

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

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada. Obratite pažnju da je pored ove .pdf datoteke, u prilogu i odgovarajuća .tex datoteka, kao i .bib datoteka korišćena za generisanje literature. Na prvoj strani seminarskog rada su naslov, apstrakt i sadržaj, i to sve mora da stane na prvu stranu! Kako bi Vaš seminarski zadovoljio standarde i očekivanja, koristite uputstva i materijale sa predavanja na temu pisanja seminarskih radova. Ovo je samo šablon koji se odnosi na fizički izgled seminarskog rada (šablon koji *morate* da koristite!) kao i par tehničkih pomoćnih uputstava. Pročitajte tekst pažljivo jer on sadrži i važne informacije vezane za zahteve obima i karakteristika seminarskog rada.

Ključne reči: prepoznavanje govora

Sadržaj

1	Uvod	2
2	Izazovi	2
3	Statisticki model	3
3.1	Procesiranje zvucnog signala	4
3.2	Akusticki model	5
3.3	Model izgovora	6
3.4	Jezicki model	6
3.5	Dekodiranje	7
4	End-to-end model	7
4.1	CTC model	7
4.2	Modeli zasnovani na paznji	8
5	Metrike za evaluaciju	10
6	Zaključak	10
	Literatura	10
A	Dodatak: Pregled skupova podataka	11

1 Uvod

Govor je za ljude najintuitivniji i prirodni način komunikacije. Zbog toga je od samog nastanka kompjutera, nastala i ideja da koristimo isti način komunikacije da interagujemo sa njima. To bi znatno smanjilo potrebno predznanje za korišćenje kompjutera i učinilo ga pristupačnijim većem broju ljudi. Najveća prepreka ovim sistemima do skoro je bio kako sa velikom tačnošću prepoznati šta 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 Assistant¹, Apple Siri², ...), transkripcija snimaka, pretraživanje audio sadržaja i pristupačnost.

Iako su istraživanja na ovu temu počela još sredinom dvadesetog veka, popularnost je počela da dobija tek u poslednjoj deceniji kada je uvođenje dubokih neuronskih mreža drastično povećalo performanse ovih sistema. Ta razlika je bila dovoljna da učini ove sisteme praktično primenljivim umesto nezgodnim za upotrebu zbog velikog broja gresaka. Jedan od najznacajnijih postignuća 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 poboljšanja se smatraju [2]:

1. Sakupljanje velike količine transkribovanih skupova podataka
2. Nagli porast u performansama grafičkih procesorskih jedinica (GPU)
3. Poboljšanje algoritama za učenje i arhitektura modela

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

2 Izazovi

Prepoznavanje govora je veoma tezak 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 zvucnih snimaka koji treba da sadrže više govornika različitog 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 što 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 poredjani po težini prepoznavanja su:

1. Izolovane reci — reci su razdvojene dugim periodima tisine
2. Povezane reci — reci su razdvojene kratkim pauzama
3. Neprekidan govor — uvezbani govor, citanje ili diktiranje
4. Spontan govor — neuvezbani, prirodni govor

Prve implementacije prepoznavanja govora su radile na nivou izolovanih reci i koristile su se za prepoznavanje određenih komandi ili cifara. Danas se najviše truda ulaze u poboljšanje neprekidnog i spontanog govora.

- **Karakteristike govornika** — Svaki čovek ima različitu boju glasa i govori različitom brzinom. Čak i starost osobe i jčina govora bitno uticu na frekvenciju glasa. Poseban problem pravi postojanje različitih dijalekata i akcenata koji mogu da imaju potpuno različite načine za izgovaranje istih reci. 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 jedno 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 (koliko je to moguće). Neke vrste sumova 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 primarnog govornika, i samim tim teško ukloniti.
- **Velicina rečnika** — Povećanje broja reci 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 velicine 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 opsteg govora današnji sistemi su trenirani na skupu od oko 50.000-100.000 reci.

3 Statisticki model

Dugo vremena statisticki pristup je bio dominantan za sisteme za prepoznavanje govora. Cilj ovih sistema je da pronađu najverovatniju transkripciju za zadati ulaz. Formalno, neka je \hat{W} optimalan niz reci za transkripciju nekog zvučnog signala X . Cilj je optimizovati formulu [8]:

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

primenom Bajesove formule to možemo 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:

$$\hat{W} = \operatorname{argmax}_W P(X|W)P(W) \quad (1)$$

Prvi deo, $P(X|W)$, racuna **akusticki model** (eng. *acoustic model*), a drugi, $P(W)$, racuna **jezicki model** (eng. *language model*). Ideja je da odvojeno optimizujemo obe prethodne velicine i ocekujemo da ce se tako maksimizirati ukupna verovatnoca.

Najprirodniji nacin za racunanje $P(X|W)$ bi bio na nivou reci zato sto krajnji izlaz svakako treba da bude niz reci. Problem sa tim je sto postoji veliki broj varijacija u izgovoru reci, a nedovoljan broj primera za svaki od njih u trening skupu. Umesto toga, reci se dele na manje delove kao sto su foneme. **Foneme** su najmanje jezikice jedinice cijom kombinacijom se dobijaju reci. One postoje samo kao apstraktna ideja a njihova fizicka realizacija se zove glas.

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

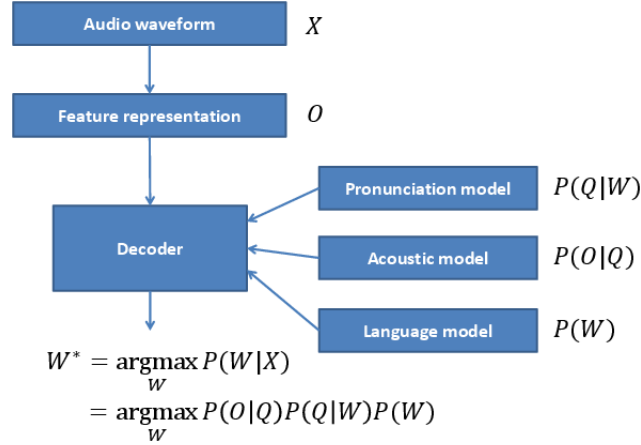
$$\hat{W} = \operatorname{argmax}_W \sum_S P(X, S|W)P(W)$$

sto se moze aproksimirati kao:

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

gde se $P(S|W)$ zove **model izgovora** (eng. *pronunciation model*).

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



Slika 1: Statisticki model

U nastavku ce biti opisana svaka od pomenutih komponenti, njena uloga i nacin rada.

3.1 Procesiranje zvucnog signala

Sirovi zvucni signal je veoma nepogodan za koriscenje zato sto sadrzi veliku kolicinu nepotrebnih informacija i šuma. Zbog toga se pre prosledjivanja akustickom modelu, signal prvo obradjuje tako da ostanu samo kljucne karakteristike i smanji sum i velicina reprezentacije.

Signal delimo na kratke segmente koje zovemo **okviri** (eng. *frame*). Svaki od njih je fiksne duzine (obicno 10-30 milisekundi) sa kratkim preklapanjem susednim okvirima radi smanjenja naglih promena prilikom prelaska iz jednog u drugi. Pretpostavka je da je u svakom od 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 najcesce zasnovana na Furijeovoj transformaciji kojom se izdvajaju samo njegove najbitnije karakteristike. Izdvajanje karakteristika pokusava da oponasa ljudski slusni sistem filtrirajuci odredjene frekvencije i skalirajuci ih onako kako bi ih covek cuo. Tacna reprezentacija koja se koristi varira u zavisnosti od modela, ali jedna od najpopularnijih je MFCC [?].

3.2 Akusticki model

Akusticki model je zaduzen da pretvori obradjeni zvucni signal u sekvencu fonema. Za resavanje ovog problema uvescemo teoriju o skrivenim Markovljevim modelima [9].

Neka je $S = \{s_1, s_2, \dots, s_n\}$ skup stanja, A matrica dimenzije $n \times n$ gde A_{ij} predstavlja verovatnocu prelaska iz stanja s_i u s_j tako da vazi $A_{ij} > 0$ i $\sum_{j=1}^n A_{ij} = 1$. U diskretnim trenucima vrsi se promena trenutnog stanja na osnovu verovatnoca zadatih matricom A . Specijalno, pocetno stanje moze da se izabere nasumicno iz neke raspodele. Za svaki trenutak $t = 1, 2, \dots, T$, oznacimo stanje u tom trenutku sa q_t . Ukoliko dodatno vazi da verovatnoca prelaska u sledece stanje vazi samo od trenutnog stanja:

$$P(q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_k, \dots) = P(q_t = S_i | q_{t-1} = S_j)$$

i da se verovatnoce prelaska iz jednog stanja u drugo ne menjaju:

$$P(q_t = S_i | q_{t-1} = S_j) = P(q_2 = S_i | q_1 = S_j) \forall t = 2..T$$

onda se prethodno opisan sistem naziva **Markovljev model**.

Prosirimo sada ovaj sistem dalje. Neka vase sve prethodno uvedene oznake i dodatno neka je $V = \{v_1, v_2, \dots, v_m\}$ skup mogucih obzervacija za svako stanje i B matrica dimenzije $n \times m$ gde B_{jk} predstavlja verovatnocu obzervacije v_k u stanju s_j . Sada u svakom trenutku t osim promene stanja dobijamo i novu obzervaciju x_t na osnovu verovatnoca zadatih matricom B . Razlika sada je sto novo stanje ostaje skriveno nego se samo vidi obzervacija u svakom trenutku. Ako jos vazi da je svaka obzervacija zavisi samo od trenutnog stanja a ne od prethodnih obzervacija:

$$P(x_t = v_i | x_{t-1}, \dots, x_1, q_t, \dots, q_1) = P(x_t = v_i | q_t)$$

tada se ovaj sistem se naziva **skriveni Markovljev model**.

Konkretno za prepoznavanje govora koristicemo ove modele da opisemo svaku fonemu. Svaki od njih ce se sastojati od nekog zadatog broja stanja pri cemu ce obzervacije govoriti da li je ta fonema trenutno primecena. Kada imamo model za svaku fonemu, njih je moguće kombinovati da se dobije model za rec, koji se dalje kombinuju u model za recenicu. Sada jos samo treba odrediti raspodele verovatnoca za obzervacije fonema. To se radi treniranjem modela nad skupom podataka uz pomoc Baum-Welch algoritma zasnovanog na dinamicnom programiranju.

Istrenirani model posle moze da predvidi najverovatniji niz fonema koristeći Vetrebi algoritam za pretragu.

3.3 Model izgovora

U prethodnoj sekciji je receno da se modeli fonema spajaju u modele reci, ali nije napomenuto tacno kako se to radi. To je bas zadatak modela izgovora. On je u sustini veliki rečnik koji mapira reci u niz fonema kako se one izgovaraju. Ako postoji vise varijacija izgovora one se smatraju kao razlicite stavke u rečniku. Sada ako je poznato ovo preslikavanje lako se moze za datu recenicu konstruisati skriveni Markovljev model spajanjem modela odgovarajucih reci za koje je sada poznato preslikavanje u foneme.

Prirodno sledece pitanje je kako se odredjuju ova preslikavanja. Ovo je zapravo jedan od najtezih zadataka za modeliranje zato se on ne uci na skupu podataka nego ga konstruisu eksperti iz tog domena. Za svaku rec, neko je morao da zapise na koji nacin se izgovara uzimajuci u obzir da potencijalno postoji vise izgovora. Sa obzirom da danasnji sistemi razlikuju oko 100.000 reci, ovo nimalo nije lak posao.

Na engleskom jeziku, jedan od najpoznatijih rečnika izgovora je CMUdict³ od Carnegie Mellon univerziteta koji sadrzi preko 134.000 reci.

3.4 Jezicki model

Povratkom na formulu 1, jezicki model dodeljuje verovatnocu pojavljivanja $P(W)$ svakoj mogucoj sekvenci reci W . Ovde se uzima u obzir relativna ucestalost reci, verovatnoca da se reci nadju jedna za drugom, i mogu da se vrse dodatne sintaksne i semanticke provere. Postoje recenice koje zvuče slicno ali nemaju sve semanticko znacenje. Tada se od njih bira ona koja ima najvise 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 $W = \{w_1, w_2, \dots, w_m\}$ 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:

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

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:

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

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 moguće je da se neka sekvenca nije ranije videla. Taj problem se resava smanjenjem n ukoliko ne postoji ta sekvenca duzine n pa koriscenjem te verovatnoce ili nekim postupkov uglađivanja.

³<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

3.5 Dekodiranje

Konstrukcijom svih prethodno opisanih komponenti i njihovim treniranjem, model je gotov i spreman za upotrebu. Jedino sto ostaje je pretražiti prostor dopustivih recenica da se pronadje ona koja najviše odgovara glasovnom signalu. U praksi, taj prostor je veoma veliki i njegova pretraga je eksponencijalne složenosti (svaka moguća kombinacija reci u recenici) stoga nije izvodljivo tražiti egzaktno rešenje. Umesto toga se koristi heuristički beam search algoritam. Ideja je da se rešenje gradi iterativno i u svakom trenutku umesto testiranja svih mogućih puteva biramo b najverovatnijih puteva. Parametar b se bira tako da balansira veličinu prostora za pretraživanje i vreme potrebno za njegov obilazak. Algoritam je dakle sledeći: odredimo verovatnocu za svaku rec da bude prva pa od njih izaberemo b najverovatnijih. U sledećem koraku svaki od tih b reci produžujemo sledećom i od njih ponovo biramo k najverovatnijih. Ovaj postupak se ponavlja sve dok ne dodjemo do kraja recenice i ona predstavlja konačnu transkripciju govora.

4 End-to-end model

4.1 CTC model

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

$$h : (\mathbb{R}^m)^* \rightarrow L^* \quad (3)$$

U praksi fiksiramo broj vremenskih trenutaka na neku vrednost T . Iako je broj trenutaka fiksiran za zvučne signale, nizovi labela ne moraju biti iste dužine za svaku ulaznu instancu, stoga jednu trening instancu predstavlja par (\mathbf{x}, \mathbf{z}) gde je \mathbf{z} vektor labela dužine najviše T . Ako bismo test skup označili sa O , tada funkciju greske možemo definisati na sledeći način:

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

ED predstavlja edit distancu (minimalni broj izmena koji dovodi jedan niz karaktera do drugog, pri čemu 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.

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

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

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

U praksi uzorkovanje zvuka se vrši u veoma sitnim vremenskim intervalima (oko 10ms). Stoga je pojava blanko ili dupliciranih oznaka veoma česta. Iz tog razloga uvodimo preslikavanje β čija je uloga preciscavanje nizova labela uklanjanjem blanko oznaka i susednih duplikata.

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

Primetimo da za jednu preciscenu putanju l može postojati više mogućih izvornih putanja, pa je verovatnoća njenog odabira jednaka sumi po svim izvornim putanjama.

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

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

$$h(x) = \operatorname{argmax}_{l \in L^U} p(l|x) \quad (6)$$

Najjednostavniji algoritam jeste pohlepni odabir najbolje oznake za svaki vremenski trenutak ponaosob. Međutim, ovakav pristup ne garantuje optimalnost. Naravno, postoje bolji algoritmi za resavanje datog problema. Isti neće biti obrađivani u ovom radu, ali se mogu naći u [1].

4.2 Modeli zasnovani na paznji

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

Glavne komponente ovog modela 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 konačno daje niz karaktera.

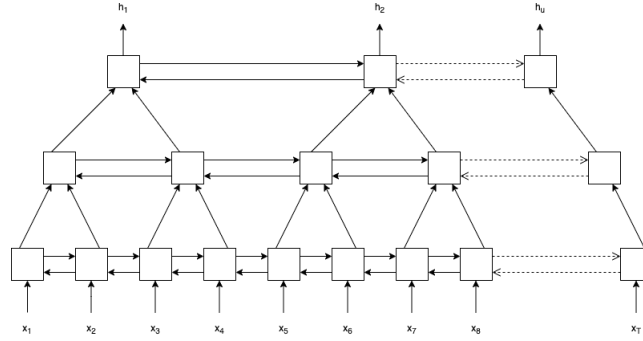
Ubaci sliku

Slično kao u poglavlju [1] sa $\mathbf{x} = (x_1, x_2, \dots, x_T)$ označavamo ulazni signal. \mathbf{y} predstavlja niz mogućih karaktera uključujući razmak, tacku, zarez, apostrof, i tri specijalna karaktera (start - početak rečenice, stop - kraj rečenice, nep - nepoznat zvuk??).

Listener jeste rekurentna neuronska mreža piramidalne strukture. Kao što se može primetiti na slici iznad broj neurona se u pravcu izlaznog sloja polovi. Ova osobina nam omogućava smanjenje složenosti izracunavanja, što je veoma poželjno u slučaju kada je interval uzorkovanja veoma uzak.

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

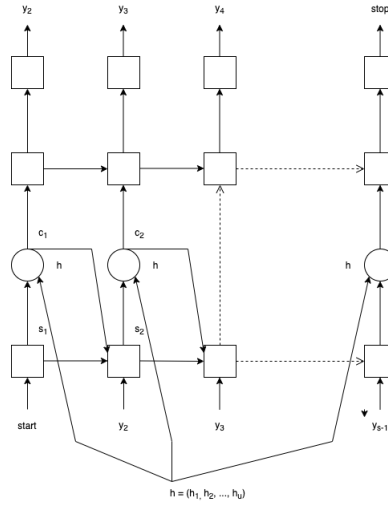
$$P(y|x) = \prod_i P(y_i|x, y_j, j < i) \quad (7)$$



Slika 2: Listener model

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.

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



Slika 3: Speller model

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.

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

gde je \bar{y}_j objektivna vrednost karaktera j .

Finalno dekodiranje vrši se algoritmom beam pretrage (*eng. beam search*). U koraku proširivanja trenutnog cvora pretrage bira se β najverovatnijih karaktera po verovatnoci dobijenoj na izlazu Speller komponente, gde je β parametar beam search algoritma.

5 Metrike za evaluaciju

Standardna mera za procenu kvaliteta sistema za prepoznavanje govora je stopa pogresnih reci (*eng. Word Error Rate (WER)*).

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

gde je S broj zamenjenih reci, D broj obrisanih reci, I broj umetnutih reci, i N ukupan broj reci u referentnoj recenici. Minimalna vrednost koju može da dobije je 0, dok maksimalna vrednost može da bude preko 1 (npr. veliki broj umetnutih reci). Stopa pogresnih reci se efikasno računa dinamičkim programiranjem, pomoću Vagner-Fiserovog algoritma (*eng.*)

Cilj sistema za prepoznavanje govora je da minimizuje ovu vrednost.

6 Zaključak

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] 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.

A Dodatak: Pregled skupova podataka

TIMIT	cell2	cell3
SwitchBoard	cell5	cell6
LibriSpeech	cell8	cell9
CommonVoice	a	a