeeeaaaaaai
52:	10.65	10.67	20.7	*rm
53:	10.74	10.78	36.3	iiiiiiiii
54:	10.79	10.83	44.1	*mmmrml
55:	10.83	10.91	83.9	ieieieaaaaooooooooo
56:	10.92	11.04	111.8	ooooooooouooooooaooaoaa
57:	11.04	11.04	0.0	*n
58:	11.05	11.18	130.0	aoaooooooooaooooooooooooo

59:	11.29	11.31	27.4	*nnnn
60:	11.32	11.41	87.5	aieaoaaeeaaaaiaaiaaa
61:	11.41	11.5	90.0	ščšššššščžč
62:	11.55	11.55	0.0	*n
63:	11.56	11.57	7.7	oou
64:	11.58	11.58	0.0	*l
65:	11.59	11.62	29.4	ouuuuuu
66:	11.63	11.64	9.4	uuu
67:	11.65	11.66	9.5	uuu
68:	11.67	11.68	13.4	uuou
69:	11.69	11.72	38.8	*nnmnm
70:	11.73	11.77	41.3	eeeeiiiiii
71:	11.78	11.96	174.6	*mnnmmmnngnm
72:	11.97	11.98	12.8	iii
73:	12.0	12.05	48.1	*nnnrn
74:	12.05	12.08	30.0	oouuuuuu
75:	12.08	12.1	24.0	*mrmm
76:	12.11	12.16	52.7	iiiiiiiiieie
77:	12.29	12.35	64.2	aeaaaaaaaaa
78:	12.37	12.43	63.2	aaaaaaaaaaaaa
79:	12.44	12.47	34.2	*nnnn
80:	12.48	12.53	53.3	iiiiiiiiiiii
81:	12.54	12.61	62.3	*mmmmmmmn
82:	12.61	12.67	55.4	aaaaaaaaaaaa

Slika 90: Rezultat klasifikacije glasova rečenice „u ime tih važnijih stvari odgovorit ću im javnom šutnjom čak i na ono na što u nekim drugim danima“ (muški glas).

R.BR.	OD	DO	DULJINA	GLASOVI
0:	7.85	7.91	60.0	oooaaaaaaaaaa
1:	7.91	7.91	0.0	*k
2:	7.91	7.93	16.3	aaoaoaa
3:	7.93	7.94	8.2	*kn
4:	7.95	8.12	168.5	aooououooooooooooooooaooooooaoaaaaaaooooaoaaaaaaaaoaaaaa
5:	8.12	8.17	50.0	zzzzzzzszs
6:	8.2	8.22	23.2	aaaaaaaa
7:	8.23	8.23	0.0	*k
8:	8.25	8.37	113.1	aooooooouoouuuuuuuoouuuuouuuuuuouuuu
9:	8.38	8.44	64.5	zsssssssc
10:	8.5	8.5	0.0	*r
11:	8.5	8.54	38.4	aaaaaaaaaaaaa
12:	8.54	8.58	33.0	aaoaooooooo
13:	8.58	8.67	88.4	*vvdnvn
14:	8.68	8.69	9.2	ooo
15:	8.69	8.72	23.4	ooaooooo
16:	8.73	8.73	0.0	*d
17:	8.73	8.77	40.0	szscss
18:	8.79	8.81	20.0	ccc
19:	8.84	8.85	13.4	*vvvv
20:	8.85	8.94	84.8	uooouooooooooooooooooooooo
21:	8.95	8.95	0.0	*k
22:	8.96	9.01	47.9	ooooooooooooooa
23:	9.02	9.05	33.4	*vvvvvvv
24:	9.05	9.1	42.9	eeiiiiiiiiiiii
25:	9.11	9.11	0.0	*n
26:	9.11	9.13	18.3	iiiiui
27:	9.13	9.14	4.7	*vv
28:	9.19	9.23	37.6	iiiiiiiiiiii
29:	9.24	9.24	5.6	*nn
30:	9.25	9.26	10.0	iii
31:	9.31	9.5	189.5	ouooooooooooooaaaaoooaouuauaaaeeeeeeiiiiiiiiiiiiiiiiii
32:	9.51	9.56	50.2	*nnmttttn
33:	9.56	9.59	34.9	eiiiiiiiiiee
34:	9.6	9.61	12.3	szsss
35:	9.64	9.68	40.0	ččččč
36:	9.72	9.76	35.5	iiieeeee
37:	9.76	9.77	11.0	eeie
38:	9.78	9.78	0.0	*n
39:	9.79	9.9	110.9	zzzsssssssssss
40:	9.92	9.93	3.9	*vk
41:	9.93	9.94	5.0	ouo
42:	9.94	9.94	0.0	*k
43:	9.94	9.95	8.9	uou
44:	9.95	10.01	54.3	uouuouooooooououuou
45:	10.01	10.02	9.5	*vv
46:	10.03	10.11	80.0	sssssczc
47:	10.14	10.16	21.1	aaaaaaaa
48:	10.16	10.18	13.0	aaaaa
49:	10.19	10.19	0.0	*k
50:	10.19	10.24	46.8	aaaaaaaaaaaaooao
51:	10.25	10.27	23.6	oaaaaaa
52:	10.34	10.37	26.8	eaaeaeeea
53:	10.37	10.44	66.4	eeeeeeeeeeeeeeeeeeeaa
54:	10.44	10.49	50.9	*nnnnnnnnr
55:	10.52	10.56	32.9	aaaaaaaeaee
56:	10.58	10.6	23.3	*kkkkkk

57: 10.61 10.62 9.2 eeee
58: 10.63 10.64 13.8 *kkkk
59: 10.65 10.71 64.4 eeeeeiiiieeiiiiiiiiii
60: 10.72 10.72 9.2 *ttt
61: 10.75 10.77 20.0 ščč
62: 10.78 10.81 26.9 iieeieeeee
63: 10.82 10.86 38.6 eiiiiii
64: 10.96 11.05 85.1 ooooooooooaaaaaaaaaaaaaaaa
65: 11.05 11.05 0.0 *k
66: 11.05 11.06 12.5 aaaoo
67: 11.07 11.09 19.2 *nnk
68: 11.09 11.16 70.0 aaaaaaaaaaaaaaaaaaaooao
69: 11.16 11.16 0.0 *n
70: 11.17 11.18 11.5 ooooo
71: 11.18 11.18 0.0 *v
72: 11.19 11.24 57.3 oooooooooooooooooo
73: 11.25 11.25 0.0 *k
74: 11.25 11.28 28.1 ooooooooo
75: 11.28 11.31 28.8 ooaooaaaao
76: 11.32 11.32 5.1 *rd
77: 11.33 11.34 7.1 ččč
78: 11.36 11.37 14.1 zszzz
79: 11.38 11.38 5.2 ooo
80: 11.39 11.39 0.0 *r
81: 11.39 11.41 15.2 oooaa
82: 11.42 11.42 5.5 zzz
83: 11.43 11.43 0.0 *v
84: 11.43 11.51 82.1 zzssssszzzzz
85: 11.54 11.55 10.7 *kk
86: 11.56 11.58 20.0 aaeaeeu
87: 11.58 11.62 39.0 *vrvvvv
88: 11.63 11.64 16.4 oooooo
89: 11.65 11.65 0.0 *v
90: 11.65 11.7 51.8 iiiiiiiiiiiiieii
91: 11.71 11.71 0.0 *k
92: 11.71 11.72 6.8 iee
93: 11.72 11.75 28.9 eeeeeeeeee
94: 11.76 11.76 0.0 *k
95: 11.76 11.78 18.3 eeeeiee
96: 11.79 11.81 22.3 *vvvv
97: 11.82 11.86 39.0 eeeieiiiiiiiii
98: 11.86 11.89 27.8 iiiiiiuii
99: 11.89 11.97 80.4 *nm
100: 11.97 12.26 289.9
iieiiiiiiiiiiiiiiiiiieeeeieiiaeeioouooooooooooooooooooooooooooooooooooaoo
oo
101: 12.27 12.27 0.0 *r
102: 12.27 12.28 10.0 ouuu
103: 12.28 12.29 5.1 *mn
104: 12.29 12.31 24.6 uuuuuuuu
105: 12.32 12.32 0.0 *n
106: 12.32 12.46 139.4
oouooooooooooooooooooooooooooooooooooooaoaoooo
107: 12.47 12.47 0.0 *v
108: 12.47 12.55 80.0 zzscsscsss
109: 12.58 12.61 26.1 *vgvvvrr
110: 12.62 12.64 14.0 aaooo
111: 12.64 12.64 0.0 *p
112: 12.64 12.7 60.0 aaaaaaoaauuooouooo
113: 12.7 12.7 0.0 *r
114: 12.71 12.71 5.2 ouu

```
115:    12.72     12.72     5.4       *mm
116:    12.73     12.85     128.2
uaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaeeeeee
117:    12.86     12.87     8.7       *vn
118:    12.87     12.88     11.1      aaaoe
119:    12.95     12.96     12.4      *gmnn
120:    12.97     13.0      22.2      ieaiieie
121:    13.01     13.02     19.3      *nnnn
122:    13.03     13.09     58.5      ooooooooooooaoeeee
123:    13.09     13.16     63.4      eieeeieiiieieiioiieii
124:    13.17     13.17     0.0       *n
125:    13.17     13.19     20.0      ššš
126:    13.21     13.23     20.0      šš
127:    13.29     13.39     101.6     ieeeeeeeaeeeeeeeeeeeeeeeieeuuuouo
128:    13.4      13.4      0.0       *r
129:    13.4      13.56     157.9
ouuoueuiiuieeeeeeeeeaeaaaaaaaaaaaaaaaaaoaoouo
130:    13.56     13.57     6.6       uau
131:    13.58     13.58     0.0       *n
132:    13.58     13.6      17.0      ooouo
133:    13.61     13.63     20.0      oao
134:    13.67     13.71     39.9      aaaooooooooo
135:    13.71     13.74     21.9      oooooooo
136:    13.75     13.75     0.0       *k
137:    13.75     13.83     75.8      ooooooooooooaoaaaaaaaaaaae
138:    13.83     13.86     22.2      eeeeeeee
139:    13.86     13.87     14.5      *kkkk
140:    13.88     13.89     10.0      aeaa
141:    14.26     14.29     28.7      *kkkkkkkg
```

Slika 91: Rezultat klasifikacije glasova rečenice „odbor za ustav poslovnik i politički sustav treba reći tko ga može zamijeniti odnosno dati mišljenje o tome“ (ženski glas).

R.BR.	OD	DO	DULJINA	GLASOVI
0:	0.36	0.53	167.0	
iiiieeieieeeeeeeeeaeeaaaaeaaaaoaooooooooooooooooooooooao				
1:	0.53	0.58	48.8	aaaaaaaaaouiiui
2:	0.58	0.6	17.7	*nnnn
3:	0.63	0.68	54.1	aeeeeeeeeeeeeeeeeeea
4:	0.69	0.69	0.0	*n
5:	0.69	0.8	110.0	eeeeeeaeeeeeeaeeaeaaaaaaaeaieeeeeieee
6:	0.8	0.82	17.5	*kkk
7:	0.82	0.84	20.0	eeieeae
8:	0.84	0.84	0.0	*k
9:	0.9	0.92	21.1	iiieeeae
10:	0.93	0.93	0.0	*r
11:	0.93	0.94	5.5	oou
12:	0.94	0.97	23.8	*nnnnl
13:	0.97	1.07	100.4	aaieeeeeieeeeeeeeeeeeeeeeeeeeuii
14:	1.08	1.08	5.1	*tt
15:	1.09	1.11	20.0	sss
16:	1.18	1.19	5.8	eee
17:	1.19	1.24	47.5	*rnrnnnnrnr
18:	1.24	1.26	14.4	eeiee
19:	1.28	1.3	19.3	*vvvtt
20:	1.31	1.34	30.0	scsc
21:	1.37	1.39	16.8	eeeeee
22:	1.4	1.44	45.5	*nnnrnnnnn
23:	1.45	1.5	56.2	ieeeeieeeeieiieiii
24:	1.51	1.53	20.4	*nnmn
25:	1.54	1.55	10.1	iiii
26:	1.55	1.57	15.1	*nv
27:	1.59	1.62	30.3	ieeiiiiiii
28:	1.62	1.72	102.1	*nnnnn
29:	1.73	1.8	68.3	iiiiiiiiiiiiiiiiii
30:	1.82	1.82	0.0	*n
31:	1.82	1.83	9.6	iii
32:	1.83	1.85	20.0	ssss
33:	1.89	1.92	30.0	čččč
34:	1.94	1.96	20.0	ččč
35:	1.97	2.02	49.4	eeeeeeeeeeeea
36:	2.03	2.03	6.9	aee
37:	2.04	2.05	6.7	*vn
38:	2.05	2.08	26.2	iiaiiii
39:	2.08	2.08	0.0	*t
40:	2.26	2.29	30.0	čščč
41:	2.31	2.43	123.4	ieeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeei
42:	2.44	2.46	20.0	sss
43:	2.47	2.59	123.6	ščščšččččččččč
44:	2.6	2.69	91.4	ieeeeeaaaaaaaaaaooooooooooa
45:	2.69	2.74	50.8	*nnmnmmm
46:	2.78	2.81	28.0	oooooooo
47:	2.81	2.88	66.0	ooooooaooooooooooo
48:	2.88	2.92	44.7	*kkkkk
49:	2.93	2.95	21.0	aaaaaee
50:	2.97	3.0	34.1	oeeeeie
51:	3.01	3.04	34.1	eeeeaaaaa
52:	3.05	3.12	65.2	*rvvvvrvn
53:	3.12	3.14	17.0	aaaaaa
54:	3.16	3.25	85.0	aaaaaaaaaaaaaauu
55:	3.25	3.25	0.0	*t
56:	3.26	3.37	110.0	ccssssscsscss
57:	3.37	3.42	45.8	aaaaaaaaaaaaa

58:	3.42	3.44	17.0	aaaaau
59:	3.44	3.44	0.0	*r
60:	3.45	3.46	5.9	sss
61:	3.46	3.49	31.5	*nntrr
62:	3.5	3.57	66.4	szzszzzsszzszzzszzz
63:	3.57	3.63	57.8	eaeaeeeeeeaaeeaee
64:	3.63	3.63	0.0	*n
65:	3.64	3.65	11.2	aeaaa
66:	3.66	3.66	0.0	*k
67:	3.66	3.67	8.5	aaa
68:	3.67	3.68	6.1	aaa
69:	3.69	3.69	6.1	*nn
70:	3.7	3.72	18.3	aaaaaa
71:	3.72	3.74	12.6	aaaa
72:	3.74	3.81	72.5	aaaaaaaaaaaaaaaaaoo
73:	3.82	3.83	13.3	*nnm
74:	3.93	3.96	25.1	aaaaaaa
75:	3.97	3.97	5.1	*kk
76:	3.98	4.07	89.4	aaaaaaaaaaaaaaaaaaaaaaaaaaaa
77:	4.07	4.1	27.2	*nnnnn
78:	4.1	4.19	92.3	aaaaaaaaaaaaaaaaaeaaaaeaaea
79:	4.2	4.21	6.9	*nt
80:	4.23	4.27	40.0	sssss
81:	4.29	4.34	50.8	aeeeeeeaeaeaia
82:	4.35	4.36	14.3	*mnn
83:	4.4	4.47	65.2	aaaaaaaaaaaaaaaa
84:	4.49	4.49	0.0	*n
85:	4.49	4.51	13.0	aaaa
86:	4.51	4.62	105.3	aaaaaeaaaaaaaaaaaaaaaaaaaaa
87:	4.63	4.63	0.0	*r
88:	4.63	4.7	70.0	zzccscscc
89:	4.77	4.87	91.9	*nnnnnnnttvvg
90:	4.87	4.9	30.0	ssss
91:	4.95	5.03	80.0	iiuiiiiea
92:	5.11	5.11	0.0	*n
93:	5.11	5.18	70.0	aaaaaaaaaaaaaaaaaaa
94:	5.18	5.18	0.0	*k
95:	5.19	5.19	6.2	aaa
96:	5.2	5.2	0.0	*k
97:	5.2	5.21	11.7	aaaa
98:	5.22	5.22	6.1	*kk
99:	5.23	5.27	40.0	aaaaaaaaaaae
100:	5.27	5.37	101.7	*rmnrrrrnnnnnnnlnvn
101:	5.38	5.45	70.9	ioiueeiiiiieiiiiiiiii
102:	5.45	5.45	0.0	*v
103:	5.46	5.54	79.4	zzzzzssssssz
104:	5.54	5.54	0.0	*v
105:	5.55	5.57	23.2	eeeeeeei
106:	5.57	5.57	0.0	*n
107:	5.58	5.66	76.3	eeeeeaeeeaeaeaaaaaaaaaaa
108:	5.66	5.68	18.5	aaaaaae
109:	5.69	5.7	9.5	*kr
110:	5.7	5.74	38.1	eaeaeaaaeaaaa
111:	5.75	5.76	14.0	*kkkk

Slika 92: Rezultat klasifikacije glasova rečenice „i u utorak će mjestimice biti kiše češće u gorju a s obzirom na to da će do nas stići hladniji zrak“ (ženski glas*).*

```
R.BR.  OD        DO        DULJINA   GLASOVI
-----  --        --        -------   -------
0:     3.18      3.34      155.2
ooooooooooooooooooooooooooouooooooooooooouoaoo
1:     3.34      3.35      12.2      *nrt
2:     3.36      3.38      20.0      ccs
3:     3.43      3.48      46.8      aaaaaaaaaaaaa
4:     3.48      3.48      0.0       *d
5:     3.56      3.76      202.8
aaaaaaaaooooouooooooooouooooooooooooooooooooooooouo
6:     3.77      3.77      0.0       *n
7:     3.77      3.79      22.2      uooouuo
8:     3.8       3.8       0.0       *n
9:     3.8       3.86      64.4      aaaaaaeeaaaeeeeeeee
10:    3.88      3.88      0.0       *r
11:    3.96      4.05      90.5      eieeeaaeaaeeeaeeeeoeuo
12:    4.06      4.07      14.0      *nn
13:    4.08      4.09      7.0       uii
14:    4.09      4.13      34.6      *vr
15:    4.13      4.2       65.2      ooeeeeeieeeeiieee
16:    4.2       4.47      268.1
eeeeeeeeaeaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaoaaoaooooooooaaaaeaaaoaooooooooo
o
17:    4.47      4.49      11.2      *vv
18:    4.49      4.59      100.0     zsssscccssss
19:    4.66      4.66      0.0       *n
20:    4.67      4.69      18.1      iiiiii
21:    4.69      4.72      28.1      iiiieiii
22:    4.74      4.78      35.0      oooooouou
23:    4.79      4.81      23.2      *vnm
24:    4.91      5.06      149.8     oooooaaaaaaaaaaaaaaaaeaeeeeeeeeeeeee
25:    5.07      5.16      85.5      *nnnnnnnnv
26:    5.17      5.17      7.1       eee
27:    5.18      5.2       14.0      eeee
28:    5.2       5.26      62.1      eeeeoeeaeeeeeeee
29:    5.37      5.39      20.0      eie
30:    5.39      5.39      0.0       *n
31:    5.4       5.47      77.8      eeeeeeeeeeeeeeeeeeeee
32:    5.49      5.49      0.0       *v
33:    5.51      5.56      50.0      sscscs
34:    5.58      5.58      0.0       *j
35:    5.58      5.62      37.0      eeeeeeeeee
36:    5.63      5.64      13.5      *nvv
37:    5.65      5.66      13.9      ououo
38:    5.67      5.7       36.2      *vnnnt
39:    5.77      5.82      54.4      aaaaaaaaaaaaee
40:    5.91      6.0       91.1      aaaaaaaaaaaaaaaaaaaeeeee
41:    6.01      6.01      0.0       *n
42:    6.02      6.02      6.6       ooo
43:    6.03      6.04      13.1      *vvv
44:    6.05      6.09      40.7      ioeeeeieeeie
45:    6.1       6.1       7.2       *vv
46:    6.11      6.2       88.5      zzzzssssszz
47:    6.2       6.21      5.7       *vv
48:    6.21      6.28      67.5      eeeeeeeeeeeeeeeeeee
49:    6.28      6.3       11.9      eaae
50:    6.31      6.31      0.0       *k
51:    6.31      6.35      43.8      aaaaaaaaaaaa
52:    6.36      6.4       34.6      aaaaaaaa
53:    6.47      6.6       126.2     aaaaaaaaaoaaaaaaaoeooooooaoooooaoa
54:    6.6       6.6       0.0       *r
```

55: 6.61 6.71 100.5 eeeeeeeeeeeeeeeeeeeaaaaa
56: 6.72 6.76 44.1 aaaaaaaaaaaaaaaaaaaaa
57: 6.87 6.98 107.8 aaaaaaaeaeaaeeeeeeeeeeeeeeeeeeeoao
58: 6.99 6.99 5.0 *nn
59: 7.0 7.01 10.0 uuou
60: 7.01 7.08 64.8 *mrnrn
61: 7.08 7.14 60.1 ooooaooaooaoaoaaaooa
62: 7.14 7.14 0.0 *r
63: 7.15 7.18 26.7 aeeeeeeee
64: 7.18 7.2 18.8 eeieei
65: 7.23 7.29 60.0 sssssss
66: 7.3 7.35 46.6 ieeeaeeeeaaa
67: 7.35 7.42 67.0 aaaaaaaaaaaaaa
*68: 7.42 7.43 7.8 *kk*
69: 7.43 7.45 20.0 aaoaaa
70: 7.47 7.49 20.0 aaa
71: 7.84 8.0 157.1
aaaaaaaaaaaaaaaaaoaooaooooooooooooooooooooaoaa
72: 8.02 8.04 24.6 zzzzzzzzc
73: 8.04 8.04 0.0 *v
74: 8.05 8.07 19.8 zzzzzzz
75: 8.07 8.08 9.8 zzzz
76: 8.09 8.09 0.0 *r
77: 8.09 8.1 6.8 zzz
78: 8.1 8.12 14.6 *vdd
79: 8.12 8.27 153.3
aoaaaaaaaoaaaaaaaaaaaaaaooaaaooooaaaoaaao
80: 8.3 8.34 40.0 zszzzz
81: 8.35 8.35 0.0 *v
82: 8.35 8.4 54.3 aaaeaeaeeeeeeee
83: 8.41 8.46 44.0 *vndnvnv
84: 8.46 8.7 234.1
iiiiiiiiiiiiiiieieeieaeeeeeaeeaaaoaooooooooooooooooooooooooooooouoo
85: 8.71 8.72 16.1 *nnn
86: 8.73 8.82 89.3 oaaaaaaaaaaaaaaaaeeeeeeiee
87: 8.83 8.83 0.0 *v
88: 8.84 8.86 23.7 oooouoo
89: 8.87 8.88 18.1 *vvv
90: 8.89 9.0 110.1 iiieiiiieiieeuieeeieeeeeeeei
91: 9.09 9.22 130.0 scsssssssssss
92: 9.24 9.28 40.0 ouuuouuouooooou
93: 9.28 9.31 20.4 *vvv
94: 9.36 9.38 18.0 oaaaaaa
95: 9.38 9.39 4.8 *rr
96: 9.39 9.42 27.6 aaaaaaoao
97: 9.43 9.44 6.4 *vr
98: 9.5 9.55 51.2 aaoaaaaooooooa
99: 9.56 9.56 0.0 *m
100: 9.57 9.6 30.0 sssc
101: 9.71 9.79 80.0 eaaaaaaaaaaaaaaaaaaaa
102: 9.79 9.79 0.0 *r
103: 9.8 9.87 68.3 aaaaaaaaaaaaaaaaae
104: 9.88 9.88 7.4 *rr
105: 9.89 9.9 10.0 szss
106: 9.93 9.95 20.0 ččč
107: 10.03 10.06 27.4 ieoiiiiu
108: 10.07 10.07 0.0 *n
109: 10.07 10.08 10.0 uui
110: 10.08 10.08 0.0 *m
111: 10.09 10.1 7.1 uuu
112: 10.11 10.25 136.2 *mmnnnnvnvvvvn

```
113:    10.31     10.36     50.0      eaeeee
114:    10.39     10.41     17.9      eae
115:    10.42     10.42     7.5       eee
116:    10.43     10.44     8.1       *vv
117:    10.45     10.56     110.0     sssssssssss
```

Slika 93: Rezultat klasifikacije glasova rečenice „polusatnom obraćanju javnosti u kojem je predsjednika hadezea andreja plenkovića prozvao za djelovanje suprotno stranačkim interesima“ (ženski glas).

6.3 Poravnavanje teksta s glasovima

U ovom dijelu prikazani su rezultati poravnanja glasova s tekstom. U tablicama 8 do 13 rezultati su prikazani u pet stupaca: riječ, početak riječi (ISPRAVNO-OD) utvrđen slušanjem, završetak riječi (ISPRAVNO-DO) utvrđen slušanjem, odstupanje od početka riječi (ODSTUPANJE-OD) i odstupanje od završetka riječi (ODSTUPANJE-DO). Odstupanja su mjerena u milisekundama. Pozitivna vrijednost odstupanja znači da je odstupanje veće od ispravne vrijednosti, a negativna da je manje. Na primjer, za odstupanje-od = 28.5 i odstupanje-do = -101.2 znači da je poravnanje označilo početak riječi 28.5 milisekundi prekasno, te da je označilo završetak riječi 101.2 milisekundi prerano.

U tablicama 8, 9 i 10 prikazani su rezultati poravnavanja teksta s glasovima tri muška glasa, a u tablicama 11, 12 i 13 poravnanje s tri ženska glasa (neuralna mreža nije trenirana sa ženskim glasovima, stoga su rezultati s takvim glasovima posebno značajni). U sljedećem dijelu dan je sažetak podataka u ovim tablicama.

Prva tablica

- Popis riječi
- Interval svake riječi utvrđen slušanjem
- Odstupanje od ispravnog intervala koje je rezultiralo postupkom poravnanja teksta sa zvukom

Druga tablica

- Ukupan broj riječi
- Maksimalno odstupanje (u milisekundama) – Ovo je par u kojem je prva vrijednost maksimalna apsolutna vrijednost za ODSTUPANJE-OD, a druga maksimalna apsolutna vrijednost za ODSTUPANJE-DO.
- Minimalno odstupanje (u milisekundama) – Slično kao za maksimalno odstupanje, ali pokazuje minimalne apsolutne vrijednosti.
- Prosječno odstupanje (br. glasova) – Prosječno odstupanje izmjereno prema broju glasova (umjesto milisekundi). Prema (Babić, i dr., 1991) prosječno trajanje glasa je oko 76 milisekundi.
- Prosječno odstupanje (ms) – Ovo je par s apsolutnim vrijednostima prosječnog odstupanja za ODSTUPANJE-OD i ODSTUPANJE-DO.

Treća tablica

Druga tablica pokazuje koji je postotak i broj riječi kod kojih je odstupanje manje od onog prikazanog u prvom stupcu. Na primjer, u sažetku za tablicu 8 broj riječi kod kojih je odstupanje bilo za manje od tri glasa je 16, odnosno 72.73%.

Tablica 8: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 88.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
a	10.39	10.41	15	-58.8
meni	10.45	10.72	-25.6	-17.8
ostaje	10.74	11.08	-7.2	-37.4
da	11.12	11.18	-0.8	-89.1
pozovem	11.19	11.81	-198.1	65
još	11.83	11.97	76.7	24.6
jednom	11.97	12.21	15	5
sve	12.22	12.43	15	-18.3
da	12.44	12.63	-35.9	60.2
ovu	12.69	13.2	121.4	485
akciju	13.27	13.47	545.6	473.5
podrže	13.48	14.03	384.9	483.1
i	14.04	14.05	135	48.5
pozivima	14.06	14.57	63.6	-34.6
na	14.59	14.71	-0.3	-8.8
broj	14.72	14.87	4.2	-81.3
ali	14.88	15.05	-85	-198.6
i	15.06	15.15	-196.8	-184.9
uplatama	15.16	15.79	-192.8	-175
na	15.8	15.84	-165.8	-245
taj	15.84	15.99	-268	-287.1
račun	16.02	16.58	-265	6.4

Ukupan broj riječi	22
Maksimalno odstupanje (ms)	(545.6, 485.0)
Minimalno odstupanje (ms)	(0.3, 5)
Prosječno odstupanje (br. glasova)	(1.7, 1.8)
Prosječno odstupanje (ms)	(128.08, 140.36)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	8	36.36
2	12	54.55
3	16	72.73
4	19	86.36
5	19	86.36

Tablica 9: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 89.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
istodobno	21.35	21.94	0	215.4
su	21.95	22.09	222.3	166
nakon	22.09	22.38	194.1	131.3
oproštaja	22.39	22.83	90	28.6
od	22.84	22.85	39.2	-0.1
saborske	22.95	23.39	100	0
većine	23.4	23.86	0	60.6
u	23.88	23.93	80	55
zagrebu	23.94	24.23	37.2	-40
članovi	24.23	24.7	-16.3	59
mosta	24.7	25.18	0	182.1
u	25.28	25.34	230	226.8
metkoviću	25.34	25.67	235	67.5
dočekani	25.69	26.06	41.2	-23.5
kao	26.08	26.29	-71.6	0
sportske	26.29	27	0	229
zvijezde	27.02	27.38	220	0

Ukupan broj riječi	17
Maksimalno odstupanje (ms)	(235.0, 229.0)
Minimalno odstupanje (ms)	(0.0, 0.0)
Prosječno odstupanje (br. glasova)	(1.2, 1.1)
Prosječno odstupanje (ms)	(92.76, 87.35)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	6	35.29
2	9	52.94
3	14	82.35
4	17	100
5	17	100

Tablica 10: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 90.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
u	7.31	7.38	20	52.6
ime	7.4	7.54	72.4	44.4
tih	7.59	7.87	40	153.7
važnijih	7.9	8.43	176.6	330
stvari	8.45	8.87	320	323.1
odgovorit	8.88	9.16	83.3	-36.7
ću	9.2	9.2	-0.4	-110.4
im	9.2	9.29	-110	-180
javnom	9.31	9.85	-160	4.6
šutnjom	9.86	10.4	10	70
čak	10.57	10.67	10	-65.6
i	10.74	10.78	0	-3.7
na	10.79	10.91	5.1	13.9
ono	10.92	11.18	23.3	30
na	11.29	11.41	16.2	7.5
što	11.41	11.57	10	-52.7
u	11.58	11.58	-36.3	-76.3
nekim	11.59	11.72	-70	-235.1
drugim	11.73	12.1	-230	-165.7
danima	12.11	12.67	-160	-44.6

Ukupan broj riječi	20
Maksimalno odstupanje (ms)	(320.0, 330.0)
Minimalno odstupanje (ms)	(0.0, 3.7)
Prosječno odstupanje (br. glasova)	(1.0, 1.3)
Prosječno odstupanje (ms)	(77.68, 100.03)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	9	45
2	12	60
3	16	80
4	18	90
5	20	100

Tablica 11: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 91.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
odbor	7.85	8.12	0	5.6
za	8.12	8.22	10	-40.6
ustav	8.25	8.67	-7.7	89.3
poslovnik	8.68	9.11	20.2	-3.7
i	9.11	9.13	-78.3	-110
politički	9.13	9.77	-188.1	-8.1
sustav	9.79	10.18	9.1	-53.4
treba	10.19	10.56	-144.6	-3.7
reći	10.61	10.86	53.4	30
tko	11.05	11.16	119.2	110
ga	11.17	11.24	118.5	94.8
može	11.25	11.39	90	-30.6
zamijeniti	11.42	11.89	-3.9	-160
odnosno	11.97	12.7	-145.8	-20
dati	12.71	13	-14.8	5.6
mišljenje	13.01	13.56	15.1	77.9
o	13.56	13.57	83.7	20.3
tome	13.58	13.83	-73	-60.8

Ukupan broj riječi	18
Maksimalno odstupanje (ms)	(188.1, 160.0)
Minimalno odstupanje (ms)	(0.0, 3.7)
Prosječno odstupanje (br. glasova)	(0.9, 0.7)
Prosječno odstupanje (ms)	(65.3, 51.36)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	7	38.89
2	16	88.89
3	18	100
4	18	100
5	18	100

Tablica 12: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 92.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
i	0.36	0.53	9.5	136.5
u	0.53	0.58	131	149.8
utorak	0.63	0.84	198.5	1.9
će	0.93	0.93	85.9	-4.1
mjestimice	0.93	1.39	0	-3.2
biti	1.4	1.62	-42.8	-50
kiše	1.73	2.02	56.8	-60.6
češće	2.03	2.69	-213.5	50
u	2.78	2.81	140	108
gorju	2.88	3.25	140.3	145
a	3.25	3.25	101.8	11.8
s	3.26	3.37	20	10
obzirom	3.42	3.67	63	-81.5
na	3.69	3.72	-63.3	-102.8
to	3.72	3.81	-186.6	-255.1
da	3.82	3.96	-268.4	-244.9
će	4.1	4.19	-120	-157.7
do	4.23	4.36	-170	-107.4
nas	4.49	4.7	15.6	-10
stići	4.87	5.18	150	220.8
hladniji	5.19	5.45	207	10.1
zrak	5.46	5.7	10.6	-54.3

Ukupan broj riječi	22
Maksimalno odstupanje (ms)	(268.4, 255.1)
Minimalno odstupanje (ms)	(0.0, 1.9)
Prosječno odstupanje (br. glasova)	(1.4, 1.2)
Prosječno odstupanje (ms)	(108.85, 89.8)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	6	27.27
2	14	63.64
3	20	90.91
4	22	100
5	22	100

Tablica 13: Rezultat poravnanja teksta s klasificiranim glasovima sa slike 93.

RIJEČ	**ISPRAVNO-OD**	**ISPRAVNO-DO**	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**
polusatnom	3.18	3.88	30	247.2
obraćanju	3.96	4.59	320	460
javnosti	4.66	5.2	481.9	475
u	5.2	5.26	471.9	464
kojem	5.37	5.56	510	430
je	5.58	5.62	447.8	349.5
predsjednika	5.63	6.21	276.6	379.9
hadezea	6.21	6.6	320	192.5
andreja	6.61	7.14	188.8	360
plenkovića	7.14	8	293.7	597.1
prozvao	8.02	8.27	185.4	-6.7
za	8.3	8.35	20	-64.4
djelovanje	8.35	9	-110	-10
suprotno	9.09	9.56	10	8
stranačkim	9.57	10.08	10	-38.3
interesima	10.09	10.56	-41	-230

Ukupan broj riječi	16
Maksimalno odstupanje (ms)	(510.0, 597.1)
Minimalno odstupanje (ms)	(10.0, 6.7)
Prosječno odstupanje (br. glasova)	(3.1, 3.5)
Prosječno odstupanje (ms)	(232.32, 269.54)

BR. GLASOVA ODSTUPANJA	BROJ RIJEČI	POSTOTAK RIJEČI
1	3	18.75
2	4	25
3	5	31.25
4	7	43.75
5	10	62.5

6.4 Primjeri detekcije naglašenih riječi

Detekcija naglašenih riječi podijeljena je u tri aspekta:

- Detekcija pojačanog intenziteta (oznaka I)
- Detekcija povišenog tona (oznaka T)
- Detekcija produljenog izgovora vokala (oznaka R)

Dodatno ovim oznakama upotrebljene su i sljedeće oznake kojima smo označili ispravnost detekcije prema onome što se po našoj procjeni čuje na snimci:

- „+“: ispravno detektirana naglašenost prema slušanju
- „-„: nedetektirana naglašenost prema slušanju
- „?“: prema slušanju nije jasno je li riječ naglašena ili nije
- “>”: prema slušanju naglašenost se (vjerojatno) odnosi na sljedeću riječ
- „<“: prema slušanju naglašenost se (vjerojatno) odnosi na prethodnu riječ

U ovim uzorcima upotrebljeni su sljedeći parametri:

- Odstupanje za ton: 20%
- Odstupanje za intenzitet: 5%
- Odstupanje za trajanje: 70%
- Veličina prozora za trajanje: 6

U tablicama 14, 15, 16, 17, 18 i 19 nalaze se rezultati detekcije naglašenosti za koje su upotrebljeni podaci u stupcima INTENZITET, TON i TRAJANJE. Intenzitet je prikazan s tri vrijednosti: minimalna vrijednost u intervalu riječi, maksimalna vrijednost intervala, te srednja vrijednost intervala. Ove su vrijednosti prikazane u formatu MIN/MAKS/PROSJEK. Na isti način prikazane su i vrijednosti za ton. Vrijednost za trajanje pokazuje najdulji segment vokala unutar jednog šireg intervala koji obuhvaća i riječ koju analiziramo.

6.4.1 Muški glasovi

U tablici 14 (stranica 153) za riječ „pozovem“ naznačeno je da je pojačan intenzitet. Uzevši u obzir maksimalni intenzitet za ostale riječi vidljivo je da je ovaj porast intenziteta mali. Sa snimke je također teško utvrditi je li ova riječ naglašena. Za riječ „sve“ naznačeno je povišenje tona, što se vidi i na slici 94 (stranica 147). Na snimci se čuje da je ova riječ naglašena. Još jedna riječ označena pojačanim tonom je „podrže“. Međutim, ovdje je odstupanje poravnanja prilično značajno, pa se vjerojatno radi o riječi „i“ koja slijedi i koja je na snimci jasno

naglašena. Riječ „uplatama“ je označena kao produljena, što odgovara snimci. Na slici 95 (stranica 147) vidi se da se na vremenskoj poziciji s početkom 15.16 nalazi niz vokala koji je znatno duži od okolnih takvih nizova pa je zbog toga na tom mjestu ustanovljen produženi izgovor.

U tablici 15 (stranica 154) riječ „zagrebu“ označena je kao naglašena zbog povišenog tona, što odgovara snimci. Na slici 96 (stranica 148) vidi se povišenje tona na segmentu ove riječi. Sljedeća označena riječ je „sportske“ kod koje je naznačeno produljenje izgovora. U ovom slučaju, međutim, odstupanje na kraju riječi je oko tri glasa, tako da navedeni segment ove riječi obuhvaća i početak iduće. Na ovoj snimci su, u stvari, produljene obje riječi, „sportske“ i „zvijezde“, što se vidi na slici 97 (stranica 148), ali je zbog odstupanja označena samo prva.

U tablici 16 (stranica 155) nalazi se popis riječi sa snimke govora koji je uglavnom jednoličan jer je govornik čitao pripremljen tekst. Ovdje je za riječ „odgovorit“ naznačen povišen ton, što se vidi na slici 98 (stranica 149)i što odgovara govoru na snimci. Za riječ „i“ je također naznačen povišen ton, iako se slušanjem ova riječ više ističe u kombinaciji s riječju „čak“, nego izolirano. Za riječ „drugim“ također je naznačen povišen ton, što odgovara govoru na snimci kao što se vidi na slici 99 (stranica 149). Jedan problem kod ovog primjera je riječ „važnijih“. Ova je riječ na snimci izgovorena naglašeno, što se vidi i na slici 100 (stranica 150). U ovom slučaju vrh intonacije nalazi se otprilike na 7.88-7.89 sekundi, međutim taj interval nije uzet u obzir jer je poravnanje odredilo završetak prethodne riječi, “tih”, na 7.87, a početak riječi “važnijih” na 7.9, pa je taj vrh preskočen u detekciji. Ovdje je vidljivo da je za riječ “važnijih” maksimalni ton dostigao 134.75 Hz, dok je onaj za prethodnu riječ dostigao 124.07 Hz, a onaj za sljedeću 114.2 Hz. To znači da je povišenje tona za riječ “važnijih” i dalje prisutan, ali ovdje nije označen jer je razlika u vrijednostima manja od 20%. Ako se prag postavi na, recimo, 8% onda je ova riječ označena kao naglašena. Međutim, u tom slučaju bi i neke druge riječi bile označene kao naglašene jer bi se tada uočavale manje razlike u povišenju tona.

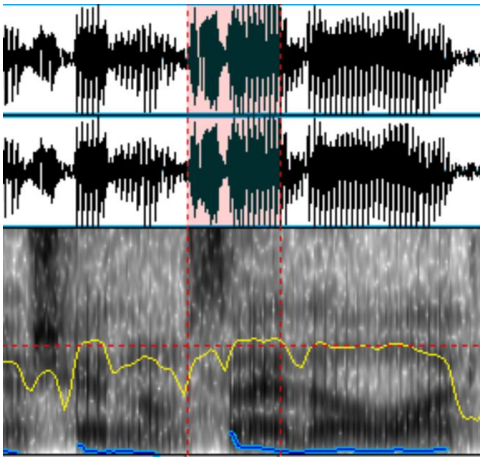
6.4.2 Ženski glasovi

Govor na snimci čiji su rezultati prikazani u tablici 17 (stranica 156) jednoličan je i nema znatnih odstupanja intenziteta i tona. Jedina riječ koja je označena kao naglašena, i to kroz trajanje, je “odnosno”, što odgovara izgovoru na snimci. S obzirom da su u govoru riječi “zamijeniti odnosno” izgovorene povezano, ovdje se kraj riječi “zamijeniti” i početak riječi “odnosno” preklapaju, tako da je glas “i” od “zamijeniti” i početni glas “o” od “odnosno”

spojeni u jedan niz vokala. Međutim, čak i ako odvojimo ta dva niza vokala “i” i “o” na slici 101 vidi se da je glas “o” izgovoren produženo.

U tablici 18 (stranica 157) naznačena je riječ “i” kao naglašena kroz trajanje. Na slici 102 (stranica 151) vidi se produljeni niz vokala koji vjerojatno jednim dijelom obuhvaća i riječ “u” jer su ove dvije riječi izgovorene spojeno (kao “iu”). Nadalje, s obzirom da riječ “utorak” počinje glasom “u” ovdje imamo tri vokala za redom. Ovdje na snimci nije sasvim jasno je li „i“ naglašen ili nije. Još jedna riječ označena kao naglašena povišenim tonom je „češće“, što odgovara govoru na snimci i što se vidi na slici 103 (stranica 151).

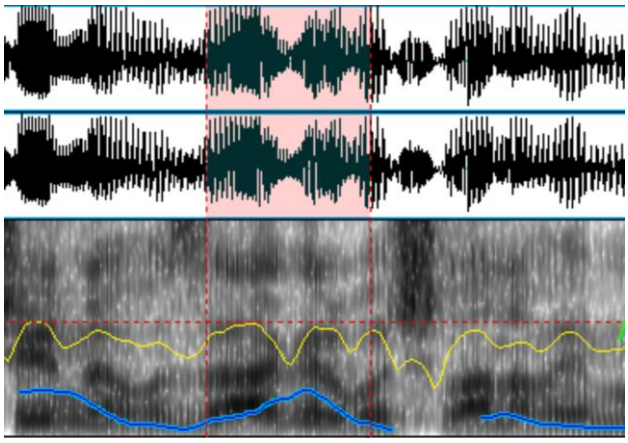
U tablici 19 (stranica 158) označena je riječ „djelovanje“ kao naglašena trajanjem. Međutim, ovdje je problem u tome što glas „l“ nije prepoznat, pa su se spojili vokali „e“ i „o“, čime se dobio produljeni niz vokala, kao što se vidi na slici 104 (stranica 152). Na toj snimci primjećuje se i da je naglašena riječ „suprotno“, ali ne kroz intenzitet, intonaciju ili trajanje, nego tako što se prije te riječi nalazi kratka pauza, što ovaj sustav trenutno ne uzima u obzir.



Slika 94: Označeni segment za riječ „sve“ gdje se vidi povišenje tona.

```
72:     14.78     14.81     36.6      ouooooooo
73:     14.85     14.87     22.0      *vv
74:     14.88     14.95     71.6      oaaaaaaaaaaaeae
75:     14.96     14.98     22.5      eeeee
76:     14.99     15.05     59.6      eaeaaeaaaaoa
77:     15.06     15.15     81.9      aaaaaaoaooaooouoo
78:     15.16     15.34     183.6     ueeeeieieiieieieieeiiiiiiiiiiiiieei
79:     15.35     15.37     12.7      uuuu
80:     15.38     15.43     56.3      uuuuuuuuuuo
81:     15.45     15.5      44.9      *mvrm
82:     15.54     15.6      50.9      ooooooaaaaa
83:     15.62     15.63     20.0      aaa
84:     15.73     15.75     14.8      aaaa
85:     15.76     15.79     32.4      aaoaooo
86:     15.8      15.8      0.0       *n
```

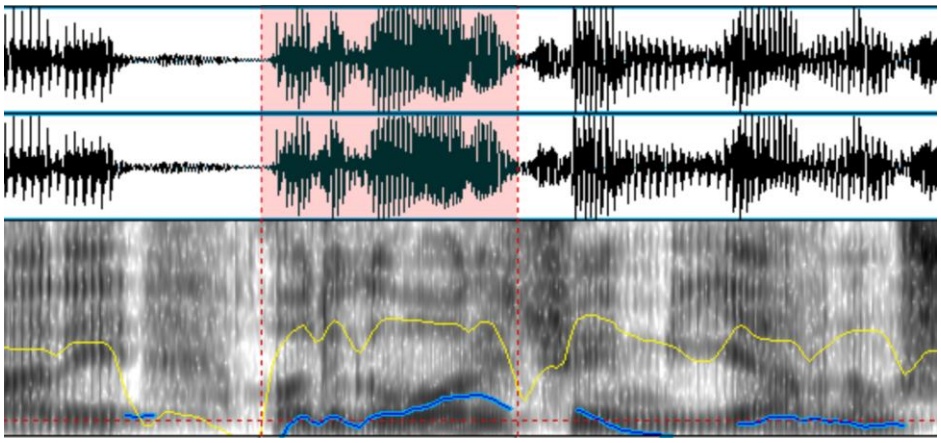
Slika 95: Ispis glasova koji obuhvaća riječ „uplatama“ i jedan dio okolnih riječi.



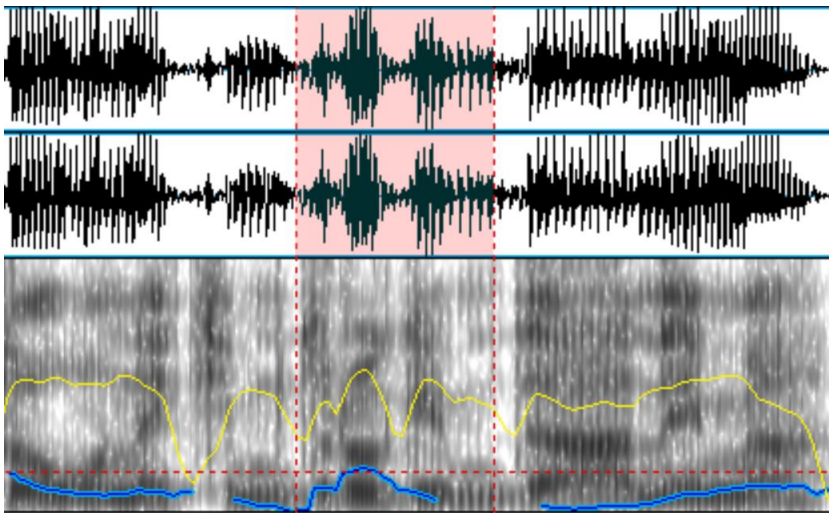
Slika 96: Označeni segment za riječ „zagrebu“ gdje se vidi povišenje tona.

82:	26.01	26.01	0.0	*v
83:	26.03	26.06	21.8	aaeee
84:	26.08	26.09	11.9	eiie
85:	26.1	26.11	11.8	*gr
86:	26.18	26.29	105.4	eaaaaaaaaoaoooooaaaa
87:	26.29	26.35	60.0	zzssssss
88:	**26.45**	**26.55**	**100.0**	**oooaooooooooooaaaaaaaaaa**
89:	26.55	26.56	9.0	*lr
90:	26.59	26.63	40.0	zssss
91:	26.72	26.76	38.6	ieieeeeee
92:	26.78	26.78	0.0	*d
93:	26.78	26.83	50.0	zzzzzzzzzz
94:	**26.83**	**27.0**	**170.8**	**eieieieeeeieieeeieieeeeeeaaeaeaaeaaaaa**
95:	27.02	27.05	27.7	zzzzzzz
96:	27.06	27.08	26.5	*vrr
97:	27.09	27.15	54.4	aaaaeaeaeaaa
98:	27.16	27.16	0.0	*l
99:	27.16	27.17	10.0	eee
100:	27.18	27.19	11.1	*nn
101:	27.23	27.25	20.0	uea
102:	27.33	27.38	50.0	aeiaaa
103:	27.41	27.44	30.0	aaea

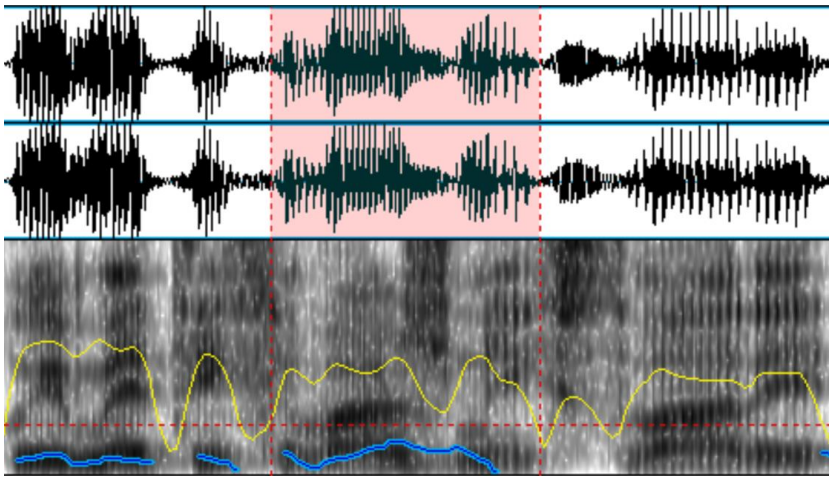
Slika 97: Ispis glasova koji obuhvaća riječi „sportske“ i „zvijezde“ i jedan dio okolnih riječi.



Slika 98: Označeni segment za riječ „odgovorit“ gdje se vidi povišenje tona.



Slika 99: Označeni segment za riječ „drugim“ gdje se vidi povišenje tona.



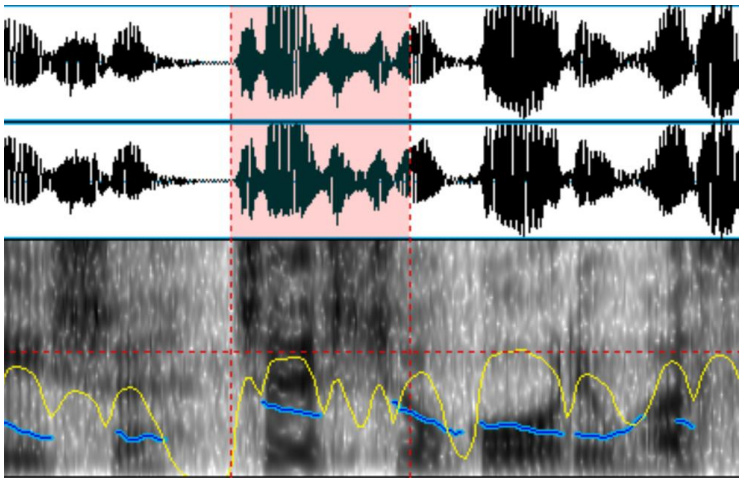
Slika 100: Označeni segment za riječ „važnijih“ gdje se vidi povišenje tona.

```
94:     11.76     11.76     0.0       *k
95:     11.76     11.78     18.3      eeeeiee
96:     11.79     11.81     22.3      *vvvv
97:     11.82     11.86     39.0      eeeieiiiiiiiii
98:     11.86     11.89     27.8      iiiiiiuii
99:     11.89     11.97     80.4      *nm
100:    11.97     12.26     289.9
iieiiiiiiiiiiiiiiiiieeeeieiiaeeioouooooooooooooooooooooooooooooooooaooo
oo
101:    12.27     12.27     0.0       *r
102:    12.27     12.28     10.0      ouuu
103:    12.28     12.29     5.1       *mn
104:    12.29     12.31     24.6      uuuuuuuu
105:    12.32     12.32     0.0       *n
106:    12.32     12.46     139.4
ooooooooooooooooooooooooooooooooooooaoaoooo
107:    12.47     12.47     0.0       *v
108:    12.47     12.55     80.0      zzscsscsss
109:    12.58     12.61     26.1      *vgvvvrr
110:    12.62     12.64     14.0      aaooo
111:    12.64     12.64     0.0       *p
112:    12.64     12.7      60.0      aaaaaaoaauuooouooo
113:    12.7      12.7      0.0       *r
114:    12.71     12.71     5.2       ouu
115:    12.72     12.72     5.4       *mm
```

Slika 101: Ispis glasova koji obuhvaća riječ „odnosno“ i jedan dio okolnih riječi.

```
0:      0.36       0.53       167.0
iiiieeieieeeeeeeeeaeeaaaaeaaaaoaooooooooooooooooooooooao
1:      0.53       0.58       48.8       aaaaaaaaaouiiui
2:      0.58       0.6        17.7       *nnnn
3:      0.63       0.68       54.1       aeeeeeeeeeeeeeeeeeea
4:      0.69       0.69       0.0        *n
5:      0.69       0.8        110.0      eeeeeeaeeeeeeaeeaeaaaaaaaeaieeeeeieee
6:      0.8        0.82       17.5       *kkk
7:      0.82       0.84       20.0       eeieeae
8:      0.84       0.84       0.0        *k
9:      0.9        0.92       21.1       iiieeeae
10:     0.93       0.93       0.0        *r
11:     0.93       0.94       5.5        oou
12:     0.94       0.97       23.8       *nnnnl
```

Slika 102: Ispis glasova koji obuhvaća riječ „i“ i jedan dio okolnih riječi.



Slika 103: Označeni segment za riječ „češće“ gdje se vidi povišenje tona.

```
82:     8.35       8.4        54.3       aaaeaeaeeeeeeee
83:     8.41       8.46       44.0       *vndnvnv
84:     8.46       8.7        234.1
iiiiiiiiiiiiiieieeieaeeeeeaeeaaaoaooooooooooooooooooooooooooooooouoo
85:     8.71       8.72       16.1       *nnn
86:     8.73       8.82       89.3       oaaaaaaaaaaaaaaaeeeeeeeiee
87:     8.83       8.83       0.0        *v
88:     8.84       8.86       23.7       ooooouoo
89:     8.87       8.88       18.1       *vvv
90:     8.89       9.0        110.1      iiieiiiieiieeuieeeieeeeeeeei
```

Slika 104: Ispis glasova koji obuhvaća riječ „djelovanje“.

Tablica 14: Detekcija naglašenih riječi za snimku 1 (muški glas).

RIJEČ	ODSTUPANJE-OD	ODSTUPANJE-DO	INTENZITET	TON	TRAJANJE	ISTAKNUTO
a	15	-58.8	69.93/70.21/70.07	76.09/76.13/76.11	4	
meni	-25.6	-17.8	72.65/75.75/74.42	92.04/108.31/100.81	12	
ostaje	-7.2	-37.4	55.49/76.89/71.23	87.68/107.73/97.91	17	
da	-0.8	-89.1	66.21/72.72/68.37	78.77/86.93/81.12	4	
pozovem	-198.1	65	53.25/77.0/70.14	78.79/96.64/85.64	24	I?
još	76.7	24.6	59.78/71.8/66.79	76.44/76.44/76.44	3	
jednom	15	5	64.16/75.35/71.11	75.96/82.04/79.0	7	
sve	15	-18.3	67.9/75.53/73.86	81.99/115.8/91.19	17	T+
da	-35.9	60.2	69.06/74.63/73.07	76.37/83.12/80.64	19	
ovu	121.4	485	49.18/76.09/67.65	77.73/89.58/82.22	22	
akciju	545.6	473.5	67.26/76.95/71.99	79.27/91.86/86.77	10	
podrže	384.9	483.1	41.85/78.16/62.38	78.84/159.88/114.12	14	T>
i	135	48.5	58.96/59.51/59.27	130.3/131.79/130.85	4	
pozivima	63.6	-34.6	66.28/76.21/72.83	83.41/127.96/97.37	17	
na	-0.3	-8.8	70.78/73.97/72.42	80.35/84.44/82.3	13	
broj	4.2	-81.3	67.17/75.03/71.83	74.95/92.75/83.68	8	
ali	-85	-198.6	73.84/76.6/75.24	82.48/91.3/87.34	15	
i	-196.8	-184.9	68.64/73.71/71.88	82.55/87.43/85.51	17	
uplatama	-192.8	-175	51.07/76.84/69.39	76.19/95.02/87.3	35	R+
na	-165.8	-245	62.4/69.92/66.36	78.75/84.2/82.36	4	
taj	-268	-287.1	59.93/71.16/66.71	75.14/86.8/79.8	4	
račun	-265	6.4	48.11/75.56/68.2	75.31/130.63/94.21	21	

Tablica 15: Detekcija naglašenih riječi za snimku 2 (muški glas).

RIJEČ	ODSTUPANJE-OD	ODSTUPANJE-DO	INTENZITET	TON	TRAJANJE	ISTAKNUTO
istodobno	0	215.4	63.46/75.51/72.33	113.26/146.28/134.91	16	
su	222.3	166	54.52/74.97/70.67	117.87/123.19/121.25	25	
nakon	194.1	131.3	48.84/76.29/69.28	77.31/133.9/116.07	22	
oproštaja	90	28.6	45.98/74.96/66.63	94.08/131.64/112.5	21	
od	39.2	-0.1	63.21/65.96/64.58	93.55/95.63/94.59	0	
saborske	100	0	46.32/76.72/68.89	90.15/144.49/122.09	32	
većine	0	60.6	50.51/76.39/70.52	84.28/166.52/125.79	30	
u	80	55	68.14/69.37/68.53	88.31/93.16/89.85	0	
zagrebu	37.2	-40	66.51/75.98/72.85	99.35/165.24/130.85	27	T+
članovi	-16.3	59	61.12/74.35/70.74	88.33/116.71/99.69	17	
mosta	0	182.1	51.95/75.68/70.64	83.98/139.91/108.99	21	
u	230	226.8	55.27/74.24/70.63	127.49/144.83/136.94	11	
metkoviću	235	67.5	57.66/74.91/69.58	86.56/126.38/104.91	10	
dočekani	41.2	-23.5	46.62/71.86/65.57	82.48/93.84/89.5	11	
kao	-71.6	0	50.16/74.49/66.51	80.53/97.28/88.0	19	
sportske	0	229	45.7/75.59/69.44	89.63/133.26/115.67	38	R+>
zvijezde	220	0	45.91/75.25/61.11	87.44/115.6/109.2	12	

Tablica 16: Detekcija naglašenih riječi za snimku 3 (muški glas).

RIJEČ	ODSTUPANJE-OD	ODSTUPANJE-DO	INTENZITET	TON	TRAJANJE	ISTAKNUTO
u	20	52.6	71.66/78.23/76.88	102.5/110.89/107.56	16	
ime	72.4	44.4	57.88/78.58/74.01	93.24/103.84/99.28	14	
tih	40	153.7	57.52/75.85/70.52	88.0/124.07/104.2	25	
važnijih	176.6	330	56.08/75.19/68.24	76.81/134.75/114.64	18	-
stvari	320	323.1	41.36/75.41/61.02	75.17/114.2/108.15	24	
odgovorit	83.3	-36.7	69.08/77.24/75.29	91.14/155.23/127.09	23	T+
ću	-0.4	-110.4	-1	-1	0	
im	-110	-180	58.51/66.58/63.56	129.31/129.31/129.31	0	
javnom	-160	4.6	67.17/77.94/72.65	75.17/124.43/98.51	29	
šutnjom	10	70	51.74/75.41/68.3	78.38/107.23/98.79	16	
čak	10	-65.6	62.35/74.32/71.44	109.31/127.5/119.96	12	
i	0	-3.7	73.63/77.4/76.42	125.2/154.36/137.52	9	T+<
na	5.1	13.9	71.52/78.01/74.7	103.85/116.76/108.65	18	
ono	23.3	30	61.14/74.44/72.86	97.91/122.36/114.81	26	
na	16.2	7.5	71.39/75.02/73.57	108.09/120.03/115.83	20	
što	10	-52.7	46.64/76.01/66.71	108.3/137.85/129.92	3	
u	-36.3	-76.3	-1	-1	0	
nekim	-70	-235.1	74.75/76.76/75.7	100.98/112.76/105.8	7	
drugim	-230	-165.7	55.39/78.05/70.21	75.04/147.13/109.97	10	T+
danima	-160	-44.6	64.91/76.63/72.77	79.0/118.34/98.84	13	

Tablica 17: Detekcija naglašenih riječi za snimku 4 (ženski glas).

RIJEČ	ODSTUPANJE-OD	ODSTUPANJE-DO	INTENZITET	TON	TRAJANJE	ISTAKNUTO
odbor	0	5.6	62.98/75.97/71.06	218.07/246.7/231.95	56	
za	10	-40.6	68.7/74.94/71.71	207.57/215.85/213.75	8	
ustav	-7.7	89.3	42.0/75.56/67.36	188.1/232.49/214.23	36	
poslovnik	20.2	-3.7	59.03/77.33/70.43	204.13/221.59/211.03	27	
i	-78.3	-110	53.98/64.51/59.17	192.06/207.9/198.85	6	
politički	-188.1	-8.1	42.46/74.53/66.89	177.31/216.17/197.65	55	
sustav	9.1	-53.4	58.41/76.4/68.78	171.41/241.7/220.89	18	
treba	-144.6	-3.7	42.25/76.1/68.29	194.49/228.96/211.55	21	
reći	53.4	30	46.96/75.83/66.78	186.03/223.02/214.98	21	
tko	119.2	110	62.95/76.44/71.13	210.1/238.5/220.37	24	
ga	118.5	94.8	65.78/73.71/68.78	208.27/210.9/209.59	19	
može	90	-30.6	66.69/75.14/70.67	190.16/211.25/201.4	10	
zamijeniti	-3.9	-160	58.43/74.33/69.74	173.61/231.51/201.83	16	
odnosno	-145.8	-20	55.12/76.65/70.17	171.57/234.53/211.51	77	R+
dati	-14.8	5.6	47.59/77.65/70.82	187.69/241.86/225.34	43	
mišljenje	15.1	77.9	58.84/72.46/69.08	174.21/240.85/204.42	45	
o	83.7	20.3	57.86/60.41/59.14	169.24/170.62/169.93	3	
tome	-73	-60.8	42.77/76.13/68.29	173.31/239.91/225.39	26	

Tablica 18: Detekcija naglašenih riječi za snimku 5 (ženski glas).

RIJEČ	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**	**INTENZITET**	**TON**	**TRAJANJE**	**ISTAKNUTO**
i	9.5	136.5	74.63/78.54/77.32	219.13/247.15/233.79	57	R?
u	131	149.8	63.47/77.74/73.0	223.39/226.36/225.06	15	
utorak	198.5	1.9	50.55/75.82/67.24	223.52/243.59/230.06	37	
će	85.9	-4.1	-1	-1	0	
mjestimice	0	-3.2	61.3/74.24/71.07	196.6/213.74/207.93	32	
biti	-42.8	-50	53.2/76.08/67.12	169.82/201.23/192.57	18	
kiše	56.8	-60.6	60.97/73.33/68.61	140.35/173.58/154.02	19	
češće	-213.5	50	41.63/75.14/66.36	135.7/209.75/181.45	37	T+
u	140	108	71.68/75.45/74.14	164.49/169.44/166.13	8	
gorju	140.3	145	60.88/76.42/69.28	144.89/180.39/157.95	17	
a	101.8	11.8	-1	-1	0	
s	20	10	68.45/75.7/72.51	200.94/200.94/200.94	0	
obzirom	63	-81.5	55.15/73.79/67.77	166.92/191.35/179.59	17	
na	-63.3	-102.8	63.23/64.92/63.92	162.41/165.42/164.31	6	
to	-186.6	-255.1	63.35/69.48/66.47	149.0/159.72/153.27	19	
da	-268.4	-244.9	40.53/73.3/57.1	149.16/197.42/176.05	7	
će	-120	-157.7	67.15/74.54/72.75	170.43/187.51/179.43	27	
do	-170	-107.4	59.74/70.98/66.83	136.48/166.93/152.33	14	
nas	15.6	-10	64.47/74.43/70.37	151.15/165.4/159.99	28	
stići	150	220.8	50.55/74.68/65.27	161.06/170.08/163.82	19	
hladniji	207	10.1	57.6/74.44/69.55	158.33/199.36/180.01	22	
zrak	10.6	-54.3	65.84/73.79/70.64	187.31/216.31/208.28	24	

Tablica 19: Detekcija naglašenih riječi za snimku 6 (ženski glas).

RIJEČ	**ODSTUPANJE-OD**	**ODSTUPANJE-DO**	**INTENZITET**	**TON**	**TRAJANJE**	**ISTAKNUTO**
polusatnom	30	247.2	49.53/77.32/71.63	150.0/205.82/177.55	55	
obraćanju	320	460	58.03/76.57/70.46	140.45/202.64/169.31	76	
javnosti	481.9	475	48.56/75.03/69.66	122.08/174.37/145.71	37	
u	471.9	464	70.81/72.53/72.09	143.73/145.69/145.11	16	
kojem	510	430	64.3/77.89/74.47	155.5/171.96/166.01	22	
je	447.8	349.5	70.03/74.36/73.05	153.25/174.44/164.45	10	
predsjednika	276.6	379.9	53.64/77.55/69.7	129.58/177.29/148.69	24	
hadezea	320	192.5	55.64/76.94/71.37	95.74/181.31/159.96	33	
andreja	188.8	360	43.67/75.13/70.23	118.17/231.33/176.1	31	
plenkovića	293.7	597.1	36.75/76.09/65.1	141.51/222.45/189.69	46	
prozvao	185.4	-6.7	66.3/76.26/72.58	148.93/215.83/192.09	44	
za	20	-64.4	68.34/69.4/68.77	158.31/158.33/158.32	0	
djelovanje	-110	-10	59.15/76.52/71.92	133.24/215.59/177.0	70	R?
suprotno	10	8	49.24/75.89/69.73	146.57/231.87/200.25	14	-
stranačkim	10	-38.3	55.36/73.53/67.78	131.16/156.59/145.37	21	
interesima	-41	-230	49.38/75.92/66.9	122.18/138.48/132.11	6	

6.5 Poravnavanje podnatpisa s govorom

U ovom dijelu prikazani su rezultati poravnavanja cijelih podnatpisa s govorom prema metodi opisanoj u dijelu 5.4.3. Pomaci su 0 sekundi (to jest, nema pomaka), +1 sekunda (pomak od jedne sekunde prema kraju snimke) ili -1 sekunda (pomak od jedne sekunde prema početku). Mjesta na kojima se nalazi '?' označavaju da za taj segment zvuka nije bilo dovoljno glasova da se napravi poravnanje (jer je taj dio izgovoren na početku ili na kraju snimke).

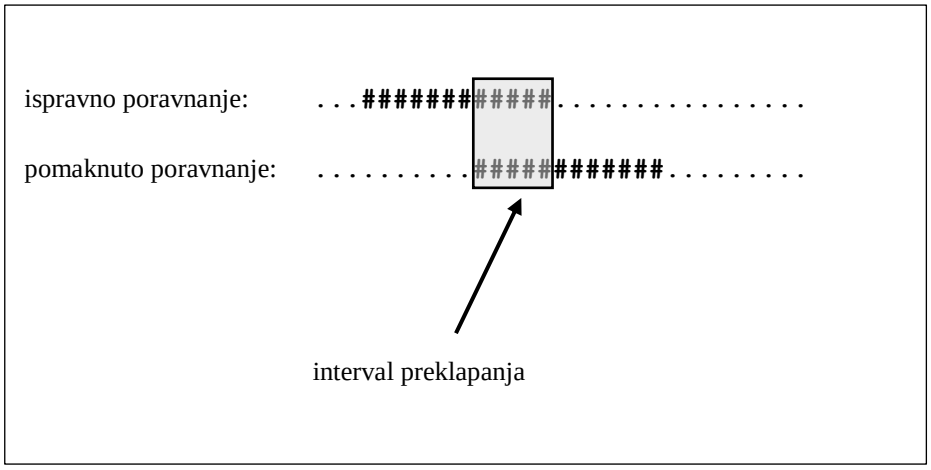
U tablicama 20, 21, 22, 23, 24 i 25 prikazani su rezultati poravnanja cijelih podnatpisa u odnosu na pomak podnatpisa i broj bodova koji je takvom poravnanju pridružio algoritam za poravnanje (5.4.2). Na kraju svake tablice pokazani su rezultati poravnanja teksta podnatpisa bez pomaka i poravnanje s prvim pomakom kojem je dodijeljeno najviše bodova. Na osnovu ovih rezultata vidi se sljedeće:

- Broj dodijeljenih bodova za neke pomake veći je od onog kod kojeg nema pomaka (gdje je početak i kraj zvuka postavljen točno u vrijeme kada se podnatpis pojavljuje odnosno nestaje). To znači da je veći broj relevantnih glasova odgovarao tekstu iako je takvo poravnanje za pojedinačne riječi rezultiralo većim odstupanjima. To se vidi, primjerice, u tablici 20 gdje je poravnanje s pomakom teksta od jednu sekundu prije početka govora dalo 43.3 boda, dok je poravnanje s tekstom bez pomaka dalo 42.9 boda.
- U nekim slučajevima točnost poravnanja nije se promijenila, bez obzira na pomak, što se vidi u tablicama 22 i 24.
- U nekim slučajevima pomak teksta u odnosu na govor rezultirao je točnijim poravnanjem, što se vidi u tablici 25, gdje je prosječno odstupanje palo sa 232.32, 269.54 milisekundi ili 3.1, 3.5 glasa na 164.25, 107.76 milisekundi ili 2.2, 1.4 glasa.
- Više različitih pomaka teksta može dati isti broj bodova. Ovdje je uzet prvi pomak koji daje najveći broj bodova.

U ovim primjerima iako su odstupanja podnatpisa od idealne pozicije imala djelomično negativan utjecaj na točnost poravnanja (osim za jedan uzorak), to nije osobito utjecalo na točnost detekcije naglašenih riječi. Razlog tome je taj da su u većini slučajeva na ovim uzorcima pojačani intenzitet, povišeni ton ili produženi izgovor bili ustanovljeni negdje unutar intervala izgovorene riječi, čak i ako njen početak i kraj nisu bili točno utvrđeni. Na slici 105 (stranica 160) ilustriran je jedan takav slučaj u kojem se ispravan interval riječi i onaj koji je pomaknut udesno jednim dijelom preklapaju. Ako je u tom dijelu, primjerice, povišen ton to

će biti ustanovljeno za odgovarajuću riječ iako ona nije točno poravnata sa zvukom. Razlika bi bila u tome da je u slučaju ispravnog poravnanja ton povišen na kraju riječi, dok bi u slučaju pomaknutog poravnanja to povišenje tona bilo utvrđeno na početku riječi. U oba slučaja gdje je utvrđen povišen ton nije bitno ako se to povišenje nalazi unutar intervala riječi.

Slika 105: Preklapanje intervala jedne riječi s različitim poravnanjem.



U tablicama 26, 27, 28, 29 i 30 (od stranice 163) i prikazane su usporedbe detekcije naglašenosti bez pomaka teksta i sa pomakom, gdje se vidi da razlike postoje u samo dva slučaja, što znači da odstupanja u poravnanju nisu znatno utjecala na detekciju naglašenosti.

Tablica 20: Poravnanje cijelog podnatpisa snimke 1 (muški glas) gdje je broj bodova bez pomaka 42.9.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	?
2	0	1	42.9
3	**-1**	**0**	**43.3**
4	-1	-1	35.3
5	-1	1	43.3
6	1	0	?
7	1	-1	?
8	1	1	29.9
Prosječno odstupanje bez pomaka (ms/br.glasova): (128.08, 140.36)/(1.7, 1.8)			
Prosječno odstupanje s pomakom pod #3 (ms/glasova): (258.89, 251.87)/(3.4, 3.3)			

Tablica 21: Poravnanje cijelog podnatpisa snimke 2 (muški glas) gdje je broj bodova bez pomaka 59.1.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	?
2	**0**	**1**	**62.6**
3	-1	0	59.1
4	-1	-1	?
5	-1	1	62.6
6	1	0	?
7	1	-1	?
8	1	1	45.6
Prosječno odstupanje bez pomaka (ms/br.glasova): (92.76, 87.35)/(1.2, 1.1)			
Prosječno odstupanje s pomakom pod #2 (ms/glasova): (92.76, 110.39)/(1.2, 1.5)			

Tablica 22: Poravnanje cijelog podnatpisa snimke 3 (muški glas) gdje je broj bodova bez pomaka 39.3.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	?
2	**0**	**1**	**39.3**
3	-1	0	39.3
4	-1	-1	?
5	-1	1	39.3
6	1	0	?
7	1	-1	?
8	1	1	?
Prosječno odstupanje bez pomaka (ms/br.glasova): (77.68, 100.03)/(1.0, 1.3)			
Prosječno odstupanje s pomakom pod #2 (ms/glasova): (77.68, 100.03)/(1.0, 1.3)			

Tablica 23: Poravnanje cijelog podnatpisa snimke 4 (ženski glas) gdje je broj bodova bez pomaka 66.2.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	59.0
2	0	1	66.2
3	**-1**	**0**	**66.5**
4	-1	-1	59.3
5	-1	1	66.5
6	1	0	48.7
7	1	-1	41.5
8	1	1	48.7
Prosječno odstupanje bez pomaka (ms/br.glasova): (65.3, 51.36)/(0.9, 0.7)			
Prosječno odstupanje s pomakom pod #3 (ms/glasova): (86.98, 60.34)/(1.1, 0.8)			

Tablica 24: Poravnanje cijelog podnatpisa snimke 5 (ženski glas) gdje je broj bodova bez pomaka 55.3.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	45.7
2	**0**	**1**	**55.3**
3	-1	0	55.3
4	-1	-1	45.7
5	-1	1	55.3
6	1	0	33.5
7	1	-1	?
8	1	1	37.4
Prosječno odstupanje bez pomaka (ms/br.glasova): (108.85, 89.8)/(1.4, 1.2)			
Prosječno odstupanje s pomakom pod #2 (ms/glasova): (108.85, 89.8)/(1.4, 1.2)			

Tablica 25: Poravnanje cijelog podnatpisa snimke 6 (ženski glas) gdje je broj bodova bez pomaka 48.4.

R.BR.	POMAK LIJEVO	POMAK DESNO	BODOVI
1	0	-1	?
2	0	1	48.4
3	**-1**	**0**	**70.9**
4	-1	-1	?
5	-1	1	70.9
6	1	0	?
7	1	-1	?
8	1	1	?
Prosječno odstupanje bez pomaka (ms/br.glasova): (232.32, 269.54)/(3.1, 3.5)			
Prosječno odstupanje s pomakom pod #3 (ms/glasova): (164.25, 107.76)/(2.2, 1.4)			

Tablica 26: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 1 (muški glas).

RIJEČ	BEZ POMAKA	SA POMAKOM
a		
meni		
ostaje		
da		
pozovem	I	
još		
jednom		
sve	T	T
da		
ovu		
akciju		
podrže	T	T
i		
pozivima		
na		
broj		
ali		
i		
uplatama	R	R
na		
taj		
račun		

Tablica 27: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 2 (muški glas).

RIJEČ	**ISTAKNUTO-1**	**ISTAKNUTO-2**
istodobno		
su		
nakon		
oproštaja		
od		
saborske		
većine		
u		
zagrebu	T	T
članovi		
mosta		
u		
metkoviću		
dočekani		
kao		
sportske	R	R
zvijezde		

Tablica 28: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 3 (muški glas).

RIJEČ	**ISTAKNUTO-1**	**ISTAKNUTO-2**
u		
ime		
tih		
važnijih		
stvari		
odgovorit	T	T
ću		
im		
javnom		
šutnjom		
čak		
i	T	T
na		
ono		
na		
što		
u		
nekim		
drugim	T	T
danima		

Tablica 29: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 4 (ženski glas).

RIJEČ	**ISTAKNUTO-1**	**ISTAKNUTO-2**
odbor		
za		
ustav		
poslovnik		
i		
politički		R
sustav		
treba		
reći		
tko		
ga		
može		
zamijeniti		
odnosno	R	R
dati		
mišljenje		
o		
tome		

Tablica 30: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 5 (ženski glas).

RIJEČ	**ISTAKNUTO-1**	**ISTAKNUTO-2**
i	R	R
u		
utorak		
će		
mjestimice		
biti		
kiše		
češće	T	T
u		
gorju		
a		
s		
obzirom		
na		
to		
da		
će		
do		
nas		
stići		
hladniji		
zrak		

Tablica 31: Usporedba detekcije naglašenosti bez i sa pomakom podnatpisa za snimku 6 (ženski glas).

RIJEČ	**ISTAKNUTO-1**	**ISTAKNUTO-2**
polusatnom		
obraćanju		
javnosti		
u		
kojem		
je		
predsjednika		
hadezea		
andreja		
plenkovića		
prozvao		
za		
djelovanje	R	R
suprotno		
stranačkim		
interesima		

6.6 Detekcija naglašenih riječi s podnatpisima emitiranim u teletekstu

U ovom dijelu prikazani su rezultati detekcije naglašenih riječi upotrebom podnatpisa emitiranih putem teleteksta, sa tri muška i četiri ženska glasa. U slučajevima u kojima se u podnatpisu nalazio broj ili akronim ti su dijelovi konvertirani u tekstualni oblik da bi se poklapali s izgovorenim. Riječi koje su bile detektirane kao naglašene podebljane su i potcrtane.

Za naglašenost postavljeni su sljedeći pragovi:

- odstupanje od prosječnog intenziteta više od 20%
- odstupanje od prosječne intonacije više od 25%
- odstupanje od prosječnog trajanja više od 60%

Rezultati su prikazani upotrebom jezika SSML u kojem su označeni prozodijski atributi na osnovu detekcije naglašenosti. Ispod SSMLa isti podaci prikazani su u tablici, s tim da je intenzitet u SSMLu prikazan kao broj decibela (koji treba biti u rasponu od 1 do 6), a u tablici kao postotak povišenja u odnosu na prosjek za taj dio zvuka.

6.6.1 Muški glas 1

Podnatpis:

00:07:39,440 --> 00:08:09,440

održan je referendum u Makedoniji na kojem su se građani

```
<speak><prosody pitch=+44.18%> održan </prosody> je referendum u makedoniji
na kojem su se građani </speak>
```

RIJEČ	ISTAKNUTO
održan	ton (44.18%)
je	
referendum	
u	
makedoniji	
na	
kojem	
su	
se	
građani	

00:07:42,680 --> 00:08:12,680

izjašnjavali o tome **jesu li za** prijedlog da se njihova

```
<speak>izjašnjavali o tome <prosody pitch=+42.84%> jesu </prosody> <prosody
pitch=+54.72%> li </prosody> <prosody pitch=+42.88%> za </prosody>
prijedlog da se njihova </speak>
```

RIJEČ	ISTAKNUTO
izjašnjavali	
o	
tome	
jesu	ton (42.84%)
li	ton (54.72%)
za	ton (42.88%)
prijedlog	
da	
se	
njihova	

00:07:45,440 --> 00:08:15,440

zemlja zove Republika Sjeverna Makedonija.

```
<speak>zemlja zove republika sjeverna makedonija </speak>
```

RIJEČ	ISTAKNUTO
zemlja	
zove	
republika	
sjeverna	
makedonija	

Ovdje su riječi „održan“, „jesu“ i „li“ ispravno detektirane kao naglašene. Riječ „za“ se nalazi na kraju naglašenog dijela, pa je zbog djelomičnog odstupanja u poravnanju i ona bila uključena u naglašeni dio.

00:07:48,440 --> 00:08:18,440

Iako je referendum **savjetodavni** mnogi su ga ocijenili

```
<speak><prosody pitch=+46.78%> iako </prosody> <prosody pitch=+27.27%> je
</prosody> referendum <prosody pitch=+36.11%> savjetodavni </prosody> mnogi
su ga ocijenili </speak>
```

RIJEČ	ISTAKNUTO
iako	ton (46.78%)
je	ton (27.27%)
referendum	
savjetodavni	ton (36.11%)
mnogi	
su	
ga	
ocijenili	

00:07:50,960 --> 00:08:20,960

kao **povijesnu** priliku da se prekine **višedesetljetni** spor

```
<speak>kao <prosody pitch=+50.86%> povijesnu </prosody> priliku da se
prekine <prosody pitch=+44.87%> višedesetljetni </prosody> spor </speak>
```

RIJEČ	ISTAKNUTO
kao	
povijesnu	ton (50.86%)
priliku	
da	
se	
prekine	
višedesetljetni	ton (44.87%)
spor	

00:07:54,200 --> 00:08:24,200

s **Grčkom** i **ubrzaju** euroatlantske integracije.

```
<speak>s <prosody volume=+1.21dB pitch=+38.28%> grčkom </prosody> <prosody
volume=+1.29dB > i </prosody> <prosody pitch=+34.16%> ubrzaju </prosody>
<prosody volume=+1.3dB > euroatlancke </prosody> integracije </speak>
```

RIJEČ	ISTAKNUTO
s	
grčkom	ton (38.28%), int (16.73%)
i	
ubrzaju	ton (34.16%)
euroatlantske	
integracije	

Ovdje su sve označene riječi ispravno detektirane kao naglašene. Za riječi „i“ i „euroatlantske“ naznačeno je da imaju lagano pojačan intenzitet, ali s obzirom da se radi o graničnom pojačanju i da za njih nije povišen ton one ovdje nisu označene kao naglašene.

6.6.2 Muški glas 2

Podnatpis:

00:36:07,680 --> 00:36:37,680

Zanimljiva energetska priča vezana je i uz nuklearnu elektranu

```
<speak><prosody pitch=+29.8%> zanimljiva </prosody> energecka priča vezana
je uz nuklearnu elektranu </speak>
```

RIJEČ	ISTAKNUTO
zanimljiva	ton (29.8%)
energetska	
priča	
vezana	
je	
uz	
nuklearnu	
elektranu	

00:36:10,680 --> 00:36:40,680

Zwentendorf. Austrijanci su je izgradili prije više od 40 **godina**

```
<speak>zventendorf austrijanci su je izgradili prije više od četrdeset
<prosody pitch=+38.47%> godina </prosody> </speak>
```

RIJEČ	ISTAKNUTO
zventendorf	
austrijanci	
su	
je	
izgradili	
prije	
više	
od	
četrdeset	
godina	ton (38.47%)

00:36:15,240 --> 00:36:45,240

no tadašnji kancelar Bruno Kreisky želio je potvrdu

```
<speak>no tadašnji kancelar bruno kreiski želio je potvrdu </speak>
```

RIJEČ	ISTAKNUTO
no	
tadašnji	
kancelar	
bruno	
kreiski	
želio	
je	
potvrdu	

00:36:18,640 --> 00:36:48,640

za uporabu nuklearne energije **dobiti** i na referendumu.

```
<speak>za uporabu nuklearne energije <prosody pitch=+147.17%> dobiti
</prosody> i na referendumu </speak>
```

RIJEČ	ISTAKNUTO
za	
uporabu	
nuklearne	
energije	
dobiti	ton (147.17%)
i	
na	
referendumu	

Ovdje je samo riječ „godina“ pogrešno detektirana kao naglašena zbog pomaka u poravnanju. Ostale riječi su ispravno detektirane kao naglašene.

6.6.3 Muški glas 3

Podnatpis:

00:40:20,040 --> 00:40:50,040

Veselog Tirolca **svi** **koji** vide srdačno i pozdrave.

```
<speak>veselog tirolca <prosody pitch=+32.9%> svi </prosody> <prosody
pitch=+34.76%> koji </prosody> vide srdačno i pozdrave </speak>
```

RIJEČ	ISTAKNUTO
veselog	
tirolca	
svi	ton (32.9%)
koji	ton (34.76%)
vide	
srdačno	
i	
pozdrave	

00:40:22,520 --> 00:40:52,520

Oduševljen **je** **Hrvatskom** i našim ljudima.

```
<speak>oduševljen <prosody pitch=+53.34%> je </prosody> <prosody
volume=+1.27dB pitch=+60.75%> hrvackom </prosody> i našim ljudima </speak>
```

RIJEČ	ISTAKNUTO
oduševljen	
je	ton (53.34%)
hrvatskom	ton (60.75%), int (17.45%)
i	
našim	
ljudima	

Ovdje su sve riječi osim „je“ ispravno detektirane kao naglašene. Riječ „je“ je zbog kratkoće bila u poravnanju malo pomaknuta u desno, prema riječi „hrvatskom“, pa je zbog toga obuhvatila dio povišenog tona i pojačanog intenziteta te riječi.

6.6.4 Ženski glas 1

Podnatpis:

00:07:27,320 --> 00:07:57,320

Kazna za ovakav čin primitivizma i huliganizma

```
<speak>kazna za ovakav čin primitivizma i huliganizma </speak>
```

RIJEČ	ISTAKNUTO
kazna	
za	
ovakav	
čin	
primitivizma	
i	
huliganizma	

00:07:30,320 --> 00:08:00,320

kojemu je posljedica teška tjelesna ozljeda je

```
<speak><prosody pitch=+48.91%> kojemu </prosody> je pos%edica teška
tjelesna oz%eda je </speak>
```

RIJEČ	ISTAKNUTO
kojemu	ton (48.91%)
je	
posljedica	
teška	
tjelesna	
ozljeda	
je	

00:07:32,840 --> 00:08:02,840

zatvor od šest mjeseci do pet godina.

```
<speak>zatvor od šest mjeseci do pet godina </speak>
```

RIJEČ	ISTAKNUTO
zatvor	
od	
šest	
mjeseci	
do	
pet	
godina	

Ovdje je riječ „kojemu“ ispravno detektirana kao naglašena. Iako slušanjem nije potpuno jasno je li ova riječ bila naglašena, riječ je dobro poravnata sa zvukom, a Praat na tom mjestu pokazuje skok u intonaciji koji se ispravno detektira kao naglašenost.

6.6.5 Ženski glas 2

Podnatpis:

00:10:13,080 --> 00:10:43,080

Time **niti** ćemo **zadržati** radnu snagu niti ćemo povećati plaće.

```
<speak><prosody pitch=+34.86%> time </prosody> <prosody pitch=+29.89%> niti
</prosody> ćemo <prosody pitch=+34.97%> zadržati </prosody> radnu snagu
niti ćemo povećati plaće </speak>
```

RIJEČ	ISTAKNUTO
time	ton (34.86%)
niti	ton (29.89%)
ćemo	
zadržati	ton (34.97%)
radnu	
snagu	
niti	
ćemo	
povećati	

Ovdje su sve riječi ispravno detektirane kao naglašene, iako je poravnanje bilo manje precizno zbog sporog izgovora i produljenih vokala.

6.6.6 Ženski glas 3

Podnatpis:

00:12:00,000 --> 00:12:30,000

istaknuo je da će odgovarati za **svoje** riječi i djela

```
<speak>istaknuo je da će odgovarati za <prosody pitch=+27.52%> svoje
</prosody> riječi i djela </speak>
```

RIJEČ	ISTAKNUTO
istaknuo	
je	
da	
će	
odgovarati	
za	
svoje	ton (27.52%)
riječi	
i	
djela	

Ovdje je riječ „svoje“ ispravno detektirana kao naglašena.

6.6.7 Ženski glas 4

Podnatpis:

00:11:42,360 --> 00:12:12,360

Nakon što su braniteljske udruge zatražile **tematsku** sjednicu Sabora

```
<speak><prosody pitch=+35.3%> nakon </prosody> što su braniteljske udruge
zatražile <prosody pitch=+28.69%> temacku </prosody> sjednicu sabora
</speak>
```

RIJEČ	ISTAKNUTO
nakon	ton (35.3%)
što	
su	
braniteljske	
udruge	
zatražile	
tematsku	ton (28.69%)
sjednicu	
sabora	

00:11:45,800 --> 00:12:15,800

o javnom djelovanju Vladina koalicijskog partnera i

```
<speak>o javnom djelovanju vladina koalicijskog partnera i </speak>
```

RIJEČ	ISTAKNUTO
o	
javnom	
djelovanju	
vladina	
koalicijskog	
partnera	
i	

00:11:48,600 --> 00:12:18,600

čelnika es de es esa Milorada **Pupovca**, a predsjednik Sabora

```
<speak>čelnika es de es esa milorada <prosody pitch=+25.96%> pupovca
</prosody> a precjednik sabora </speak>
```

RIJEČ	ISTAKNUTO
čelnika	
es	
de	
es	
esa	
milorada	
pupovca	ton (25.96%)
a	
pretsjednik	
sabora	

Ovdje su sve riječi ispravno detektirane kao naglašene, iako je riječ „Pupovca“ na granici zadanog praga naglašenosti, pa je slušanjem manje jasno je li bila naglašena ili nije.

7. Zaključak

Prozodijska obilježja jezika važan su element jezične komunikacije jer ona mijenjaju informativnost poruke i time doprinose njenom razumijevanju. S obzirom da se u tekstu prozodijska obilježja gube, cilj ovog rada bio je istražiti mogućnost vraćanja informacija o prozodiji, specifično o naglašenim riječima, iz govora u tekst. Za analizu naglašavanja uzeta su u obzir tri prozodijska obilježja:

1. Intonacija
2. Intenzitet
3. Tempo govora

Vraćanje ovih informacija u tekst dodaje informativnost zbog semantike izgovorenog teksta, dok istovremeno, s tehničke strane, takav tekst zahtijeva puno manje memorijskog kapaciteta od zvuka, pa u tom obliku može biti pogodan tamo gdje postoji potreba za spremanjem velikih količina podataka, kao što je arhiviranje. Isto tako, ovako obogaćeni tekst može biti koristan za osobe s oštećenim sluhom ili gluhonijeme osobe jer bi se njima na ovaj način olakšalo razumijevanje sadržaja time što bi im se približio izvorni oblik onoga što je i kako je bilo izgovoreno.

Glavna hipoteza od koje se krenulo u ovom radu bila je sljedeće:

- Informacija o naglašenim riječima iz jedne zvučne snimke govora s podnatpisima može se vratiti u tekst bez da se u potpunosti radi automatsko prepoznavanje govora.

Ova je hipoteza potvrđena time što je sustav za ukupno 115 riječi svih upotrebljenih uzoraka izdvojio 14 riječi kao naglašene, od kojih je 9 riječi bilo ispravno detektirano, dok u nekim slučajevima naglašenost nije bila jasno istaknuta (dvije riječi), a u tri riječi bila je pogrešno detektirana ili preskočena. Nadalje, u tri riječi naglašenost je uključivala prethodnu ili sljedeću riječ.

Iako je prepoznavanje govora bila važna komponenta ovog rada, ovdje se htjelo ustanoviti kako se prozodija može vratiti u tekst bez izgradnje kompletnog sustava za prepoznavanje govora (na hrvatskom jeziku) i upotrebom skromnih tehničkih resursa. Rezultati su pokazali da je uz vrlo mali uzorak govora moguće dobiti preciznost poravnanja s prosječnim odstupanjima

manjim od tri glasa (zavisno od kvalitete snimke). Iako je ovo istraživanje uključivalo samo govor na hrvatskom jeziku, ovakvi rezultati mogu biti relevantni i za neke druge jezike za koje možda ne postoje komercijalni sustavi za anotaciju govora SSMLom, a za koje bi se na isti način mogao implementirati ovakav sustav za detekciju naglašenih riječi.

S vrlo malo segmentiranog govora bilo je moguće dobiti dobre rezultate poravnanja individualnih riječi s govorom, pod uvjetom da se za zadani tekst otprilike odredi gdje se nalazi na snimci, što je slučaj kod snimki s podnatpisima. Od šest ovdje prikazanih primjera samo je na jednom prosječno odstupanje bilo nešto veće od 3 glasa, dok je na ostalima bilo manje od 2 glasa (ako se uzme da je prosječno trajanje glasa oko 76 milisekundi). To jednim dijelom zavisi i od tehnologije upotrebljene za prepoznavanje govora. Kao što su rezultati pokazali, ta odstupanja u poravnanju pojedinačnih riječi nisu znatno utjecala na rezultate analize naglašenosti jer je samo u manjem broju riječi detekcija naglašenosti odredila riječ prije ili poslije one koja je slušanjem bila utvrđena kao naglašena. Nadalje, to je bilo moguće primjenom vrlo bazičnih tehnologija koje nemaju velike zahtjeve za resursima, kao što je feed-forward neuralna mreža.

Originalni doprinosi ovog rada su

1. segmentirani govor na hrvatskom jeziku u trajanju od 150 sekundi na snimkama nekoliko muških govornika,
2. program za treniranje neuralne mreže s podacima generiranim programom Praat,
3. program za prepoznavanje govora upotrebom neuralne mreže,
4. algoritam za poravnanje teksta s klasificiranim glasovima, te njegova implementacija,
5. program za analizu naglašenosti pojedinačnih riječi.

Program za prepoznavanje govora upotrebom neuralne mreže može imati i primjenu u analizi varijacija u tempu govora, kako je pokazano u (Stojanović & Lazić, 2019) zbog toga što je brzina govora jedan aspekt naglašenosti riječi koji se ovdje analizira.

Algoritam za poravnavanje teksta s govorom nije upotrebljiv isključivo za tu svrhu jer njegova je općenita funkcija da za dva niza bilo kakvih elemenata (u ovom slučaju slova) nađe poravnanje jednog niza s nepotpunim ili „pogrešnim“ elementima s drugim nizom koji sadrži samo ispravne elemente. Takav bi se algoritam, primjerice, mogao upotrijebiti za pretraživanje teksta u kojem neka slova nedostaju ili su pogrešna (ako se, recimo, traži neko ime ili prezime

u dokumentu u kojem je to ime i/ili prezime možda pogrešno napisano). Nadalje, ovaj bi se algoritam mogao primijeniti i za implementaciju jednog oblika podudaranja uzoraka, gdje jedan niz sadrži podatke, a drugi predstavlja uzorak koji može sadržavati i varijable i kojim se deklarativno definira ono što se traži u nizu podataka.

Budući rad iz ovog područja biti će usmjeren ka sljedećem:

1. poboljšati detekciju tako da se uzimaju u obzir i kratke pauze prije riječi kojima se ta riječ može istaknuti,
2. razviti sustav za analizu emocija u govoru metodom sličnom ovoj, ali koja bi se više temeljila na specifičnim analizama zvuka pojedinačnih riječi i glasova i načinu izgovora,
3. poboljšanje algoritma za poravnanje teksta s govorom,
4. poravnavanje teksta s govorom upotrebom klasifikatora umjesto egzaktnog algoritma.

Iako isticanje riječi kako je navedeno pod 1) nije često kao isticanje (naglašavanje) intenzitetom, tonom ili trajanjem, ovakvo proširenje detekcije naglašenih riječi dalo bi još cjelovitiju sliku o izvornom sadržaju. Sustav za analizu emocija pod 2) vjerojatno se ne bi morao temeljit na analizi pojedinačnih riječi nego više na kretanju relevantnih zvučnih indicija kroz veće govorne cjeline, kao što su rečenice. Točke 3) i 4) odnosile bi se više na tehnička poboljšanja postojećeg sustava.

Literatura

1. Anguera, X., Luque, J., & Gracia, C. (2014). Audio-to-text alignment for speech recognition with very limited resources. *Interspeech 2014.*

2. Arons, B. (1994). Pitch-Based Emphasis Detection for Segmenting Speech Recordings. *International Conference on Spoken Language Processing*, (str. 1931 – 1934). Yokohama, Japan.

3. Atal, B. S., & Hanauer, S. L. (1971). Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. *J. Acoust. Soc. Am., 50*(2), 637-655.

4. Babić, S., Brozović, D., Moguš, M., Pavešić, S., Škarić, I., & Težak, S. (1991). *Povijesni pregled, glasovi i oblici hrvatskoga književnog jezika.* Zagreb: Hrvatska akademija znanosti i umjetnosti, Globus, Nakladni zavod. Dohvaćeno iz www.scribd.com

5. Bakran, J. (1996). *Zvučna slika hrvatskoga govora.* Zagreb: IBIS grafika.

6. Barić, E., Lončarić, M., Malić, D., Pavešić, S., Peti, M., & Zečević, V. (1995). *Hrvatska gramatika.* Zagreb: Školska knjiga.

7. Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities,, 3*, 1-8.

8. Bordel, G., Nieto, S., Penagarikano, M., Rodriguez-Fuentes, L. J., & Varona, A. (2012). A simple and efficient method to align very long speech signals to acoustically imperfect transcriptions. *Interspeech 2012.*

9. Brenier, J. M., Cer, D. M., & Jurafsky, D. (2005). The Detection of Emphatic Words Using Acoustic and Lexical Features. *Interspeech*, (str. 3297-3300). Lisbon, Portugal.

10. Caseiro, D., Meinedo, H., Serralheiro, A., Trancoso, I., & Neto, J. (2002). Spoken book alignment using WFSTs. *HLT '02 Proceedings of the second international conference on Human Language Technology Research, Pages 194-196.* San Diego, California.

11. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, 3rd ed.* The MIT Press.

12. Davis, K. H., Biddulph, R., & Balashek, S. (1952). Automatic Recognition of Spoken Digits. *J. Acoust. Soc. Am., Vol 24, No. 6,* 627-642.

13. Dhanashri, D., & Dhonde, S. (2015). Speech Recognition Using Neural Networks: A Review. *International Journal of Multidisciplinary Research and Development*, 226-229.

14. F. A. Everest, & K. C. Pohlmann. (2009). *Master Handbook of Acoustics, 5th ed.* McGraw-Hill.

15. Ferguson, J. D. (1980). *Hidden Markov Analysis: An Introduction.* Princeton, NJ: Institute for Defense Analyses.

16. Fletcher, H. (1922). The Nature of Speech and its Interpretations. *Bell Syst. Tech. J., Vol 1*, 129-144.

17. Forgie, J. W., & Forgie, C. D. (1959). Results Obtained from a Vowel Recognition Computer Program. *J. Acoust. Soc. Am., 31*(11), 1480-1489.

18. Fry, D. B., & Denes, P. (1959). The Design and Operation of the Mechanical Speech Recognizer at University College London. *J. British Inst. Radio Engr., 19*(4), 211-229.

19. Geron, A. (2017). *Hands-On Machine Learning with Scikit-Learn & TensorFlow.* O'Reilly Media.

20. Hansen, C. H. (2016). *Fundamentals of Acoustics.* Dohvaćeno iz http://www.who.int/.

21. Hardcastle, W., & Hewlett, N. (2006). *Coarticulation: Theory, Data, and Techniques.* Cambridge University Press.

22. Hawkins, S. (2016). Dohvaćeno iz http://www.ling.cam.ac.uk/li9/lab3_m08_speechandspectralanalysis.pdf

23. Hazen, T. J. (2006). Automatic alignment and error correction of human generated transcripts for long speech recordings. *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing.* Pittsburgh, PA, USA, September 17-21, 2006.

24. Heldner, M., Strangert, E., & Deschamps, T. (1999). A focus detector using overall intensity and high frequency emphasis. *ICPhS'99.*

25. Hillenbrand, J. M. (2016). Dohvaćeno iz http://homepages.wmich.edu/ ~hillenbr/501.html

26. Hoffmann, S., & Pfister, B. (2013). Text-to-Speech Alignment of Long Recordings Using Universal Phone Models. *14th Annual Conference of the International SpeechCommunication Association (INTERSPEECH), pp. 1520–1524, Lyon (2013).*

27. Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley.

28. Huang, C.-w. (2003). *Automatic closed caption alignment based on speech recognition transcripts.* Technical Report, University of Columbia, 2003.

29. Itakura, F. (1975). Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Trans. Acoustics, Speech and Signal Proc., ASSP-23*, 57-72.

30. Itakura, F., & Saito, S. (1970). A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies. *Electronics and Communications in Japan, 53A*, 36-43.

31. Jelinek, F. (1976). Continuous Speech Recognition by Statistical Methods. *Proc. IEEE, 64*, 532-536.

32. Jelinek, F., Bahl, L. R., & Mercer, R. L. (1975). Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech. *IEEE Trans. On Information Theory, IT-21*, 250-256.

33. Juang, B.-H., & Rabiner, L. R. (n.d.). *Automatic speech recognition – A brief history of the technology development.* Elsevier.

34. Klatt, D. H. (1977). Review of the DARPA Speech Understanding Project (1). *J. Acoust. Soc.Am., 62*, 1345-1366.

35. Kroul, M. (2009). Automatic Detection of Emphasized Words for Performance Enhancement of a Czech ASR System. *SPECOM'2009.* St. Petersburg.

36. Kuijk, D. B. (1999). Acoustic characteristics of lexical stress in continuous telephone speech. *Speech Communication 27*, 95-11.

37. Lee, K.-F. (1988). *Large-vocabulary speaker-independent continuous speech recognition: The Sphinx system, Ph.D. Thesis.* Carnegie Mellon University.

38. Levinson, S. E., Rabiner, L. R., & Sondhi, M. M. (1983). An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell Syst. Tech. J., 62*(4), 1035-1074.

39. Liporace, L. A. (1982). Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Trans. On Information Theory, IT-28*(5), 729-734.

40. Lippmann, R. P. (1990). Review of Neural Networks for Speech Recognition. *Readings in Speech Recognition, A. Waibel and K. F. Lee, Editors, Morgan Kaufmann Publishers*, 374-392.

41. Lowerre, B. (1990). The HARPY Speech Understanding System. U *Trends in Speech Recognition, W. Lea, Editor, Speech Science Publications* (str. 576-586). Morgan Kaufmann Publishers.

42. McCullough, W. S., & Pitts, W. H. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bull. Math Biophysics, 5*, 115-133.

43. Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* The MIT Press.

44. Moreno, P. J., & Alberti, C. (2009). A Factor Automaton Approach for the Forced Alignment of Long Speech Recordings. *ICASSP 2009.*

45. Moreno, P. J., Joerg, C., Van Thong, J.-M., & Glickman, O. (1998). A Recursive Algorithm for the Forced Alignment of Very Long Audio Segments. *In 5th Int. Conf. on Spoken Language Processing (1998).*

46. Murveit, H., Cohen, M., Price, P., Baldwin, G., Weintraub, M., & Bernstein, J. (1989). SRI's DECIPHER System. *Speech and Natural Language Workshop.* Philadelphia, PA.

47. Nagata, K., Kato, Y., & Chiba, S. (1963). Spoken Digit Recognizer for Japanese Language. *NEC Res. Develop.*(6).

48. Navarro, G. (2001). A Guided Tour to Approximate String Matching. *ACM Computing Surveys, 33*(1), 31-88.

49. Olson, H. F., & Belar, H. (1956). Phonetic Typewriter. *J. Acoust. Soc. Am., 28*(6), 1072-1081.

50. Poritz, A. (1982). Linear predictive hidden Markov models and the speech signal. *ICASSP-82*. Paris.

51. Rabiner, L. R., & Juang, B. H. (2004). *Statistical Methods for the Recognition and Understanding of Speech.* Encyclopedia of Language and Linguistics.

52. Rabiner, L. R., Levinson, S. E., Rosenberg, A. E., & Wilpon, J. G. (1979). Speaker Independent Recognition of Isolated Words Using Clustering Techniques. *IEEE Trans. Acoustics, Speech and Signal Proc., Assp-27*, 336-349.

53. Russel, S. J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

54. Sakai, J., & Doshita, S. (1962). The Phonetic Typewriter. *Information Processing 1962, Proc. IFIP Congress.* Munich.

55. Schwartz, R., Barry, C., Chow, Y.-L., Derr, A., Feng, M.-W., Kimball, O., . . . Vandegrift, J. (1989). The BBN BYBLOS Continuous Speech Recognition System. *Speech and Natural Language Workshop.* Philadelphia, PA.

56. Silipo, R., & Crestani, F. (2000). Prosodic stress and topic detection in spoken sentences. *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium*, (str. 243-252).

57. Slujter, A. M., & Heuven, V. J. (1996). Acoustic Correlates of Linguistic Stress and Accent in Dutch and American English. *ICSLP96*, (str. 630–633). Philadelphia, PA.

58. Stan, A., Bell, P., & King, S. (2012). A Grapheme-Based Method for Automatic Alignment of Speech And Text Data. *Spoken Language Technology Workshop (SLT), 2012 IEEE.*

59. Stojanović, A., & Lazić, N. (2019). A Method for Estimating Variations in Speech Tempo from Recorded Speech. *MIPRO*.

60. Suzuki, J., & Nakata, K. (1961). Recognition of Japanese Vowels—Preliminary to the Recognition of Speech. *J. Radio Res. Lab, 37*(8), 193-212.

61. Vermeulen, P., Bernard, E., Yan, Y., Fanty, M., & Cole, R. (1996). Comparison of HMM and Neural Network Approaches to Real World. *Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology*.

62. Wilpon, J. G., Rabiner, L. R., Lee, C. H., & Goldman, E. R. (1990). Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models. *IEEE Trans. On Acoustics, Speech and Signal Processing, 38*(11), 1870-1878.

63. Winston, P. H. (1993). *Artificial Intelligence, 3rd ed.* Addison-Wesley.

Popis slika

Popis tablica

Prilozi

U ovom dijelu nalaze se neki skriptovi upotrebljeni za ovo istraživanje. Ostali skriptovi nalaze se na CDu priloženom doktorskoj disertaciji.

Prilog 1: Praat skript 1 – za klasifikaciju po 10 ms

```
;;;
;;; NIJE POTREBNO UCITATI KOLEKCIJU U PRAAT!

;;; *** OBAVEZNI PARAMETRI ***
;;;
direktorij$ =
"C:\Users\Alex\OneDrive\FFZG\doktorat\podaci\za_detekciju\detekcija\test\"
ime_datoteke$ = "zvuk"  ; ovo treba biti .wav datoteka
pocetak = 0
kraj = 3.22

;---------------------------

korak = 0.010           ; 10 ms
ltas_raspon = 100 ; 100 Hz

ltas_datoteka$ = direktorij$ + "ltas_10ms.txt"
int_datoteka$ = direktorij$ + "int_10ms.txt"
pitch_datoteka$ = direktorij$ + "pitch_10ms.txt"

;=====================================================

Read from file: direktorij$ + ime_datoteke$ + ".wav"
;;;writeFileLine: int_datoteka$, "INTENZITET"
;;;writeFileLine: pitch_datoteka$, "TON"

selectObject: "Sound " + ime_datoteke$
To Intensity: 75, 0.001
Rename: "intensity"

selectObject: "Sound " + ime_datoteke$
To Pitch: 0.001, 75, 300
Rename: "pitch"

selectObject: "Sound " + ime_datoteke$
zvuk = selected("Sound")
Edit
a = pocetak
b = pocetak + korak
while a < kraj
      select zvuk
      editor: zvuk

      Select: a, b
      v = View spectral slice
      endeditor

      ltas = To Ltas: ltas_raspon
      vrijeme$ = fixed$(a, 4)
      ime$ = replace$(vrijeme$, ".", "_", 1)
      Rename: ime$      ;;; postavi ime za LTAS objekt
```

```
      ;;; --- informacije o intenzitetu
      selectObject: "Intensity intensity"
      intensity = Get value at time: a, "Cubic"
      appendFileLine: int_datoteka$, string$(a) + ":" + string$(intensity)

      ;;; --- informacije o tonu
      selectObject: "Pitch pitch"
      pitch = Get value at time: a, "Hertz", "Linear"
      appendFileLine: pitch_datoteka$, string$(a) + ":" + string$(pitch)

      a = a + korak
      b = b + korak

      removeObject: v
endwhile

;;; oznaci sve LTAS objekte
;;;
a = pocetak
b = pocetak + korak
while a < kraj
      vrijeme$ = fixed$(a, 4)
      ime$ = replace$(vrijeme$, ".", "_", 1)
      plusObject: "Ltas " + ime$

      a = a + korak
      b = b + korak
endwhile

;;; spremi oznacene LTASove u datoteku
;;;
Save as text file: ltas_datoteka$

;;; obrisi oznacene LTAS objekte
;;;
Remove

selectObject: "Intensity intensity"
Remove
```

Prilog 2: Praat skript 2 – za klasifikaciju po glotalnim pulsevima

```
;;; NIJE POTREBNO UCITATI KOLEKCIJU U PRAAT!

;;; *** OBAVEZNI PARAMETRI ***
;;;
direktorij$ =
"C:\Users\Alex\OneDrive\FFZG\doktorat\podaci\za_detekciju\detekcija\test\"
ime_datoteke$ = "zvuk"  ; ovo treba biti .wav datoteka

;===============================================================================

ltas_raspon = 100 ; 100 Hz

ltas_datoteka$ = direktorij$ + "ltas_puls.txt"
int_datoteka$ = direktorij$ + "int_puls.txt"
pitch_datoteka$ = direktorij$ + "pitch_puls.txt"

;===============================================================================

;;; *** ucitavanje pozicija pulseva ***
Read Strings from raw text file: direktorij$ + "_pulsevi.txt"
broj_pulseva = Get number of strings
for redni_broj from 1 to broj_pulseva
      str$ = Get string: redni_broj
      v = number(str$)
      pulsevi[redni_broj] = v
endfor


Read from file: direktorij$ + ime_datoteke$ + ".wav"
;;;writeFileLine: int_datoteka$, "INTENZITET"
;;;writeFileLine: pitch_datoteka$, "TON"

selectObject: "Sound " + ime_datoteke$
To Intensity: 75, 0.001
Rename: "intensity"

selectObject: "Sound " + ime_datoteke$
To Pitch: 0.001, 75, 300
Rename: "pitch"

selectObject: "Sound " + ime_datoteke$
zvuk = selected("Sound")
Edit

for puls from 1 to broj_pulseva
      pozicija = pulsevi[puls]
      a = pozicija - 0.005
      b = pozicija + 0.005

      select zvuk
      editor: zvuk

      Select: a, b
      v = View spectral slice
      endeditor

      ltas = To Ltas: ltas_raspon
      vrijeme$ = fixed$(pozicija, 4)
      ime$ = replace$(vrijeme$, ".", "_", 1)
```

```
      Rename: ime$      ;;; postavi ime za LTAS objekt

      ;;; --- informacije o intenzitetu
      selectObject: "Intensity intensity"
      intensity = Get value at time: a, "Cubic"
      appendFileLine: int_datoteka$, string$(a) + ":" + string$(intensity)

      ;;; --- informacije o tonu
      selectObject: "Pitch pitch"
      pitch = Get value at time: a, "Hertz", "Linear"
      appendFileLine: pitch_datoteka$, string$(a) + ":" + string$(pitch)

      removeObject: v
endfor

;;; oznaci sve LTAS objekte
;;;
for puls from 1 to broj_pulseva
      pozicija = pulsevi[puls]
      vrijeme$ = fixed$(pozicija, 4)
      ime$ = replace$(vrijeme$, ".", "_", 1)
      plusObject: "Ltas " + ime$
endfor

;;; spremi oznacene LTASove u datoteku
;;;
Save as text file: ltas_datoteka$

selectObject: "Intensity intensity"
```

Prilog 3: Programski kôd za modul *neuralna mreža.py*

```
from sklearn.neural_network import MLPClassifier
import pickle
import os
import re

import citac
import util


def skalirano_01(ltas: map):
    """
    * Skaliranje vrijednosti na interval [0, 1] (potrebno za neuralnu
mrezu)
    * Podesavanje vrijednosti na odredjeni intenzitet (tako da ne ovisimo o
    razlikama u intenzitetu)

    :param ltas: mapa gdje je kljuc vrijeme, a vrijednost LTAS (LTAS je
    lista koja se nalazi u drugoj listi)
    :return: Mapa gdje je kljuc naziv glasa, a vrijednost LTAS za taj glas.
    Vrijednost je u stvari lista koja sadrzi drugu listu (zbog treniranja
    i upita za mrezu, ali to je vjerojatno zbog toga sto se mrezi moze
    zadati niz vrijednosti koje treba klasificirati, ne samo jednu kako sad
    radim).
    """

    def _skalirano_po_intenzitetu(ltas_vr):
        najveci = max(ltas_vr)
        granica = 30
        promjena = (najveci - granica) / najveci
        rezultat = list(map(lambda v: v - v * promjena, ltas_vr))
        return rezultat

    def _skalirana_vrijednost_na_01(ltas_vr):
        p = _skalirano_po_intenzitetu(ltas_vr)
        vr_min = min(p)
        vr_max = max(p)

        # algoritam "feature scaling"
        return list(map(lambda x: (x - vr_min) / (vr_max - vr_min), p))

    def _skaliran_glas_na_01(ltas_liste: list):
        return list(
            map(lambda ltas_glasa: _skalirana_vrijednost_na_01(ltas_glasa),
                ltas_liste))

    skalirano = {}
    for naziv in ltas:
        skalirano[naziv] = _skaliran_glas_na_01(ltas[naziv])

    return skalirano


def treniraj():
    """
    Korak 1: Formiraj sljedecu strukturu:
        ([LTAS-1, LTAS-2, ..., LTAS-n],
         [GLAS-1, GLAS-2, ..., GLAS-n])
```

```
    gdje je LTAS-k lista LTAS vrijednosti, a GLAS-k glas koji odgovara
listi
    LTAS-k. Na primjer:
        ([[0.3, 0.1, ...], [0.7, 0.2, ...], ...], ['n', 'b', ...])

    Prvi element para je lista drugih lista koje sadrze LTAS vrijednosti
    nekog glasa, a drugi element para je taj glas (na poziciji na kojoj se
    nalazi njegova LTAS lista).
    Svaka LTAS lista ce, pored ostalih vrijednosti, sadrzavati 1 kao
najvecu
    vrijednost i 0 kao najmanju.

    Korak 2: Treniraj NM koristeci prethodno napravljene kategorije.

    :return: (vidi gore)
    """

    def _serijaliziraj(obj):
        with open(util.konfig('datoteka', 'kategorije'), 'wb') as f:
            pickle.dump(obj, f, pickle.HIGHEST_PROTOCOL)

    direktorij = util.konfig('datoteka', 'tset')
    ltas = []
    nove_kategorije = []
    print(util.konfig('datoteka', 'tset'))
    oznacene_ltas_datoteke = list(os.walk(util.konfig('datoteka',
'tset')))[0][
        2]
    for dat in oznacene_ltas_datoteke:  # ucitavamo iz svih datoteka
        print('uzorak:', dat)
        tr_set = citac.PraatLTASDatoteka(direktorij + dat)
        tr_set.ucitaj()
        skalirano = skalirano_01(tr_set.vrijednosti())

        for k in skalirano:
            ltas += skalirano[k]
            nove_kategorije += [k] * len(
                skalirano[k])  # jedan glas moze imati vise LTAS lista

    _serijaliziraj((ltas, nove_kategorije))
    print('treniranje ...')
    treniraj_mrezu(util.konfig('datoteka', 'ann'), (ltas, nove_kategorije))


def treniraj_mrezu(ime_ann_datoteke, ltas_kategorije):
    """
    Ovo je format podataka za NM:
        ltas = ([23., 14., ...], [18., 72., ...])   vrijednosti (LTAS)
        kategorije = ['A', 'B']             kategorije za svaku podlistu u
ltas

    :param ime_ann_datoteke
    :param ltas_kategorije
    :return: mreza istrenirana sa zadanim LTAS datotekama
    """
    mreza = MLPClassifier(
            solver='lbfgs',  # preporucuje se za manju kolicinu podataka
            alpha=1e-5,      # zadano (default) za korekciju
            hidden_layer_sizes=(120, 120, 160),
            random_state=1)  # za generator slucajnih brojeva
```

```
    # izgleda da se mreza mora trenirati odjednom sa svim podacima
    # ltas, kategorije = kategorije_glasova(oznacene_ltas_datoteke)
    ltas, kategorije = ltas_kategorije
    mreza.fit(ltas, kategorije)  # treniranje

    # serijaliziraj treniranu mrezu
    with open(ime_ann_datoteke, 'wb') as f:
        pickle.dump(mreza, f, pickle.HIGHEST_PROTOCOL)

    return mreza


def klasificiraj(direktorij, ltas_datoteka, int_datoteka,
izlazna_datoteka):
    """
    Trazimo od trenirane NM da klasificira glasove. Klasificirani glasovi
    se spremaju u izlaznu datoteku. NM je serijalizirani objekt prethodno
    istrenirane mreze.

    :param direktorij: ...
    :param ltas_datoteka: LTAS datoteka neoznacene snimke govora
    :param int_datoteka:
    :param izlazna_datoteka: ...
    :return: Lista ntorki (slovo, vrijeme)
    """

    ulaz = citac.PraatLTASDatoteka(rf'{direktorij}\{ltas_datoteka}')
    ulaz.ucitaj()
    vrijednosti = skalirano_01(ulaz.vrijednosti())

    try:
        with open(util.konfig('datoteka', 'ann'), 'rb') as f:
            mreza = pickle.load(f)

        popis_glasova = []
        with open(rf'{direktorij}\{izlazna_datoteka}', 'w',
                  encoding='utf8') as f:
            # vrijeme_glasa je u stvari naziv koji oznacava vrijeme kao
5_23
            for vrijeme_glasa in vrijednosti:
                vrijeme = round(
                    float(vrijeme_glasa.replace('_', '.')), 4)

                # klasificiraj skalirani glas (rezultat je tipa
numpy.ndarray)
                klasificiran_glas =
mreza.predict(vrijednosti[vrijeme_glasa])
                slovo = util.slovo_glasa(klasificiran_glas[0])

                # ovdje je vrijeme u formatu x.yz
                popis_glasova += [(slovo, vrijeme)]
                f.write(slovo + ' (' + vrijeme_glasa + ')\n')

        return popis_glasova
    except Exception as e:
        print(e)


def frekv(niz):
    m = {}
    for e in niz:
```

```
        if e in m:
            m[e] += 1
        else:
            m[e] = 1

    for e in m:
        m[e] = round(m[e] / len(niz) * 100, 2)

    return m


def glas_pulsa(puls, a, b):
    """
    Ovi glasovi su uzeti iz pulseva.
    :param puls:
    :param a:
    :param b:
    :return: string (ostalih) glasova iz pulseva
    """

    c_glas = 0
    c_vrijeme = 1

    g_ostali_glasovi = [(g[c_glas], g[c_vrijeme]) for g in puls if
                        g[c_glas] not in 'ieaoušđžčcsz']
    r = []
    od = None
    for g in g_ostali_glasovi:
        if a < g[c_vrijeme] < b:
            r += [g]
            if od is None:
                od = g[c_vrijeme]

    if len(r) > 0:
        do = r[-1][c_vrijeme]
    else:
        do = None

    popis_glasova = list(map(lambda e: e[c_glas], r))
    return ''.join(popis_glasova), od, do


def postotak_glasova(niz: str):
    """
    Koliki je udio pojedinih glasova u grupi glasova (samo informativno).
    :param niz: Niz glasova.
    :return: Lista [(glas, postotak), ...]
    """
    return sorted(frekv(niz).items(), key=lambda x: x[1], reverse=True)


def izlazni_podaci(ms10, puls):
    """
    Ovo je pocetna funkcija iz koje se dobija rezultat prepoznavanja
glasova
    od strane neuralne mreze. Ova funkcija grupira glasove prema zadanim
    kategorijama tako da se glasovi iste kategorije nalaze u jednom nizu
koji
    predstavlja tu grupu glasova.

    :param ms10: Lista parova (glas, vrijeme)
```

```
    :param puls: Lista parova (glas, vrijeme)
    :return: Lista ntorki. FORMAT: (od, do, niz_glasova, udio_glasova)
    """
    # popis_glasova = objedini_popis_glasova(ms10, puls)
    popis_glasova = util.objedini(ms10, puls, lambda p: p[1])
    vremena = [a[1] for a in popis_glasova]
    slova = [a[0] for a in popis_glasova]
    popis = []
    prethodna_ntorka = None
    istaknuti_glasovi = re.finditer('([aeiou]+)|([šđžč]+)|([csz]+)',
                                    ''.join(slova))

    # dodaju se svi glasovi iz gornjeg RE (10 ms i pulsevi), a ostali
glasovi
    # dolaze iz pulseva
    for glas in istaknuti_glasovi:
        string_glasova = glas.group()
        if len(string_glasova) < 3:  # ne gledamo nizove s manje od 3 glasa
            continue

        a = vremena[glas.span()[0]]
        b = vremena[glas.span()[1] - 1]
        udio_glasova = postotak_glasova(string_glasova)

        ntorka = (a, b, string_glasova, udio_glasova)

        # umetni ostale glasove koji se nalaze u ovom intervalu (ako
postoje)
        if prethodna_ntorka is not None:
            pocetak_prethodni = prethodna_ntorka[0]
            pocetak_tekuci = ntorka[0]

            ostali_glasovi, od, do = glas_pulsa(puls, pocetak_prethodni,
                                                pocetak_tekuci)
            if len(ostali_glasovi) > 0:
                udio_ostalih_glasova = postotak_glasova(ostali_glasovi)
                popis += [(od, do, '*' + ostali_glasovi,
udio_ostalih_glasova)]

        popis += [ntorka]
        prethodna_ntorka = ntorka

    return popis


def klasificiraj_glasove(direktorij):
    """
    Ovdje se klasificira cijela snimka.

    :return: Lista ntorki formata [(od, do, grupa glasova, statistika),
...]
    kao [(0.0894, 0.0947, 'eee', [('e', 100.0)]),
         (0.1, 0.1634, 'zzzzzszzzzz', [('z', 90.91), ('s', 9.09)]),
         ...]
    """
    klasf_ms10 = klasificiraj(util.konfig('datoteka', 'dir') + direktorij,
                              'ltas_10ms.txt',
                              'int_10ms.txt',
                              'klasifikacija_10ms.txt')

    klasf_puls = klasificiraj(util.konfig('datoteka', 'dir') + direktorij,
```

```
                                'ltas_puls.txt',
                                'int_puls.txt',
                                'klasifikacija_puls.txt')

    return izlazni_podaci(klasf_ms10, klasf_puls)


if __name__ == "__main__":
    treniraj()
```

Prilog 4: Programski kôd za modul *poravnanje.py*

```
import util


class Cvor:
    memo_id = 0  # ovo je samo info za debugiranje

    def __init__(self, dubina, vrijednost, indeks):
        self.prethodnik = None
        self.sljedbenici = []
        self.vrijednost = vrijednost
        self.ukupno = vrijednost
        self.putanja = []
        self.indeks = indeks
        self.dubina = dubina
        self.id = Cvor.memo_id
        Cvor.memo_id += 1

    def __repr__(self):  # info za debugiranje
        return '<Cvor>'

    def dodaj(self, cvor):
        cvor.prethodnik = self
        self.sljedbenici += [cvor]
        return cvor

    def sljedbenik(self, cvor):
        if len(self.sljedbenici) == 0:
            return self.dodaj(cvor)
        else:
            cvor.prethodnik = self
            self.sljedbenici[-1] = cvor
            return cvor

    @staticmethod
    def ispisi(cvor, oznaka, dubina=0):
        print(str(dubina) + oznaka, ' ' * (dubina * 2), cvor.indeks,
              '(' + str(cvor.vrijednost) + '/' + str(cvor.ukupno) + ')',
              cvor.putanja, '#' + str(cvor.id) if cvor.id is not None else
''')
        for e in cvor.sljedbenici:
            Cvor.ispisi(e, oznaka, dubina + 1)


def nadji_poravnanje(c_glasovi, c_tekst, c_maks_razmak, fn_procjena,
                     ispis=True):
    """
    Ovaj je algoritam detaljno opisan u disertaciji.

    :param c_glasovi:
    :param c_tekst:
    :param c_maks_razmak:
    :param fn_procjena:
    :param ispis:

    :return: Lista parova (indeks grupe glasova, grupa glasova), kao
    [(0, ['z', 's', 'z']), (2, ['a', 'o']), (3, ['r', 'r', 'r']),
    (4, ['e', 'a', 'a', 'a'])]
    U ovom primjeru tekst je 'soba'. Indeks prvog para znaci da se grupa
    ['z', 's', 'z'] povezuje sa slovom 's', grupa ['a', 'o'] sa slovom 'o',
```

```
    grupa ['r', 'r', 'r'] sa slovom 'b' i grupa ['e', 'a', 'a', 'a'] sa
slovom
    'a'. Indeks na kojem se nalazi cijeli par, kao sto je (2, ['a', 'o'])
koji
    se nalazi na indeksu 1 (ne 2), oznacava indeks slova s kojim se doticna
    grupa poravnava. U ovom primjeru, grupa ['a', 'o'] poravnava se sa
slovom
    na indeksu 1 ('o') jer se taj par nalazi na indeksu 1.
    """
    memo = {}

    def _izdvoji_kombinaciju(cvor, popis):
        i = cvor.indeks
        popis += [(i, c_glasovi[i])]
        if len(cvor.putanja) > 1:
            _izdvoji_kombinaciju(cvor.putanja[1], popis)

        return popis

    najbolji_cvor = None

    def _trazi(
            dubina,
            indeks,
            indeks_roditelja,
            put,
            ukupno,
            cvor,
            c_maks_dubina,
            c_maks_indeks):

        nonlocal najbolji_cvor

        # memoizacija
        if 0 < dubina < c_maks_dubina:
            kljuc = str(dubina - 1) + ':' + str(indeks - 1)
            if kljuc in memo:
                memo_cvor = memo[kljuc]
                if ispis:
                    Cvor.ispisi(memo_cvor, 'M', dubina)

                # Trenutni cvor nam ne treba pa memoizirano stablo dodajemo
                # kao sljedbenik prethodnika.
                cvor.prethodnik.sljedbenik(memo_cvor)

                # Kljucna naredba - ovdje prekidamo nepotrebno ponavljanje
                # kombinacija.
                return memo_cvor
            else:
                # Dodaj novi memo (samo inicijalni podkorijen - sljedbenici
ce
                # se nadovezati kasnije).
                memo[kljuc] = cvor

        # _ispisi_cvor(cvor, dubina)
        if ispis:
            Cvor.ispisi(cvor, ':', dubina)

        if dubina >= c_maks_dubina:
            cvor.putanja += [cvor.indeks]
            return cvor
```

```
        for i in range(indeks, c_maks_indeks + 1):
            # dubina pocinje od -1
            # preskacemo razmake vece od maks_razmak
            if dubina >= 0 and i > indeks_roditelja + c_maks_razmak:
                return cvor

            # ovaj cvor ide na sljedecu razinu
            procjena = fn_procjena(c_tekst, c_glasovi, indeks, dubina + 1)
            sljedbenik = cvor.dodaj(Cvor(dubina + 1, procjena, i))
            zadnji_cvor = _trazi(dubina + 1, indeks + 1, i, put + [i],
                                 ukupno + sljedbenik.vrijednost,
                                 sljedbenik, c_maks_dubina, c_maks_indeks)

            # Nastavljamo od indeksa koji je za 1 veci od trenutnog indeksa
na
            # ovoj dubini.
            indeks += 1

            if cvor.vrijednost + zadnji_cvor.ukupno > cvor.ukupno:
                cvor.ukupno = cvor.vrijednost + zadnji_cvor.ukupno
                cvor.putanja = [zadnji_cvor.id, zadnji_cvor]

            if dubina == 0:
                if najbolji_cvor is None:
                    najbolji_cvor = (cvor.id, round(cvor.ukupno, 2), cvor)
                elif najbolji_cvor[1] < cvor.ukupno:
                    najbolji_cvor = (cvor.id, round(cvor.ukupno, 2), cvor)

        return cvor

    korijen = Cvor(-1, 0, -1)

    # korijen je na dubini -1
    if len(c_glasovi) < len(c_tekst):
        _trazi(korijen.dubina, 0, 0, [], 0, korijen, len(c_glasovi) - 1,
               len(c_tekst) - 1)
    else:
        _trazi(korijen.dubina, 0, 0, [], 0, korijen, len(c_tekst) - 1,
               len(c_glasovi) - 1)

    if ispis:
        print('\n*** NAJBOLJI CVOR')
        Cvor.ispisi(najbolji_cvor[2], '>')

    # print('najbolji cvor:', najbolji_cvor)
    kombinacija = _izdvoji_kombinaciju(najbolji_cvor[2], [])
    return kombinacija, najbolji_cvor[1]  # kombinacija i bodovi


def poravnaj_slova_s_glasovima(uredjen_tekst, tabela_glasova):
    def _procjena(tekst, glasovi, indeks, dubina):
        try:
            dodatni_bodovi = {
                'ieaou': 0.3,
                'čš': 3.5,
                'cs': 3.5,
                'đžz': 3.5,
            }

            slovo = tekst[dubina]
```

```
            # Ovo baca iznimku ako je indeks veci od broja glasova, sto se
            # desava kada glas ili grupa glasova nije mogla biti poravnata
sa
            # slovom.
            gl = glasovi[indeks]

            r = 0
            if slovo in gl:
                r = 0.5  # glasovima ide 0.5 boda jer se ne zna jesu li
ispravni

            # ako je glas jedan od onih koje mreza bolje prepoznaje dodaj
bodove
            for grupa_glasova in dodatni_bodovi.keys():
                # jesu li i slovo i glas u istom skupu?
                if slovo in grupa_glasova and \
                        (len(set(gl) & set(grupa_glasova)) > 0):
                    r += dodatni_bodovi[grupa_glasova]
                    break

            return r
        except:
            return 0  # glas nije povezan s tekstom

    # Izdvoji samo glasove (t.j. skupove glasova za svaki element u
izlaznom
    # popisu).
    grupe_glasova = list(map(lambda redak: set(redak['grupa_glasova']),
                             tabela_glasova))
    vremena = list(map(lambda redak: (redak['od'], redak['do']),
                       tabela_glasova))

    pozicije, bodovi = nadji_poravnanje(
            grupe_glasova,
            uredjen_tekst,
            2,
            _procjena,
            ispis=False)

    # ovdje se radi konacni popis poravnatih slova s glasovima u formatu
    # [[slovo, (od, do)], ...]
    popis = []
    for i in range(len(uredjen_tekst)):
        # ako su iscrpljene sve grupe glasova, prekidamo
        if i >= len(pozicije):
            break

        popis += [(uredjen_tekst[i], vremena[pozicije[i][0]])]

    return popis, bodovi


def napravi_tabelu_glasova(
        glasovi,
        pocetak_zvuka,
        zavrsetak_zvuka):
    """
    Daje popis glasova organiziranih u tabelu <od, do, grupa_glasova>.

    :param glasovi: Direktorij s podacima generiranih od strane Praata.
```

```
    :param pocetak_zvuka: Pocetak segmenta zvuka.
    :param zavrsetak_zvuka: Zavrsetak segmenta zvuka.
    :return: Lista mapa s informacijama o grupi glasova.
    """
    tabela = []
    gr_gl = []  # izdvojena statistika iz grupe glasova
    vremena = []
    for e in glasovi:
        od = e[0]
        if pocetak_zvuka is not None \
                and od != 0 \
                and od < pocetak_zvuka:
            continue

        do = e[1]
        grupa_glasova = e[2]  # ovo je u tekstualnom obliku (niz slova)
        gr_gl += [e[3]]
        vremena += [(od, do)]

        # svaki redak je mapa s poljima koji odgovaraju stupcima tablice
        tabela += [{
            'od': od,
            'do': do,
            'grupa_glasova': grupa_glasova,
            # 'statistika': e[3],
        }]

        if (zavrsetak_zvuka is not None
                and do != 0
                and do > zavrsetak_zvuka):
            break

    return tabela


def intervali_rijeci(priprema_teksta_obj, poravnanje):
    """
    Razdvaja popis glasova prema granicama rijeci (oznacenim s ' ').

    :return:
    """
    # niz prvo mora imati sva slova iz titla (bez konverzije)
    izvorni_niz = priprema_teksta_obj.izvorni_niz(poravnanje)
    intervali_rijeci = []
    jedan_interval = []
    for e in izvorni_niz:
        if e == ' ':  # zavrsava rijec?
            intervali_rijeci += [jedan_interval]
            jedan_interval = []
        else:
            jedan_interval += [e]

    '''
    Ovo daje niz s informacijama o poziciji glasova; svaka podlista je
jedna
    rijec. Primjer rijeci 'su':
        [..., [('s', (21.9523, 22.06)), ('u', (22.07, 22.086))], ...]
    '''
    intervali_rijeci += [jedan_interval]  # dodaj ostatak
    rezulat = {}
    for elementi_rijeci in intervali_rijeci:
```

```
        try:
            interval = (round(elementi_rijeci[0][1][0], 2),
                        round(elementi_rijeci[-1][1][0], 2))
            rijec = ''
            for e in elementi_rijeci:
                rijec += e[0]
            obradjena_rijec = util.PripremaTeksta(rijec).obradjen_tekst()
            rezulat[obradjena_rijec] = interval
        except:
            # print('*** GRESKA:', elementi_rijeci)
            pass

    return rezulat


#
================================================================================
===


if __name__ == '__main__':
    def g_procjena(c_tekst, c_glasovi, indeks, dubina):

        def _dodatni_bodovi(s, g):
            for k in dodatni_bodovi.keys():
                # jesu li i slovo i glas u istom skupu?
                if s in k and (len(set(g) & set(k)) > 0):
                    return dodatni_bodovi[k]

            return 0

        dodatni_bodovi = {
            'ieaou': 0.3,
            'čš': 3.5,
            'cs': 3.5,
            'đžz': 3.5,
        }

        slovo = c_tekst[dubina]
        glasovi = c_glasovi[indeks]

        r = 0
        if slovo in glasovi:
            r = 0.5  # glasovima ide 0.5 boda jer se ne zna jesu li
ispravni

        # Ako je glas jedan od onih koje mreza bolje prepoznaje dodaju se
        # dodatni bodovi.
        r += _dodatni_bodovi(slovo, glasovi)
        return r


    def test_podudaranje():
        grupe = [
            ['z', 's', 'z'],
            ['m', 'n'],
            ['a', 'o'],
            ['r', 'r', 'r'],
            ['e', 'a', 'a', 'a'],
            ['r', 'v', 'r', 'v'],
            ['e', 'a', 'e'],
```

```
    ]

    tekst = 'soba'

    k, b = nadji_poravnanje(grupe, tekst, 2, g_procjena, True)
    print(k, '\nBODOVI:', b)


test_podudaranje()
```

Prilog 5: Programski kôd za modul *detektor_rucni.py*

```
import statistics

import podjela_na_slogove


def izdvoji_rijeci(
        ispravni_intervali,
        tabela_glasova,
        tg_poravnanje,
        intenzitet,
        ton,
        obrada_teksta):
    """

    :param tabela_glasova:
    :param ton:
    :param intenzitet:
    :param ispravni_intervali: Intervali rijeci utvrdjeni slusanjem.
    :param tg_poravnanje: Poravnanje teksta s glasovima (vidi funkciju).
    :param obrada_teksta: Objekt tipa PripremaTeksta.

    :return: Niz mapa gdje svaka mapa predstavlja informacije o jednoj
    rijeci, formata [{'rijec':..., 'od':..., 'do':..., ...}, {...}, ...].
    """

    def _intervali_rijeci():
        """
        Razdvaja popis glasova prema granicama rijeci (oznacenim s ' ').

        :return: Niz s informacijama o poziciji glasova; svaka podlista je
        jedna rijec. Primjer rijeci 'su':
            [..., [('s', (21.9523, 22.06)), ('u', (22.07, 22.086))], ...]
        """
        izvorni_niz = obrada_teksta.izvorni_niz(tg_poravnanje)
        intervali_rijeci = []
        jedan_interval = []
        for e in izvorni_niz:
            if e == ' ':  # zavrsava rijec?
                intervali_rijeci += [jedan_interval]
                jedan_interval = []
            else:
                jedan_interval += [e]

        intervali_rijeci += [jedan_interval]  # dodaj ostatak
        return intervali_rijeci

    def _popis_rijeci(intervali_rijeci):
        """
        Daje listu rijeci s informacijom o vremenskoj poziciji.

        :param intervali_rijeci: (gornja funkcija)
        :return: Niz formata [(rijec, od, do), ...].
        """
        popis_rijeci = []
        for interval in intervali_rijeci:
            pocetno_vrijeme, zavrsno_vrijeme = None, None
            rijec = ''
            for e in interval:
                if isinstance(e, tuple):
```

```
                    if pocetno_vrijeme is None:
                        pocetno_vrijeme = e[1][0]
                    zavrsno_vrijeme = e[1][1]
                    rijec += e[0]
                else:
                    rijec += e

            if rijec != '':
                popis_rijeci += [(rijec, pocetno_vrijeme, zavrsno_vrijeme)]

        return popis_rijeci

    def _tabela(popis_rijeci):
        """
        Formira tabelu koja sadrzi sve potrebne podatke.
        :param popis_rijeci: (vidi poziv)
        :return: Niz mapa.
        """
        tabela = []
        for i in range(len(popis_rijeci)):
            if i >= len(ispravni_intervali):
                break
            rijec = popis_rijeci[i][0]
            od = popis_rijeci[i][1]
            do = popis_rijeci[i][2]
            ispravno_od = ispravni_intervali[i][1]
            ispravno_do = ispravni_intervali[i][2]
            odst_od = (popis_rijeci[i][1] - ispravni_intervali[i][1]) *
1000
            odst_do = (popis_rijeci[i][2] - ispravni_intervali[i][2]) *
1000
            slogovi = podjela_na_slogove.Slogovi(rijec).podjela()
            br_slogova = len(slogovi)

            # intenzitet
            int_segment = list(filter(lambda v: od <= v[0] <= do,
intenzitet))
            int_min = -1
            int_maks = -1
            int_prosjek = -1
            if len(int_segment) > 0:
                int_min = round(min([v[1] for v in int_segment]), 2)
                int_maks = round(max([v[1] for v in int_segment]), 2)
                int_prosjek = round(statistics.mean([v[1]
                                                     for v in
int_segment]), 2)

            # ton
            ton_segment = list(filter(lambda v: od <= v[0] <= do, ton))
            ton_min = -1
            ton_maks = -1
            ton_prosjek = -1
            if len(ton_segment) > 0:
                ton_min = round(min([v[1] for v in ton_segment]), 2)
                ton_maks = round(max([v[1] for v in ton_segment]), 2)
                ton_prosjek = round(statistics.mean([v[1]
                                                     for v in
ton_segment]), 2)

            # trajanje: za svaku rijec izdvajamo najdulji niz vokala
            interval_rijeci = od, do
```

```
            segment_glasova = list(
                    filter(lambda g: g['od'] >= interval_rijeci[0]
                                     and g['do'] <= interval_rijeci[1]
                                     and g['grupa_glasova'][0] in 'ieaou',
                           tabela_glasova))
            if len(segment_glasova) > 0:
                sortirani_segment_glasova = \
                    sorted(segment_glasova,
                           key=lambda e: len(e['grupa_glasova']))
                duljina_vokala = \
                    len(sortirani_segment_glasova[-1]['grupa_glasova'])
            else:
                duljina_vokala = 0

            redak = {
                'rijec': rijec,
                'od': od,
                'do': do,
                'ispravno_od': ispravno_od,
                'ispravno_do': ispravno_do,
                'odstupanje_od': odst_od,
                'odstupanje_do': odst_do,
                'broj_slogova': br_slogova,
                'slogovi': slogovi,
                'intenzitet': (int_min, int_maks, int_prosjek),
                'ton': (ton_min, ton_maks, ton_prosjek),
                'trajanje': duljina_vokala,

                # ovo se postavlja kasnije kod analize prozodije
                'istaknut_intenzitet': None,
                'istaknut_ton': None,
                'istaknuto_trajanje': None,
            }

            tabela += [redak]

        return tabela

    intervali = _intervali_rijeci()
    rijeci = _popis_rijeci(intervali)
    return _tabela(rijeci)


def analiza_tempa(tabela, min_br_slogova):
    """
    Iz popisa rijeci izdvaja rijeci u grupe od minimalno min_br_slogova.

    :param min_br_slogova: Minimalni broj slogova po grupi.
    :param tabela: Tabela s rijecima.

    :return: Lista redova tabele (mapa) grupiranih tako da sadrze najmanje
    min_br_slogova. FORMAT: [[{redak}, {redak}, ...], [...], ...].
    """
    grupe = []
    grupa_redova = []
    br_slogova = 0
    for redak in tabela:
        grupa_redova += [redak]
        br_slogova += redak['broj_slogova']
        if br_slogova >= min_br_slogova:  # pripremi za novu grupu
            grupe += [grupa_redova]
```

```
            grupa_redova = []
            br_slogova = 0

    if len(grupa_redova) > 0:
        grupe += [grupa_redova]  # dodaj ostatak

    return grupe


def analiziraj_prozodiju(
        tabela_rijeci,
        tabela_glasova,
        odstupanje_ton,
        odstupanje_int,
        odstupanje_trajanje,
        velicina_prozora):
    """
    Ova funkcija postavlja informacije o intenzitetu, tonu i trajanju u
tabelu
    rijeci.

    :param tabela_glasova:
    :param velicina_prozora: Koliko rijeci uzimamo kao kontekst za
detekciju
                             trajanja.
    :param tabela_rijeci: Popis rijeci sa svim informacijama.
    :param odstupanje_ton: Postotak odstupanja od okolnih vrijednosti koji
se
                           smatra isticanjem.
    :param odstupanje_int: Isto kao za ton.
    :param odstupanje_trajanje: Isto kao za ton.
    :return: None
    """

    def _istaknut_ton():
        for i in range(len(tabela_rijeci)):
            redak = tabela_rijeci[i]
            ton_maks = redak['ton'][1]
            if 0 < i < len(tabela_rijeci) - 1:
                ton_prethodni = tabela_rijeci[i - 1]['ton'][1]
                ton_iduci = tabela_rijeci[i + 1]['ton'][1]
                if ton_maks >= ton_prethodni + ton_prethodni *
odstupanje_ton \
                        and ton_maks >= ton_iduci + ton_iduci *
odstupanje_ton:
                    tabela_rijeci[i]['istaknut_ton'] = True

    def _istaknut_intenzitet():
        for i in range(len(tabela_rijeci)):
            redak = tabela_rijeci[i]
            int_maks = redak['intenzitet'][1]
            if 0 < i < len(tabela_rijeci) - 1:
                int_prethodni = tabela_rijeci[i - 1]['intenzitet'][1]
                int_iduci = tabela_rijeci[i + 1]['intenzitet'][1]
                if int_maks >= int_prethodni + int_prethodni *
odstupanje_int \
                        and int_maks >= int_iduci + int_iduci *
odstupanje_int:
                    tabela_rijeci[i]['istaknut_intenzitet'] = True

    def _najduzi_segment_vokala(od, do):
```

```
        """
        Vraca segment vokala za zadani interval glasova.

        :param od:
        :param do:
        :return: Lista grupa vokala zadanog segmenta, npr.
                ['aaaaea', 'iieieeaa', ...].
        """
        vokali_rijeci = []
        for redak in tabela_glasova:
            if redak['od'] >= od \
                    and redak['do'] <= do \
                    and redak['grupa_glasova'][0] in 'ieaou':
                vokali_rijeci += [redak['grupa_glasova']]

        if len(vokali_rijeci) > 0:
            return max(vokali_rijeci, key=len)
        else:
            return ''

    def _iznad_praga(prozor, rijec_od, rijec_do, indeks_rijeci):
        najduzi_trenutni = _najduzi_segment_vokala(rijec_od, rijec_do)
        segment_prozora = tabela_rijeci[prozor[0]:indeks_rijeci] \
                          + tabela_rijeci[indeks_rijeci + 1:prozor[1] + 1]
        ostali = []
        for rijec in segment_prozora:
            od = rijec['od']
            do = rijec['do']
            ostali += [_najduzi_segment_vokala(od, do)]

        najduzi_od_ostalih = max(ostali, key=len)
        if len(najduzi_trenutni) > 0:
            if len(najduzi_od_ostalih) / len(najduzi_trenutni) \
                    < odstupanje_trajanje:
                return True
            else:
                return False
        else:
            return False

    def _istaknuto_trajanje():
        """
        Za svaku rijec odredi je li istaknuto trajanje.

        :return: None
        """
        prozor = 0, velicina_prozora - 1

        # sve unutar prozora uzimamo kao kontekst za analizu trenutne
rijeci
        for i, redak in enumerate(tabela_rijeci):
            if i > prozor[1]:  # dosli smo do kraja prozora, pocni ga
spustati
                prozor = prozor[0] + 1, prozor[1] + 1

            # Prvo uzmemo najdulji segment vokala trenutne rijeci, a zatim
            # najdulji segment ostalih rijeci unutar prozora.
            if _iznad_praga(prozor, redak['od'], redak['do'], i):
                tabela_rijeci[i]['istaknuto_trajanje'] = True

    _istaknut_ton()
```

```
_istaknut_intenzitet()
_istaknuto_trajanje()
```

Prilog 6: Programski kôd za modul *detektor_automatski.py*

```
from collections import namedtuple
from enum import Enum
import pickle
import string_util
import os

import util
import citac
import neuralna_mreza as nm
import poravnavanje

RijecInfo = namedtuple('RijecInfo',
                       ['rijec',
                        'glasovi',
                        'interval1',
                        'interval2',
                        'int_maks_1',
                        'ton_maks_1',
                        'int_maks_2',
                        'ton_maks_2'])

Rezultat = namedtuple('Rezultat',
                      ['rijec',
                       'interval1',
                       'int_maks_1',
                       'odst_int_maks_1',
                       'int_oznaka_1',
                       'ton_maks_1',
                       'odst_ton_maks_1',
                       'ton_oznaka_1',
                       'interval2',
                       'int_maks_2',
                       'odst_int_maks_2',
                       'int_oznaka_2',
                       'ton_maks_2',
                       'odst_ton_maks_2',
                       'ton_oznaka_2',
                       'trajanje',
                       'trajanje_oznaka'])


class Detektor:
    def __init__(
            self,
            direktorij,
            pocetno_vrijeme_titla,
            pomak,
            odstupanje_int,
            odstupanje_ton,
            odstupanje_trajanje):
        """

        :param direktorij: Direktorij s podacima za analizu segmenta.
        :param pocetno_vrijeme_titla: Pozicija titla na cijeloj snimci
(kako je
            dobiveno iz transkripta).
        :param pomak: Gdje se izdvojeni segment nalazi na cijeloj snimci
(sec).
```

```
            Ovaj parametar je vazan jer su titlovi oznaceni u odnosu na
cijelu
            snimku.
        :param odstupanje_int: Prag odstupanja intenziteta.
        :param odstupanje_ton: Prag odstupanja tona.
        """
        self._direktorij = direktorij
        self._datoteka_intenziteta = 'int_10ms.txt'
        self._datoteka_tona = 'pitch_10ms.txt'
        self._pocetno_vrijeme_titla = pocetno_vrijeme_titla
        self._pomak = pomak
        self.poravnanje = None
        self._odstupanje_int = odstupanje_int
        self._odstupanje_ton = odstupanje_ton
        self._odstupanje_trajanje = odstupanje_trajanje
        self._rezultat_analize = []

    def inicijalizacija(self, tekst_titla, interval_titla):
        """
        Inicijalizacija pocetnih vrijednosti.

        :param tekst_titla:
        :param interval_titla:
        :return: Lista RijecInfo objekata.
        """
        tabela = []
        pauze = self.__intervali_pauza()
        rijeci = tekst_titla.split(' ')
        tekst = tekst_titla.replace(' ', '').lower()
        interval_zvuka = interval_titla[1] - interval_titla[0]
        nastavak = interval_titla[0]

        for interval_rijeci in rijeci:
            rijec = util.PripremaTeksta(interval_rijeci) \
                .obradjen_tekst().lower()
            postotak_teksta = len(rijec) / len(tekst) * 100
            utroseni_dio_zvuka = interval_zvuka * postotak_teksta / 100

            od, do = round(nastavak, 2), round(nastavak +
utroseni_dio_zvuka, 2)
            info = (rijec, od, do)
            interval_rijeci = Detektor.__uklopi_pauze(pauze, info)
            if interval_rijeci != info:
                nastavak = interval_rijeci[2]
            else:
                nastavak += utroseni_dio_zvuka

            tabela += [RijecInfo(rijec=interval_rijeci[0],
                                 glasovi=[],
                                 interval1=interval_rijeci[1:],
                                 interval2=None,
                                 int_maks_1=0,
                                 ton_maks_1=0,
                                 int_maks_2=0,
                                 ton_maks_2=0)]

        return tabela

    def __interval_titla(
            self,
            pozicija_pocetnog_titla,
```

```
            pozicija_segmenta,
            vrijeme_od,
            vrijeme_do):
        """
        S obzirom da je segment zvuka za analizu izvucen iz vece snimke ova
        funkcija izracunava poziciju titla na izdvojenom segmentu na osnovu
        njegove pozicije na cijeloj (vecoj) snimci. Podatke o poziciji
titla
        koji dolaze preko teleteksta NE modificiramo u izvornoj (TS)
datoteci!

        :param pozicija_pocetnog_titla: Pozicija titla na cijeloj snimci.
        :param pozicija_segmenta: Pozicija izdvojenog segmenta zvuka na
cijeloj
                                  snimci (pomak).
        :param vrijeme_od: Pocetno vrijeme titla (iz TS datoteke).
        :param vrijeme_do: Zavrsno vrijeme titla (iz TS datoteke).
        :return: Preracunati interval titla u odnosu na segment snimke.
        """

        def pomak_titla(pozicija_titla: tuple):
            intenziteti = citac.vrijeme_vrijednost_dat(
                    self._direktorij,
                    self._datoteka_intenziteta)
            iznad_praga = [p for p in intenziteti if p[1] > 60]
            pocetak_zvuka = iznad_praga[0][0]
            return (pozicija_titla[0] * 60 + pozicija_titla[1]
                    - pozicija_segmenta - pocetak_zvuka)

        pomak = pomak_titla(pozicija_pocetnog_titla)
        pocetak = (util.sekunde(vrijeme_od[0], vrijeme_od[1]) - pomak
                   - pozicija_segmenta)
        zavrsetak = (util.sekunde(vrijeme_do[0], vrijeme_do[1]) - pomak
                     - pozicija_segmenta)
        # pozicija = (round(pocetak, 2), round(zavrsetak, 2))
        return round(pocetak, 2), round(zavrsetak, 2)

    def __intervali_pauza(self):
        """
        Daje pozicije zvuka s intenzitetom ispod zadanog praga (tisina).

        :return: Lista parova (od, do).
        """
        intenziteti = citac.vrijeme_vrijednost_dat(
                self._direktorij,
                self._datoteka_intenziteta)
        ispod_praga = [p for p in intenziteti if p[1] < 60]  # <60 dB
        pauze = []

        interval = []
        prethodno_vrijeme = round(ispod_praga[0][0], 2)
        for i in range(1, len(ispod_praga)):
            vrijeme = round(ispod_praga[i][0], 2)
            if round(abs(vrijeme - prethodno_vrijeme), 2) <= 0.01:
                interval += [vrijeme]
                prethodno_vrijeme = round(ispod_praga[i][0], 2)
            else:
                if len(interval) > 1 \
                        and interval[-1] - interval[0] >= 0.077:
                    pauze += [(interval[0], interval[-1])]
                interval = []
```

```
                prethodno_vrijeme = round(ispod_praga[i][0], 2)

        return pauze

    @staticmethod
    def __uklopi_pauze(pauze, rijec_info: tuple):
        """
        Dodavanje vremena pauze u trajanje rijeci (zbog preciznijeg
poravnanja).

        :param pauze: Podaci o pozicijama pauza.
        :param rijec_info: Podaci o rijeci za koju treba uklopiti pauze.
        :return: ntorka (rijec, od, do)
        """

        def izmedju(n, interval):
            return interval[0] <= n <= interval[1]

        def unutar(interval_1, interval_2):
            return izmedju(interval_1[0], interval_2) \
                   and izmedju(interval_1[1], interval_2)

        for p in pauze:
            if unutar(p, rijec_info[1:]):
                return (rijec_info[0],
                        rijec_info[1],
                        round(rijec_info[2] + (p[1] - p[0]), 2))

        return rijec_info

    def __postavi_zvucne_podatke(self, rijeci):
        """
        Postavljanje podataka o tonu i intenzitetu za segmente zvuka kojima
        odgovaraju rijeci.

        :param rijeci: Popis objekata RijecInfo.
        :return: Popis objekata RijecInfo.
        """
        popis = []
        for rijec_info in rijeci:
            intenziteti_1 = [n for (v, n) in citac.vrijeme_vrijednost_dat(
                    self._direktorij,
                    'int_10ms.txt',
                    interval=rijec_info.interval1)]
            tonovi_1 = [n for (v, n) in citac.vrijeme_vrijednost_dat(
                    self._direktorij,
                    'pitch_10ms.txt',
                    interval=rijec_info.interval1)]

            intenziteti_2 = [n for (v, n) in citac.vrijeme_vrijednost_dat(
                    self._direktorij,
                    'int_10ms.txt',
                    interval=rijec_info.interval2)]
            tonovi_2 = [n for (v, n) in citac.vrijeme_vrijednost_dat(
                    self._direktorij,
                    'pitch_10ms.txt',
                    interval=rijec_info.interval2)]

            int_maks_1 = max(intenziteti_1) \
                if intenziteti_1 not in (None, []) \
                else None
```

```
            ton_maks_1 = max(tonovi_1) \
                if tonovi_1 not in (None, []) else None
            int_maks_2 = max(intenziteti_2) \
                if intenziteti_2 not in (None, []) else None
            ton_maks_2 = max(tonovi_2) \
                if tonovi_2 not in (None, []) else None
            popis += [RijecInfo(
                    rijec=rijec_info.rijec,
                    glasovi=rijec_info.glasovi,
                    interval1=rijec_info.interval1,
                    interval2=rijec_info.interval2,
                    int_maks_1=int_maks_1,
                    ton_maks_1=ton_maks_1,
                    int_maks_2=int_maks_2,
                    ton_maks_2=ton_maks_2)]

        return popis

    @staticmethod
    def __izdvoji_glasove_rijeci(tabela_glasova, poravnanje):
        """
        Za intervale poravnanja izdvaja klasificirane glasove (da se moze
        mjeriti trajanje).

        :param tabela_glasova:
        :param poravnanje:
        :return: Lista mapa {od, do, grupa_glasova}.
        """

        class Stanje(Enum):
            PRESKOCI = 0
            DODAJ = 1

        glasovi_rijeci = []
        for rijec_info in poravnanje:
            stanje = Stanje.PRESKOCI
            glasovi = []
            for glas_info in tabela_glasova:
                if stanje == Stanje.DODAJ:
                    if glas_info['do'] <= rijec_info.interval1[1]:
                        glasovi += [glas_info['grupa_glasova']]
                    else:
                        glasovi_rijeci += [glasovi]
                        # prekidamo dodavanje glasova i idemo na iducu
rijec
                        break
                else:
                    if glas_info['od'] >= rijec_info.interval1[0]:
                        stanje = Stanje.DODAJ
                        glasovi += [glas_info['grupa_glasova']]

        return glasovi_rijeci

    def __poravnanje(
            self,
            titl,
            vrijeme_titla_od,
            vrijeme_titla_do):
        """
        Postavljanje poravnanja na osnovu klasificiranih glasova.
```

```
        :param titl: Tekst transkripta.
        :param vrijeme_titla_od: Pocetak titla.
        :param vrijeme_titla_do: Zavrsetak titla.
        :return: Lista RijecInfo objekata.
        """
        pozicija_titla = self.__interval_titla(
                self._pocetno_vrijeme_titla,
                self._pomak,
                vrijeme_titla_od,
                vrijeme_titla_do)

        print('INTERVAL TITLA: ', pozicija_titla)
        glasovi = klasifikacija(self._direktorij)
        tabela_glasova = poravnavanje.napravi_tabelu_glasova(
                glasovi,
                pozicija_titla[0] - 0.2,
                pozicija_titla[1] + 0.2)
        # print('=> Glasovi:', tabela_glasova)
        priprema_teksta_obj = util.PripremaTeksta(titl)
        uredjen_tekst = priprema_teksta_obj.obradjen_tekst()
        self.poravnanje, _ = poravnavanje.poravnaj_slova_s_glasovima(
                uredjen_tekst.lower(),
                tabela_glasova)
        intervali_rijeci = poravnavanje.intervali_rijeci(
                priprema_teksta_obj,
                self.poravnanje)
        glasovi_rijeci = Detektor.__izdvoji_glasove_rijeci(tabela_glasova,
                                                           self.poravnanje)

        # postavi intervale rijeci
        rijeci = []
        for i, info in enumerate(self.poravnanje):
            rijeci += [RijecInfo(
                    rijec=info.rijec,
                    glasovi=glasovi_rijeci[i]
                    if i < len(glasovi_rijeci) else [],
                    interval1=info.interval1,
                    interval2=(intervali_rijeci[info.rijec]
                               if info.rijec in intervali_rijeci else (0,
0)),
                    int_maks_1=0,
                    ton_maks_1=0,
                    int_maks_2=0,
                    ton_maks_2=0)]

        return self.__postavi_zvucne_podatke(rijeci)

    def __prosjek_intenziteta_i_tona(self, interval):
        """
        Daje podatke o prosjecnom intenzitetu i tonu za zadani interval
zvuka.

        :param interval: Interval zvuka.
        :return: Par (prosjecni intenzitet, prosjecni ton)
        """
        intz = citac.vrijeme_vrijednost_dat(
                self._direktorij,
                self._datoteka_intenziteta)
        ton = citac.vrijeme_vrijednost_dat(
                self._direktorij,
                self._datoteka_tona)
```

```
        vr_int = [n for (v, n) in intz if interval[0] <= v <= interval[1]]
        vr_ton = [n for (v, n) in ton if interval[0] <= v <= interval[1]]
        pr_int = sum(vr_int) / len(vr_int) if len(vr_int) > 0 else 0
        pr_ton = sum(vr_ton) / len(vr_ton) if len(vr_ton) > 0 else 0
        return round(pr_int, 2), round(pr_ton, 2)

    @staticmethod
    def __trajanja(rijeci):
        """
        Izdvoji najduzi niz vokala i izracunaj odstupanje njegove duljine
od
        ostalih nizova vokala unutar titla ili prozora.

        :param rijeci:
        :return:
        """
        duljine_niza_vokala = []
        for r in rijeci:
            duljine_vokala_za_rijec = []
            for grupa_glasova in r.glasovi:
                if grupa_glasova[0] in 'ieaou':
                    duljine_vokala_za_rijec += [len(grupa_glasova)]
            duljine_niza_vokala.append(duljine_vokala_za_rijec)

        prosjek_trajanja_rijeci = [sum(v) / len(v) if len(v) > 0 else -1
                                   for v in duljine_niza_vokala]
        prosjek_trajanja_za_titl = sum(prosjek_trajanja_rijeci) /
len(rijeci)
        return prosjek_trajanja_za_titl, prosjek_trajanja_rijeci

    def __analiza(self, titl, od, do):
        """
        Analiza zvuka za svaku rijec gdje je interval rijeci dobiven
prethodnim
        poravnanjem.

        :param titl: Tekst transkripta.
        :param od: Pocetak titla.
        :param do: Zavrsetak titla.
        :return: self
        """
        rijeci = self.__poravnanje(titl, od, do)  # lista RijecInfo
objekata
        # print('==> Tabela:', rijeci)

        # odredi prosjek intenziteta i tona za cijeli titl
        pr_int, pr_ton = self.__prosjek_intenziteta_i_tona(
                (rijeci[0].interval1[0], rijeci[-1].interval1[1]))

        # odredi prosjek trajanja
        pr_trajanja_titl, pr_trajanja_rijeci = Detektor.__trajanja(rijeci)
        print(f'PROSJEK INT.: {pr_int:<10}PROSJEK TON: {pr_ton:<10}'
              f'PROSJEK TRAJANJE: {str(round(pr_trajanja_titl, 2)):<10}')

        self._rezultat_analize = []
        for i in range(len(rijeci)):
            int_maks_1 = rijeci[i].int_maks_1
            ton_maks_1 = rijeci[i].ton_maks_1
            int_maks_2 = rijeci[i].int_maks_2
            ton_maks_2 = rijeci[i].ton_maks_2
```

```
            if int_maks_1 is None or ton_maks_1 is None:
                odst_int_maks_1 = 0
                odst_ton_maks_1 = 0
            else:
                odst_int_maks_1 = round((int_maks_1 / pr_int - 1) * 100, 2)
                odst_ton_maks_1 = round((ton_maks_1 / pr_ton - 1) * 100, 2)

            if int_maks_2 is None or ton_maks_2 is None:
                odst_int_maks_2 = 0
                odst_ton_maks_2 = 0
            else:
                odst_int_maks_2 = round((int_maks_2 / pr_int - 1) * 100, 2)
                odst_ton_maks_2 = round((ton_maks_2 / pr_ton - 1) * 100, 2)

            int_oznaka_1 = ''
            if odst_int_maks_1 > self._odstupanje_int:
                int_oznaka_1 = '*'

            int_oznaka_2 = ''
            if odst_int_maks_2 > self._odstupanje_int:
                int_oznaka_2 = '*'

            ton_oznaka_1 = ''
            if odst_ton_maks_1 > self._odstupanje_ton:
                ton_oznaka_1 = '*'

            ton_oznaka_2 = ''
            if odst_ton_maks_2 > self._odstupanje_ton:
                ton_oznaka_2 = '*'

            trajanje = round((pr_trajanja_rijeci[i] - pr_trajanja_titl)
                             / pr_trajanja_titl * 100, 2)
            trajanje_oznaka = ('*'
                               if trajanje > self._odstupanje_trajanje else
''')

            rezultat = Rezultat(
                    rijec=rijeci[i].rijec,
                    interval1=rijeci[i].interval1,
                    int_maks_1=int_maks_1,
                    odst_int_maks_1=odst_int_maks_1,
                    int_oznaka_1=int_oznaka_1,
                    ton_maks_1=ton_maks_1,
                    odst_ton_maks_1=odst_ton_maks_1,
                    ton_oznaka_1=ton_oznaka_1,
                    interval2=rijeci[i].interval2,
                    int_maks_2=int_maks_2,
                    odst_int_maks_2=odst_int_maks_2,
                    int_oznaka_2=int_oznaka_2,
                    ton_maks_2=ton_maks_2,
                    odst_ton_maks_2=odst_ton_maks_2,
                    ton_oznaka_2=ton_oznaka_2,
                    trajanje=trajanje,
                    trajanje_oznaka=trajanje_oznaka)
            self._rezultat_analize += [rezultat]

        return self

    def __formatiraj_rezultat(self):
        """
        Prikaz rezultata tabularno i SSML.
```

```
        :return: None
        """
        ssml = '<speak>'
        print(f'{"RIJEC":15}'
              f'{"INTERVAL":15}'
              f'{"MI (dB)":13}'
              f'{"Odst. MI":12}'
              f'{"Ist. int.":12}'
              f'{"MT (Hz)":13}'
              f'{"Odst. MT":12}'
              f'{"Ist. ton":12}'
              f'{"Ist. trajanje":12}')
        for rijec_info in self._rezultat_analize:
            print(f'{rijec_info.rijec:15}'
                  f'{str(rijec_info.interval1):15}'
                  f'{str(rijec_info.int_maks_1):13}'
                  f'{str(rijec_info.odst_int_maks_1) + " %":12}'
                  f'{str(rijec_info.int_oznaka_1):^12}'
                  f'{str(rijec_info.ton_maks_1):13}'
                  f'{str(rijec_info.odst_ton_maks_1) + " %":12}'
                  f'{str(rijec_info.ton_oznaka_1):^12}'
                  f'{str(rijec_info.trajanje_oznaka):^12}')

            dodaj_prozodiju = False
            prozodija = '<prosody '
            if rijec_info.int_oznaka_1 == '*':
                # 6 je maksimum (prema pravilima SSMLa)
                db = round(6 * (rijec_info.odst_int_maks_1 / 100), 2)
                prozodija += 'volume=+' + str(db) + 'dB '
                dodaj_prozodiju = True

            if rijec_info.ton_oznaka_1 == '*':
                prozodija += 'pitch=+' + str(rijec_info.odst_ton_maks_1) +
'%'
                dodaj_prozodiju = True

            prozodija += '> ' + rijec_info.rijec + ' </prosody> '

            if dodaj_prozodiju:
                ssml += prozodija
            else:
                ssml += rijec_info.rijec + ' '

        ssml += '</speak>'
        print('\nSSML:', ssml, '\n')

    def prikazi(self):
        """
        Glavna metoda za prikaz rezultata automatske detekcije naglasenih
        rijeci.

        :return: None
        """
        titlovi = citac.Titlovi(self._direktorij)
        popis_titlova = titlovi.ucitaj()
        for titl in popis_titlova:
            rezultat = self.__analiza(titl.titl, titl.vrijeme_od,
                                      titl.vrijeme_do)
            rezultat.__formatiraj_rezultat()
```

```
#
================================================================================
===


def klasifikacija(putanja):
    """
    Ovo se koristi samo ako treba nanovo klasificirati glasove. Ako
datoteka
    s klasificiranim glasovima vec postoji onda ova funkcija vraca popis
    tih glasova, inace se glasovi nanovo klasificiraju i taj se popis
glasova
    vraca.
    :param putanja: Putanja gdje treba smjestiti datoteku s klasificiranim
                    glasovima.
    :return: Klasificirane glasove
    """
    datoteka = util.konfig('datoteka', 'dir') + putanja + '/glasovi.pyobj'
    if os.path.exists(datoteka):
        with open(datoteka, 'rb') as f:
            glasovi = pickle.load(f)
            return glasovi
    else:
        glasovi = nm.klasificiraj_glasove(putanja)
        with open(datoteka, 'wb') as f:
            pickle.dump(glasovi, f, pickle.HIGHEST_PROTOCOL)
        return glasovi


def proba():
    odst_int = 20  # odstupanje intenziteta vece od ovog postotka je nagl.
rijec
    odst_ton = 25  # odstupanje tona vece od ovog postotka je nagl. rijec
    odst_trajanje = 250  # odstupanje trajanja od ovog postotka je nagl.
rijec

    info = [
        # ('/detekcija/test', (0, 0), 0),

        ('/detekcija/proba_1_m1', (7, 39.44), 459),
        # ('/detekcija/proba_2_m1', (7, 48.44), 467.7),
        # ('/detekcija/proba_5_m2', (36, 7.68), 2166.85),
        # ('/detekcija/proba_12_m4', (40, 20.04), 2418.56),
        # ('/detekcija/proba_3_z1', (7, 27.32), 446.7),
        # ('/detekcija/proba_8_z4', (10, 13.08), 612.5),
        # ('/detekcija/proba_10_z5', (12, 0), 719.2),
        # ('/detekcija/proba_11_z5', (11, 42.36), 701.56),
    ]

    for p in info:
        print('=' * 120, '\n***** DIREKTORIJ:', p[0])
        Detektor(*p, odst_int, odst_ton, odst_trajanje).prikazi()


if __name__ == '__main__':
    proba()
```

Životopis

Aleksandar Stojanović rođen je 1969. godine u Zagrebu. Godine 1996. diplomirao je Informacijske znanosti i fonetiku na Filozofskom fakultetu Sveučilišta u Zagrebu, a 1999. godine magistrirao iz računarskih znanosti u SADu na sveučilištu Midwestern State University. Nakon završetka školovanja radio je kao programski inženjer na razvoju financijskih aplikacija te aplikacija za telekomunikacijsku i energetsku industriju.

Godine 2015. izabran je u zvanje asistenta na Tehničkom veleučilištu u Zagrebu, gdje sudjeluje u nastavi iz predmeta "Objektno-orijentirano programiranje I", "Objektno-orijentirano programiranje II", "Napredno programiranje u jeziku Python", "Programsko inženjerstvo i informacijski sustavi" i "Programsko inženjerstvo u otvorenim sustavima". Na svojoj matičnoj ustanovi preuzima rad na sustavu za prijavu upisa na specijalističke stručne studije, gdje radi na razvoju web-aplikacije i baze podataka. Prethodno sudjeluje na projektu HKO na razvoju informacijskog sustava za praćenje, pretraživanje i analizu kompetencija i stručnih kvalifikacija, te na projektu za razvoj informacijskog sustava za Zavod za hitnu medicinu grada Zagreba.

Popis objavljenih radova

- Stojanović, A., & Lazić, N. – A Method for Estimating Variations in Speech Tempo from Recorded Speech. MIPRO 2019 (prihvaćeno za objavu).
- Aleksandar Stojanović, Nikolaj Lazić, Željko Kovačević – Tehnologije integracije informacijskih sustava, MIPRO 2016, 39th international convention on information and communication technology, electronics and microelectronics, 783 – 787, ISSN 1847-3946.
- Željko Kovačević, Miroslav Slamić, Aleksandar Stojanović – Razvoj mobilnih aplikacija u Embarcadero alatima, CASE 28 konferencija (2016), 65 – 70, ISSN 1334-448X.
- Aleksandar Stojanović - Osvrt na NoSQL baze podataka - četiri osnovne tehnologije, Polytechnic & Design (1849-1995) 4 (2016), 1; 44-53, ISSN 1849-1995.
- Nikolaj Lazić, Aleksandar Stojanović - Grafički prikaz formanata, Istraživanja govora: Knjiga sažetaka / Speech Research: Book of Abstracts / Lazić, Nikolaj; Pletikos Olof,

Elenmari (ur.). - Zagreb: Hrvatsko filološko društvo, 2016. 63-63, ISBN 9789532961256.

- Aleksandar Stojanović - Elementi računalnih programa s primjerima u Pythonu i Scali, Element 2012, ISBN 978-953-197-616-9.