# Automatsko prepoznavanje govora

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Vladimir Vuksanović, Aleksa Kojadinović, Lazar Čeliković
kontakt email prvog, drugog, trećeg autora

18. novembar 2021.

**Sažetak**

Tema ovog rada je da priblizi citaoca zadatku automatskog prepoznavanja govora, problemima koji se javljaju i najznacajnijim arhitekturama ovih sistema. U prva dva poglavlja bavimo se definisanjem osnovnih pojmova, kratkim pregledom istorije i otezavajucim faktorima. Trece i cetvrto poglavlje prikazuje dve razlicite metodologije prepoznavanja govora i relevantne modele. Peto poglavlje opisuje nacin evaluacije prethodno navedenih kao i drugih modela.

**Ključne reči:** prepoznavanje govora

## Sadržaj

# 1 Uvod

Govor je za ljude najintuitivniji i prirodniji nacin komunikacije. Zbog toga je od samog nastanka kompjutera, nastala i ideja da koristimo isti nacin komunikacije da interagujemo sa njima. To bi znatno smanjilo potrebno predznanje za koriscenje kompjutera i ucnilio ga pristupacnijim vecem broju ljudi. Najveca prepreka ovim sistemima do skoro je bio kako sa velikom tacnosti prepoznati sta je korisnik rekao. Taj postupak se naziva automatsko prepoznavanje govora.

**Automatsko prepoznavanje govora** (eng. *Automatic Speech Recognition, ASR*) je proces pretvaranja zvučnog signala govora u sekvencu reči pomoću kompjutera. Neke od najznacajnijih primena ovih sistema su: pametni licni asistenti (Google Assistant[1], Apple Siri[2],...), transkripcija snimaka, pretrazivanje audio sadrzaja i pristupacnost.

Iako su istrazivanja na ovu temu pocela jos sredinom dvadesetog veka, popularnost je pocela da dobija tek u poslednjoj deceniji kada je uvodjenje dubokih neuronskih mreza drasticno povecalo performanse ovih sistema. Ta razlika je bila dovoljna da ucini ove sisteme prakticno primenljivim umesto nezgodnim za upotrebu zbog velikog broja gresaka. Jedan od najznacajnijih postignuca je ostvareno 2016. godine je kompanija Majkrosoft napravila sistem koji je ostvario iste rezultate kao ljudski eksperti na transkripciji Switchboard skupa podataka [1]. Za glavne uzroke ovog naglog poboljsanja se smatraju [2]:

1. Sakupljanje velike kolicine tanskribovanih skupova podataka
2. Nagli porast u performansama grafickih procesorskih jedinica (GPU)
3. Poboljsanje algoritama za ucenje i arhitektura modela

U nastavku cemo prvo navesti neke izazove koje treba da resimo da bi smo napravili dobar sistem za prepoznavanje govora, zatim cemo opisati nacin rada dva najpopularnija modela: statisticki i end-to-end i na kraju cemo predstaviti nacin za njihovu evaluaciju.

# 2 Izazovi

Prepoznavanje govora je veoma težak zadatak zato što je potrebno da radi podjednako dobro u veoma različitim uslovima. Neki od najvećih izazova su:

- **Mala količina podataka za trening** — Za ostvarivanje dobrih rezultata potrebno je sakupiti više stotina ili čak hiljada sati labeliranih zvučnih snimaka koji treba da sadrže više govornika razlicitog pola i starosti, koji govore različitim akcentima. Dok u skorije vreme jeste nastao porast u količini dostupnih podataka, veliki problem još uvek predstavlja reprezentativnost različitih varijacija u govoru i nedostatak podataka za jezike sa manjim brojem govornika. Zbog toga se istražuju alternativni načini za treniranje kao sto su samo-treniranje (eng. *self-training*) [3], iterativno treniranje [4] ili treniranje koristeći kompjuterski generisan glas [5]. U dodatku A se može naći tabela sa pregledom nekih od najpopularnijih trening skupova na engleskom jeziku.

[1] https://assistant.google.com/
[2] https://www.apple.com/siri/

- **Stil govora** — Postoje različiti sistemi u zavisnosti od toga koji tip govora mogu da prepoznaju [6]. Tipovi govora poređani po težini propoznavanja su:
  1. Izolovane reči — reči su razdvojene dugim periodima tišine
  2. Povezane reči — reči su razdvojene kratkim pauzama
  3. Neprekidan govor — uvežbani govor, čitanje ili diktiranje
  4. Spontani govor — neuvežbani, prirodni govor

  Prve implementacije prepoznavanja govora su radile na nivou izolovanih reči i koristile su se za prepoznavanje određenih komandi ili cifara. Danas se najviše truda ulaže u poboljšanje prepoznavanja neprekidnog i spontanog govora.
- **Karakteristike govornika** — Svaki čovek ima različitu boju glasa i govori različitom brzinom. Čak i starost osobe i jačina govora bitno utiču na frekvenciju glasa. Poseban problem pravi postojanje različitih dijalekata i akcenata koji mogu da imaju potpuno različite načine za izgovaranje istih reči. Jedan način za rešavanje ovog problema je treniranje sistema na glasu govornika koji će ga koristiti. To su sistemi zavisni od korisnika (eng. *speaker dependent*) i koriste se u slučajevima da samo jedna osoba treba da ih koristi. Sa druge strane postoje sistemi nezavisni od korisnika (eng. *speaker independent*) koji treba da rade podjednako dobro za sve govornike.
- **Okruženje govornika** — Ovi sistemi će retko biti korišćeni u potpuno tihim prostorijama sa profesionalnom opremom za snimanje. Zbog toga treba da budu tolerantni na različite vrste pozadinske buke ili kvaliteta mikrofona. Neke vrste šumova je moguće otkloniti analizom zvuka ili naprednijim metodama [7], ali jedan od najvećih problema predstavlja postojanje drugih govornika u okolini. Te signale je često teško razlikovati od glasa primarnig govornika, i samim tim teško ukloniti.
- **Veličina rečnika** — Povećanje broja reči koje model može da prepozna takođe povećava njegovu složenost i otežava treniranje, ali se time dobija na tačnosti. Zbog toga je potrebno naći dobar kompromis između veličine rečnika i složenosti modela. U slučajevima kada je potrebno pouzdano prepoznati samo neki skup komandi koriste se mali rečnici i oni su često veoma pouzdani, ali za prepoznavanje opšteg govora današnji sistemi su trenirani na skupu od oko 50.000-100.000 reči.

# 3 Statisticki model

Dugo vremena statisticki pristup je bio dominantan za sisteme za prepoznavanje govora. Iako je u skorije vreme pao u senku modela zasnovanih na dubokim neuronskim mrezama opisanim u poglavlju 4 ovaj model je jos uvek u sirokoj upotrebi i veoma vredan izucavanja.

Cilj ovih sistema je da pronadju najverovatniju transkripciju za zadati ulaz. Formalno, neka je $\hat{W}$ optimalan niz reci za transkripciju nekog zvucnog signala $X$. Cilj je optimizovati formulu [8]:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X)$$

primenom Bajesove formule to mozemo da zapisemo kao:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)}$$

a kako je $P(X)$ konstantno za konkretan ulaz, mozemo da ga eliminisemo:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(X|W)P(W) \tag{1}$$

Ideja je da umesto modeliranja $P(W|X)$ sto je tesko, odvojeno modeliramo verovatnoce iz prethodne formule za koje imamo bolje tehnike.

Najprirodniji nacin za racunanje $P(X|W)$ bi bio da se podeli na reci i za svaku od njih racuna verovatnoca da je izgovorena. U nekim slucajevima, kao na primer kada treba prepoznati samo neki mali skup komandi, tada se i koristi ovaj pristup. Problem nastaje u sistemima sa velikim vokabularom zato sto postoji veliki broj varijacija u izgovoru reci, a trening skup sadrzi mozda par primera za svaku od njih sto nije dovoljno da se dobro nauci njeno prepoznavanje. Dakle, umesto na reci potrebna je finija podela i za to potrebu cemo koristiti foneme. **Foneme** su najmanje jezicke jedinice na osnovu kojih mogu da se razlikuju znacenja vecih jedinica. One postoje samo kao apstraktna ideja a njihova fizicka realizacija se zove glas.

Ako je $S$ niz fonema, $P(X|W)$ iz formule 1 se razlaze i dobija se:

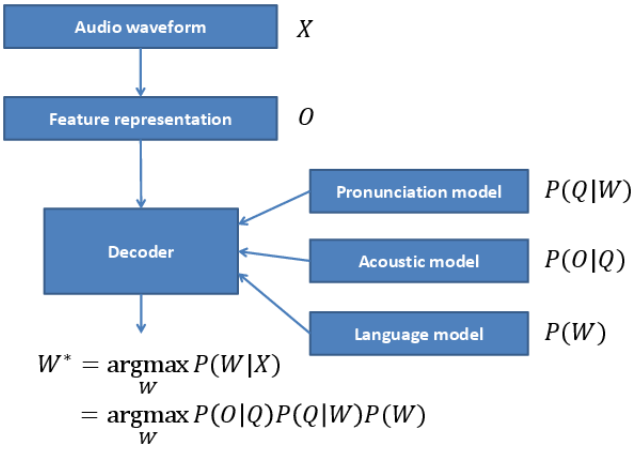
$$\hat{W} = \underset{W}{\operatorname{argmax}} \sum_{S} P(X, S|W)P(W)$$

sto se moze aproksimirati kao:

$$\hat{W} \approx \underset{W,S}{\operatorname{argmax}} P(X|S)P(S|W)P(W) \tag{2}$$

Veličine iz prethodne formule imaju svoja imena na osnovu komponente koja ih računa: $P(X|S)$ se naziva **akusticki model** (eng. *acoustic model*), $P(S|W)$ je **model izgovora** (eng. *pronunciation model*), a $P(W)$ se zove **jezički model** (eng. *language model*)[3].

Na slici 1 je prikazana cela struktura statistickog modela od zvucnog signala do transkripcije:




Slika 1: Statisticki model

U nastavku će biti opisana svaka od prikazanih komponenti, njena uloga i način rada.

[3]Neka literatura $P(X|W)$ naziva akustičkim modelom, a model izgovora tretira kao njegov deo. U ovom radu će biti podrazumevano da su oni odvojeni modeli.

## 3.1 Obrada zvučnog signala

Sirovi zvučni signal je veoma nepogodan za korišćenje zato što sadrži veliku količinu nebitnih informacija i šuma. Zbog toga se, pre prosleđivanja akustičkom modelu, signal prvo obrađuje tako da ostanu samo ključne karakteristike i smanje šum i veličina reprezentacije.

Signal se deli na kratke segmente koji se zovu **okviri** (eng. *frame*). Svaki od njih je fiksne duzine (obično 10-30 milisekundi) sa kratkim preklapanjem sa susednim okvirima radi smanjenja naglih promena prilikom prelaska iz jednog u drugi. Pretpostavka je da je u svakom okviru glas konstantan, to jest da se glasovi mogu menjati samo prelaskom iz jednog okvira u drugi. Na svaki od tih novodobijenih delova se zatim primenjuje neka vrsta spektralne analize najčešće zasnovana na Furijeovoj transformaciji kojom se izdvajaju samo njegove najbitnije karakteristike. Tačna reprezentacija koja se koristi varira u zavisnosti od modela, ali jedna od najpopularnijih je **MFCC** (Mel-Frequency Cepstral Coefficients) [9] zasnovana na Mel skali koja oponaša ljudski slušni sistem. Ovako obrađen signal se prosleđuje akustičkom modelu.

## 3.2 Akustički model

Akustički model je zadužen da pretvori obrađeni zvučni signal u niz fonema. Ovaj zadatak predstavlja idealan slučaj za primenu skrivenih Markovljevih modela [10].

**Skriveni Markovljev model** (eng. *hidden Markov model*) je dinamički sistem kojeg karakteriše sledeće:

1. $N$ skrivenih stanja $S = \{S_1, S_2, \ldots, S_N\}$ pri cemu $q_t$ označava stanje u trenutku $t$
2. $M$ obzervacionih simbola $V = \{V_1, V_2, \ldots, V_M\}$ pri cemu $o_t$ označava obzervaciju u trenutku $t$
3. Raspodela verovatnoća promene stanja predstavljena matricom $A = \{a_{ij}\}$ dimenzije $N \times N$ gde važi:

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \quad 1 \leq i, j \leq N$$

4. Raspodela verovatnoća obzervacionih simbola iz stanja $j$ predstavljena matricom $B = \{b_j(k)\}$ dimenzije $N \times M$ gde važi:

$$b_j(k) = P(o_t = V_k | q_t = S_j) \quad 1 \leq j \leq N, 1 \leq k \leq M$$

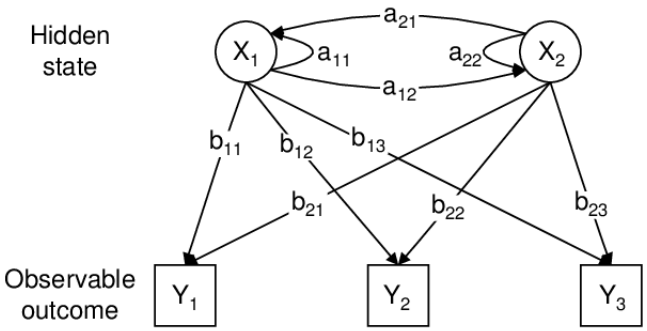
5. Raspodela inicijalnog stanja $\pi = \{\pi_i\}$ gde važi:

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N$$

Primer jednog modela je prikazan na slici 2.

Konkretno za prepoznavanje govora, skriveni Markovljevi modeli se koriste za opisivanje svake foneme. Svaki od njih se sastoji od nekog zadatog broja stanja (obicno 3 ili 5), a obzervacije su okviri zvučnog signala cije su verovatnoće predstavljene kao mešavina Gausovih raspodela. Raspodele verovatnoća su dobijene treniranjem modela na skupu podataka. Za svaku rečenicu iz trening skupa se konstruiše model spajanjem modela fonema u reči, pa reči u rečenicu. Na njega se onda primeni Baum-Welch algoritam koji popravlja verovatnoće da više odgovaraju datim podacima.

Istrenirani model posle moze da predvidi najverovatniji niz fonema (stanja) za datu obzevaciju koristeci Vetrebi algoritam za pretragu.




Slika 2: Primer skrivenog Markovljevog modela sa 2 stanja i 3 obzervaciona simbola

## 3.3 Model izgovora

U prethodnoj sekciji je rečeno da se prilikom treniranja modeli fonema spajaju u modele reči, ali nije objašnjeno tačno kako se to radi. To je baš zadatak modela izgovora. On je u suštini veliki rečnik koji za svaku reč čuva niz fonema kako se ona izgovara. Ako postoji više varijacija izgovora, one se smatraju kao ražlicite stavke u rečniku. Sada konstrukcija modela reči postaje trivijalna, samo se pronađe njen izgovor u rečniku i nadovežu odgovarajuci modeli fonema. Ukoliko se reč ne nalazi u rečniku, sistem za prepoznavanje govora neće biti u stanju da je prepozna.

Prirodno sledeće pitanje je kako se određuju ova preslikavanja. Ovo je zapravo jedan od najtežih zadataka za modeliranje zato što se on ne uči na skupu podataka nego ga konstruišu eksperti iz tog domena. Za svaku reč, neko je morao da zapiše na koji način se izgovara uzimajući u obzir da potencijalno postoji više izgovora. Sa obzirom da današnji sistemi razlikuju oko 100.000 reči, ovo nimalo nije lak posao.

## 3.4 Jezički model

Povratkom na formulu 2, jezički model dodeljuje verovatnoću pojavljivanja $P(W)$ svakoj mogućoj sekvenci reči $W$. Ovde se uzima u obzir relativna učestalost reči, verovatnoća da se reči nađu jedna za drugom, i mogu da se vrše dodatne sintaksne i semantičke provere. Postoje rečenice koje zvuče slično ali nemaju sve semantičko znacenje. Tada se od njih bira ona koja ima najviše smisla.

Kako $P(W)$ ne zavisi od zvucnog signala, moze se odvojeno trenirati na samo tekstualnom skupu podataka kojih postoji dosta vise i imaju veci broj primera od skupova sa transkribovanim snimcima. U nekim slucajevima trenirana raspodela se moze promeniti u zavisnosti od korisnika (npr. pametni asistenti prepoznaju kontakte na korisnikovom telefonu).

Najvesci vid implementacije ovog modela je pomocu n-grama. Neka se $W$ razdvaja na reci $W = \{w_1, w_2, \ldots, w_m\}$ i neka je $n$ duzina n-grama. To znaci da pri racunanju verovatnoce pojavljivanja neke reci u obzir uzimamo samo $n-1$ njenih prethodnika a za ostale pretpostavljamo da

ne uticu. Tada se $P(W)$ moze predstaviti kao:

$$P(W) = \prod_{i=1}^{m} P(w_i|w_1, \ldots, w_{i-1}) = \prod_{i=1}^{m} P(w_i|w_{i-n+1}, \ldots, w_{i-1})$$

ako sa $C(x)$ oznacimo broj pojavljivanja sekvence $x$ u trening skupu tada se verovatnoca odredjene reci moze proceniti kao udeo pojavljivanja neke sekvence u broju pojavljivanja njenog prefiksa:

$$P(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})}$$

Radi jednostavne implementacije na pocetak $W$ se dodaje $n-1$ "prazna"rec.

Sledeci problem je kako odrediti broj $n$. On ne sme da bude preveliki zato sto se time smanjuje sansa da se neka kombinacija reci te duzine uopste nasla u trening skupu. Cak i za male vrednosti moguce je da se neka sekvenca nije ranije videla. Taj problem se resava smanjenjem $n$ ukoliko ne postoji ta sekvenca duzine $n$ pa koriscenjem te verovatoce ili nekim postupkov ugladjivanja.

### 3.5 Dekodiranje

Konstrukcijom svih prethodno opisanih komponenti i njihovim treniranjem, model je gotov i spreman za upotrebu. Jedino sto ostaje je pretraziti prostor dopustivih recenica da se pronadje ona koja najvise odgovara glasovnom signalu. U praksi, taj prostor je veoma veliki i njegova pretraga je eksponencijalne slozenosti (svaka moguca kombinacija reci u recenici) stoga nije izvodljivo traziti egzaktno resenje. Umesto toga se koristi heuristicki beam search algoritam. Ideja je da se resenje gradi iterativno i u svakom trenutku umesto testiranja svih mogucih puteva biramo $b$ najverovatnijih puteva. Parametar $b$ se bira tako da balansira velicinu prostora za pretrazivanje i vreme potrebno za njegov obilazak. Algoritam je dakle sledeci: odredimo verovatnocu za svaku rec da bude prva pa od njih izaberemo $b$ najverovatnijih. U sledecem koraku svaki od tih $b$ reci produzujemo sledecom i od njih ponovo biramo $k$ najverovatnijih. Ovaj postupak se ponavlja sve dok ne dodjemo do kraja recenice i ona predstavlja konacnu transkripciju govora.

## 4 End-to-end modeli

Za razliku od statistickog modela koji proces prepoznavanja govora deli u vise slozenijih celina, end-to-end modeli su po strukturi prostiji. Bave se direktnim prevodjenjem ulaznog zvucnog signala u niz grafema, karatera ili reci. U nastavku prikazujemo dva najreprezentativnija modela: CTC model [11] i LAS model [12] [13]

### 4.1 CTC model

Neka je ulazni zvuk uzorkovan u proizvoljnom broju jednakih vremenskih intervala, gde je svaki od njih predstavljen vektorom realnih brojeva duzine $m$. Neka je $L$ konacna azbuka labela (oznaka). Cilj je napraviti preslikavanje $h$ koje slika proizvoljan zvucni signal u niz labela:

$$h : (\mathbb{R}^m)^* \to L^* \tag{3}$$

U praksi fiksiramo broj vremenskih trenutaka na neku vrednost $T$. Iako je broj trenutaka fiksiran za zvucne signale, nizovi labela ne moraju

biti iste duzine za svaku ulaznu instancu, stoga jednu trening instancu predstavlja par $(\mathbf{x}, \mathbf{z})$ gde je $z$ vektor labela duzine najvise $T$. Ako bismo test skup oznacili sa O, tada funkciju greske mozemo definisati na sledeci nacin:

$$LER(h, O) = \frac{1}{|O|} \sum_{(\mathbf{x},\mathbf{z})\in O} \frac{ED(h(\mathbf{x}), \mathbf{z})}{|\mathbf{z}|} \tag{4}$$

$ED$ predstavlja edit distancu (minimalni broj izmena koji dovodi jedan niz karaktera do drugog, pri cemu dozvoljene izmene podrazumevaju brisanje, supstituciju i umetanje karaktera). Prethodna mera naziva se stopa greske labela (*eng. label error rate* - $LER$).

Formula 4 jeste prirodna ocena greske za probleme koji za cilj imaju minimizaciju greske prevodjenja.

Koristeci sva prethodna razmatranja konstruisemo rekurentnu neuronsku mrezu koja na ulazu ima $mT$ ulaza, dok se na izlazu dobija $T$ vektora dimenzija $L' = L \cup \{\epsilon\}$ pri cemu svaki predstavlja raspodelu verovatnoca oznaka za svaki trenutak prosirujuci azbuku blanko labelom $\epsilon$.

Neformalno receno, prolaskom kroz izlazne softmax nizove dobijamo putanju $\pi \in (L')^T$ koja predstavlja jedan moguci odabir labela. Ako $y^t_{\pi_t}$ predstavlja softmax vrednost $t$-tog trenutka oznake $\pi_t$ tada verovatnocu odabira kompletne putanje dobijamo kao proizvod:

$$p(\pi|x) = \prod_{t=1}^{T} y^t_{\pi_t}$$

U praksi uzorkovanje zvuka se vrsi u veoma sitnim vremenskim intervalima (oko 10ms). Stoga je pojava blanko ili dupliciranih oznaka veoma cesta. Iz tog razloga uvodimo preslikavanje $\beta$ cija je uloga preciscavanje nizova labela uklanjanjem blanko oznaka i susednih duplikata.

$$\beta : (L')^T \to L^U, U \leq T$$

Primetimo da za jednu preciscenu putanju $l$ moze postojati vise mogucih izvornih putanja, pa je verovatnoca njenog odabira jednaka sumi po svim izvornim putanjama.

$$p(l|x) = \sum_{\pi\in\beta^{-1}(l)} p(\pi|x) \tag{5}$$

Imajuci u vidu sve prethodno navedeno, zadatak preslikavanja $h$ je odabir najverovatnije preciscene putanje za dati ulaz.

$$h(x) = \underset{l\in L^U}{\operatorname{argmax}}\, p(l|x) \tag{6}$$

Najjednostavniji algoritam jeste pohlepni odabir najbolje oznake za svaki vremenski trenutak ponaosob. Medjutim, ovakav pristup ne garantuje optimalnost. Naravno, postoje bolji algoritmi za resavanje datog problema. Isti nece biti obradjivani u ovom radu, ali se mogu naci u [].

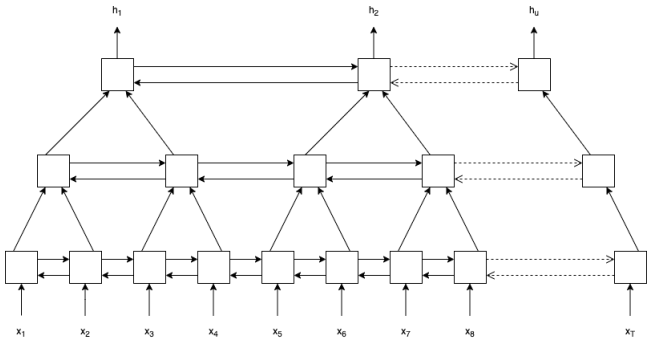
## 4.2 Modeli zasnovani na paznji

Ovu grupu modela analiziramo na osnovu LAS (eng. Listen Attend and Spell) modela. Glavna razlika izmedju CTC modela i LAS modela ogleda se u odbacivanju pretpostavke o nezavisnosti u okviru niza oznaka.

Glavne komponente ovog modola jesu Listener i Speller. Okvirno gledano uloga listener komponente jeste da transformise ulazni signal u predefinisane karakteristike viseg nivoa (labele). Takav izlaz prosledjuje se Speller komponenti koja konacno daje niz karaktera.

Slicno kao u poglavlju 4.1 sa $\mathbf{x} = (x_1, x_2, ..., x_T)$ oznacavamo ulazni signal. $\mathbf{y}$ predstavlja niz mogucih karaktera ukljucujuci razmak, tacku, zarez, apostrof, i tri specijalna karaktera (start - pocetak recenice, stop - kraj recenice, blank - blanko karakter).

Listener jeste rekurentna neuronska mreza piramidalne strukture. Kao sto se moze primetiti na slici iznad broj neurona se u pravcu izlaznog sloja polovi. Ova osobina nam omogucava smanjenje slozenosti izracunavanja, sto je veoma pozeljno u slucaju kada je interval uzorkovanja veoma uzak.




Slika 3: Listener model

Cilj je modelovati $\mathbf{y}$ kao uslovnu raspodelu u zavisnosti od ulaznog signala $\mathbf{x}$ i prethodnih izlaza $y_j, j \in (1, i-1)$

$$P(y|x) = \prod_i P(y_i|x, y_j, j < i) \tag{7}$$

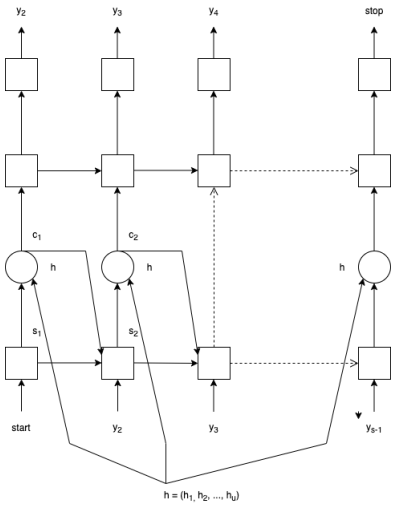
Speller je takodje rekurentna neuronska mreza. Kao sto je vec pomenuto verovatnoca odabira svakog karaktera zavisi od odabira svih prethodnih. Na izlaz $y_i$ utice stanje $s_i$ i kontekst $s_i$. Stanje se racuna rekurentno na osnovu prethodnog stanja, izlaza i konteksta.

$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1}) \tag{8}$$

Kontekst se izracunava koriscenjem standardnog mehanizma paznje (*eng. Attention mechanism*). Intuitivno, kontekst prikuplja relevantno znanje o okolnom ulaznom signalu potrebno za predikciju narednog karaktera, na taj nacin odredjivajuci koliko znanje o okolini utice na znanje o trenutnom karakteru.

Treniranje se sprovodi odredjivanjem parametara koji maksimizaciju sumu uslovnih log verovatnoca za svaki karakter.

$$\max_\theta \sum_i \log P(y_i|x, \bar{y}_j, j < i; \theta) \tag{9}$$




Slika 4: Speller model

gde je $\bar{y_j}$ objektivna vrednost karaktera $j$.

Finalno dekodiranje vrsi se algoritmom beam pretrage (*eng. beam search*). U koraku prosirivanja trenutnog cvora pretrage bira se $\beta$ najverovatnijih karaktera po verovatnoci dobijenoj na izlazu Speller komponente, gde je $\beta$ parametar beam search algoritma.

# 5 Metrike za evaluaciju

Standardna mera za procenu kvaliteta sistema za prepoznavanje govora je **stopa pogrešnih reci** (eng. *word error rate, WER*). Za njeno izračunavanje potrebno je da imamo generisan tekst koji evaluiramo i tačnu transkripciju kao referencu. Formula je izvedena normalizacijom Levenštajnovog rastojanja i izgelda ovako:

$$WER = \frac{I + D + S}{N}$$

gde je:

- $I$ broj umetnutih reči
- $D$ broj obrisanih reči
- $S$ broj zamenjenih reči
- $N$ ukupan broj reči u referenci

Minimalna vrednost koju može da ima je 0, dok maksimalna vrednost može da bude preko 1 (npr. izgovorena je jedna reč a prepoznate dve). Stopa pogrešnih reči se efikasno računa dinamičkim programiranjem, pomoću Vagner-Fišerovog algoritma (eng. *Wagner-Fischer algorithm*).

Cilj sistema za prepoznavanje govora je da minimizuje ovu vrednost.

# 6 Zaključak

Ovim radom obradili smo najznacajnije i najuspesnije modele za automatsko prepoznavanje govora. Uprkos tome sto su end to end modeli dosta korisceniji u praksi, statisticki modeli ne zaostaju previse Cilj je bio opisati generalne ideje i principe na kojima se zasnivaju navedeni modeli, dok se detalji implementacije istih mogu naci u originalnim radovima.

# Literatura


[1] Microsoft, "Historic achievement: Microsoft researchers reach human parity in conversational speech recognition," 2016.

[2] A. Hannun, "The history of speech recognition to the year 2030," 2021.

[3] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[4] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *Interspeech 2020*, Oct 2020.

[5] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014.

[6] M. A. Anusuya and S. K. Katti, "Speech recognition by machine, a review," 2010.

[7] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, p. 7–19, jan 2015.

[8] U. Kamath, J. Liu, and J. Whitaker, *Deep Learning for NLP and Speech Recognition*. Springer Publishing Company, Incorporated, 1st ed., 2019.

[9] N. Dave, "Feature extraction methods lpc, plp and mfcc in speech recognition," *International journal for advance research in engineering and technology*, vol. 1, no. 6, pp. 1–4, 2013.

[10] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, (New York, NY, USA), p. 369–376, Association for Computing Machinery, 2006.

[12] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," 2015.

[13] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," 2015.

# A Dodatak: Pregled skupova podataka

| Naziv skupa | Duzina (sati) | Broj recenica | Broj govornika | Nivo transkripcije |
|---|---|---|---|---|
| TIMIT | 5,4 | ? | ? | foneme, reci |
| Switchboard-1 | 260 | ? | ? | reci |
| LibriSpeech | 100 | ? | ? | ? |
| CommonVoice | 2015 | ? | ? | ? |
| GigaSpeech | 10.000 | ? | nepoznato | reci |
| VoxPopuli | 543 | 1313 | 413.581 | reci |