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

17. 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

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 tezak zadatak zato sto je potrebno da radi podjednako dobro u veoma razlicitim uslovima. Neki od najvecih izazova su:

- **Mala kolicina podataka za trening** — Za ostvarivanje dobrih rezultata potrebno je sakupiti vise stotina ili cak hiljada sati labeliranih zvucnih snimaka koji treba da sadrze vise govornika razlicitog pola i starosti, koji govore razlicitim akcentima. Dok u skorije vreme jeste nastao porast u kolicini dostupnih podataka, veliki problem jos uvek predstavlja reprezentativnost razlicitih varijacija u govoru i nedostatak podataka za jezike sa manjim brojem govornika. Zbog toga se istrazuju alternativni nacini za treniranje kao sto su samotreniranje (eng. *self-training*) [3], iterativno treniranje [4] ili treniranje koristeci kompjuterski generisan glas [5]. U dodatku A se moze naci 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 razliciti sistemi u zavisnosti od toga koji tip govora mogu da prepoznaju [6]. Tipovi govora poredjani po tezini propoznavanja 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 odredjenih komandi ili cifara. Danas se najvise truda ulaze u poboljsanje neprekidnog i spontanog govora.
- **Karakteristike govornika** — Svaki covek ima razlicitu boju glasa i govori razlicitom brzinom. Cak i starost osobe i jacina govora bitno uticu na frekvenciju glasa. Poseban problem pravi postojanje razlicitih dijalekata i akcenata koji mogu da imaju potpuno razlicite nacine za izgovaranje istih reci. Jedan nacin za resavanje ovog problema je treniranje sistema na glasu govornika koji ce ga koristiti. To su sistemi zavisni od korisnika (eng. *speaker dependent*) i koriste se u slucajevima 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.
- **Okruzenje govornika** — Ovi sistemi ce retko biti korisceni u potpuno tihim prostorijama sa profesionalnom opremom za snimanje, zbog toga treba da budu tolerantni na razlicite vrste pozadinske buke ili kvaliteta mikrofona (koliko je to moguce). Neke vrste sumova je moguce otkloniti analizom zvuka ili naprednijim metodama [7], ali jedan od najvecih problema predstavlja postojanje drugih govornika u okolini. Te signale je cesto tesko razlikovati od glasa primarnig govornika, i samim tim tesko ukloniti.
- **Velicina recnika** — Povecanje broja reci koje model moze da prepozna takodje povecava njegovu slozenost i otezava treniranje, ali se time dobija na tacnosti. Zbog toga je potrebno naci dobar kompromis izmedju velicine recnika i slozenosti modela. U slucajevima kada je potrebno pouzdano prepoznati samo neki skup komandi koriste se mali recnici i oni su cesto veoma pouzdani, ali za prepoznavanje opsteg govora danasnji 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 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:

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

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 jezicke 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:

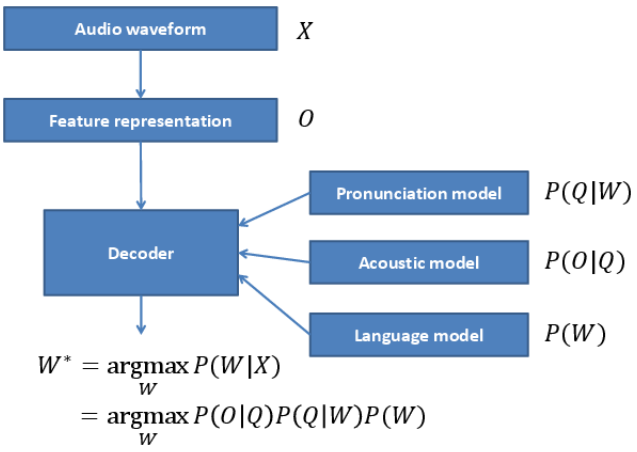
$$\hat{W} = \underset{W}{\operatorname{argmax}} \sum_{S} P(X, S|W)P(W)$$

sto se moze aproksimirati kao:

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

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 $S = \{s_1, s_2, \ldots, s_n\}$ 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 $t = 1, 2, \ldots, T$, oznacicemo stanje u tom trenutku sa q_t. Ukoliko dodatno vazi da verovatnoca prelaska u sledece stanje vazi samo od trenutnog stanja:

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

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 vaze sve prethodno uvedene oznake i dodatno neka je $V = \{v_1, v_2, \ldots, v_m\}$ 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:

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

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 moguce 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 dinamickom programiranju.

Istrenirani model posle moze da predvidi najverovatniji niz fonema koristeci 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 recnik koji mapira reci u niz fonema kako se one izgovaraju. Ako postoji vise varijacija izgovora one se smatraju kao razlicite stavke u recniku. 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 presikavanje 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 recnika izgovora je CMUdict [3] 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 zvuce 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 $W = \{w_1, w_2, \ldots, w_m\}$ 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] 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 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 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 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 ϵ.

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 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 vrsi u veoma sitnim vremenskim intervalima (oko 10ms). Stoga je pojava blanko ili dupliciranih oznaka veoma cesta. Iz tog razloga uvodimo preslikavanje β cija 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 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

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 moze da dobije je 0, dok maksimalna vrednost moze da bude preko 1 (npr. veliki broj umetnutih reci). Stopa pogresnih reci se efikasno racuna dinamickim programiranjem, pomocu 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