Razpoznavanje slovenskega govora z metodami globokih nevronskih mrež

Matej Ulčar[1], Simon Dobrišek[2], Marko Robnik-Šikonja[1]
[1]Univerza v Ljubljani, Fakulteta za računalništvo in informatiko,
Večna pot 113, 1000 Ljubljana
matej.ulcar@fri.uni-lj.si marko.robnik@fri.uni-lj.si
[2]Univerza v Ljubljani, Fakulteta za elektrotehniko,
Tržaška 25, 1000 Ljubljana
simon.dobrisek@fe.uni-lj.si

Izvleček

V zadnjem času se na področju samodejnega razpoznavanja govora uveljavljajo globoke nevronske mreže, ki nadomeščajo akustično modeliranje z uporabo modelov HMM in GMM ter n-grame za jezikovni model. Za razpoznavanje govorjene slovenščine smo izdelali in preizkusili več arhitektur časovno zakasnjenih nevronskih mrež in nevronskih mrež z dolgim kratkoročnim spominom na akustičnem in jezikovnem modelu v sistemu Kaldi. Razpoznavalnik smo učili na obširnem besednjaku, ki vsebuje približno milijon različnih besed. Najboljše rezultate smo dosegli s časovno zakasnjenimi nevronskimi mrežami, kjer smo dosegli 27,16 % napako po kriteriju WER. Preliminarni rezultati kažejo boljšo natančnost v primerjavi z Googlovim modelom »speech-to-text«, vendar pa je za zanesljivo primerjavo potrebno več dodatnega testiranja.

Ključne besede: strojno učenje, globoke nevronske mreže, razponavanje govora, govorne tehnologije, obdelava naravnega jezika

Abstract

Recently, deep neural networks have become the predominant approach to automatic speech recognition, replacing classical acoustical modelling using GMM and HMM models and n-grams for the language model. For the recognition of spoken Slovene, we have developed and tested several architectures of time-delayed neural networks and neural networks with a long short-term memory for both acoustic and language models in the Kaldi environment. We used a large lexicon containing about a million words. Time-delayed neural networks achieved the best results on continuous speech, with a 27.16% error according to the WER criterion. Preliminary results show better performance compared to Google's speech-to-text model. However, more testing is needed for a statistically valid comparison.

Keywords: Machine learning, deep neural networks, speech recognition, speech technologies, natural language processing

1 UVOD

Govor večkrat želimo zapisati kot besedilo, na primer zapisnik sestanka, zapiske s predavanj, podnapise na televiziji v pomoč slušno prizadetim, ipd. Za zapis je potrebno govor večkrat poslušati in ga sproti zapisovati, kar je lahko časovno potratno, še posebej, ko je govor hiter in je potrebno posnetek ustavljati in ponovno predvajati.

Problem razpoznavanja govora je sestavljen iz dveh delov: akustičnega modeliranja in jezikovnega modeliranja. Akustično modeliranje se nanaša na gradnjo modelov posameznih glasov oziroma fonemov za dani govorjeni jezik. Fonem je osnovna enota glasu, ki razločuje pomen besed. Pri akustičnem modeliranju namesto fonemov pogosto uporabljamo trifone, to je, po tri foneme združene skupaj v eno enoto. Jezikovno modeliranje pa se nanaša na modeliranje preslikave fonemov v besede in nizanja besed v besedila. Za reševanje problemov obdelave naravnih jezikov se v zadnjem času zelo uspešno uporabljajo globoke nevronske mreže (DNN – Deep Neural Networks). Z uporabo globokih nevronskih mrež smo poizkusili izboljšati rezultate do sedaj najuspešnejših metod strojnega učenja pri razpoznavanju govora v slovenščini in izdelati kakovostno odprtokodno rešitev.

Članek sestavlja šest razdelkov. V drugem na kratko opišemo sorodna dela. V tretjem razdelku

opišemo uporabljene vire. V četrtem razdelku opišemo uporabljene tehnologije in predstavimo arhitekturo razpoznavalnika, oziroma različne postopke učenja, ki smo jih uporabili. V petem razdelku sledi predstavitev in analiza rezultatov. V sklepnem delu opišemo opravljeno delo ter predstavimo možnosti za izboljšave.

2 PREGLED SORODNIH DEL

Razpoznavanje govora je najbolj razvito za angleški jezik, kjer so trenutno najuspešnejši modeli, naučeni z uporabo globokih nevronskih mrež. Microsoft v svojem sistemu za razpoznavanje pogovornega govora (Xiong in sod., 2017) uporablja globoke nevronske mreže za akustični in jezikovni model. Akustični model je naučen s kombinacijo konvolucijskih nevronskih mrež in nevronskih mrež z dolgim kratkoročnim spominom (angl. long short-term memory, LSTM). Za učenje jezikovnega modela so uporabili rekurenčne nevronske mreže (angl. recurrent neural network, RNN).

Googlova aplikacija za pametne telefone, ki uporablja glasovno upravljanje ter omogoča glasovno iskanje, uporablja za razpoznavanje nevronske mreže pri akustičnem modelu (Sak, Senior, Rao, Beaufays, Schalkwyk, 2015). Sak, Senior, Rao in Beaufays (2015) so uporabili dvosmerne globoke nevronske mreže LSTM. Za poravnavo zvočnega posnetka s transkripcijo v učni množici so namesto modela GMM-HMM (GMM – Gaussian Mixture Model - mešanica Gaussovih porazdelitev, HMM – Hidden Markov Model - prikriti Markovov model) uporabili metodo povezovalne časovne klasifikacije (angl. Connectionist Temporal Classification – CTC).

Hernandez in sod. (2018) so uporabili zbirko orodij Kaldi za učenje sistema za razpoznavanje govora v angleščini. Za govorno zbirko so uporabili zbirko predavanj TED. Akustični model so naučili s hibridnim modelom (DNN-HMM), kjer so najprej uporabili pristop GMM-HMM za učenje in poravnavo zvočnih posnetkov s transkripcijo. Nato so namesto GMM uporabili časovno zakasnjene nevronske mreže (angl. time delayed neural network, TDNN). Naučili so dva jezikovna modela. Prvega z n-grami reda 4 in drugega z uporabo nevronskih mrež, kjer so uporabili tri nivoje TDNN, med njimi pa dva nivoja LSTM.

Bolka (2016) je v diplomskem delu uporabil zbirko orodij Kaldi za razpoznavanje fonemov v slovenščini. Uporabil je več metod učenja akustičnega modela; model zgrajen z nevronskimi mrežami je dosegel najboljše rezultate. Nevronske mreže LSTM za prevajanje med fonemi in grafemi (tekstovnimi zapisi fonemov) predlagajo tudi Rao, Peng, Sak in Beaufays (2015). V svojem delu izdelajo model, ki napoveduje foneme glede na dane grafeme, z drugimi besedami napovedujejo izgovor besede. Uporabili so tako plitke kot globoke nevronske mreže LSTM. V našem delu obravnavamo obraten problem, kjer določamo zapis besede glede na njen izgovor. Globoke nevronske mreže LSTM vsebujejo posebne enote, imenovane spominske celice. Te so si zmožne podatke zapomniti poljubno dolgo. Pozabna vrata spominske celice skrbijo, da se podatek lahko po potrebi tudi pozabi. Zaradi teh lastnosti so globoke nevronske mreže LSTM zelo dobre pri prepoznavanju govora, saj pri učenju upoštevajo tudi kontekst (zapis besede je odvisen tudi od predhodnih glasov, ne samo od trenutnega) (Bolka, 2016; Rao in sod., 2015).

Jezikovni modeli pri razpoznavanju govora v slovenščini večinoma uporabljajo Good-Turingovo glajenje (Žgank, Donaj, Sepesy Maučec, 2014; Žgank, Verdonik, Sepesy Maučec, 2016; Donaj, 2015). Žgank in sod. (2014) so uporabili dve govorni bazi, eno z večjim deležem spontanega govora, drugo z večjim deležem branega govora. Akustični model so osnovali na zveznih prikritih Markovovih modelih. Žgank in sod. (2014) so ugotavljali predvsem vpliv velikosti uporabljenih besedilnih in govornih korpusov. Njihova analiza je pokazala, da večanje uporabljenih virov izboljša rezultate, vendar je to izboljšanje majhno, za večjo izboljšavo je potrebna tudi uporaba drugih algoritmov.

Žgank in sod. (2016) so uporabili enak razpoznavalnik kot prej opisani (Žgank in sod., 2014) za transkribiranje nove govorne baze SI TEDx-UM. Uporabili so dva jezikovna modela, enega grajenega na govorni bazi BNSI, drugega pa zgolj na besedilnem korpusu FidaPLUS. Njihovi rezultati so pokazali, da je modeliranje govorjene rabe jezika pomemben del jezikovnega modela, po drugi strani pa ima tematika govora velik vpliv na natančnost razpoznavanja besed. Oba jezikovna modela sta dosegla enak rezultat (napako 50,7 %). Ker se domeni govora pri SI TEDx-UM in BNSI med seboj precej razlikujeta, so rezultati pri razpoznavanju govora na predavanjih SI TEDx-UM precej slabši od rezultatov pri razpoznavanju govora posnetkov BNSI.

3 OPIS VIROV

Učenje dobrega razpoznavalnika govora zahteva mnogo učnih podatkov. Potrebujemo posnetke govora, njihove prepise, korpus in slovar besed z izgovorjavami. Za govorne posnetke smo uporabili korpuse Gos 1.0 (Zwitter Vitez in sod., 2013), Gos VideoLectures 2.0 (od tu dalje uporabljamo oznako GosVL) (Verdonik in sod., 2017) in Sofes 1.0 (Dobrišek in sod., 2017). Lastnosti govornih korpusov so opisane v tabeli 1.

Tabela 1: **Lastnosti govornih korpusov**

Lastnost	Gos	GosVL	Sofes
Dolžina posnetkov	120 ur	9 ur 48 minut	9 ur 52 minut
Št. vseh govorcev	1526	44	134
Št. ženskih govork	681	17	32
Št. moških govorcev	845	27	102
Število stavkov	96334	4073	12536

Korpus Sofes ima nekatere stavke podvojene, vendar so ti posneti v različnih kvalitetah. Posnetke nizke kvalitete smo pri učenju izločili. Korpusa Gos in Sofes smo razdelili na učno in validacijsko množico z namenom, da na validacijski množici izberemo dobro delujoče parametre in se s tem izognemo pretiranemu prilagajanju učni množici. Posnetki iz obeh korpusov so prisotni tako v učni kot v validacijski množici. Razdelili smo ju tako, da se nihče izmed govorcev ne pojavi v obeh množicah, vedno le v eni. Korpus GosVL smo uporabili za končno testno množico.

Korpusa Gos in Sofes imata govor prepisan na dva načina. V prvem so stavki in besede zapisani v zbornem knjižnem jeziku, v drugem pa je zapis fonetičen. Ta dva zapisa smo želeli uporabiti za izdelavo slovarja izgovarjav. Korpus Sofes je sam premajhen, vsebuje premalo različnih besed. Pri obdelavi podatkov v korpusu Gos pa smo našli neujemanja med obema transkripcijama. Zapisa nimata vedno enakega števila besed, morda tudi ne vedno enakega vrstnega reda. Rezultati so bili neuporabni, zato smo se odločili, da uporabimo leksikon besednih oblik Sloleks (Dobrovoljc in sod., 2015). Prednost je, da vsebuje več besed, oziroma besednih oblik kot korpusa Gos in Sofes. Za naš namen pripravljen leksikon Sloleks vsebuje 1.129.141 različnih besednih oblik, korpus Gos pa le 83.000. Slabost uporabe Sloleksa je, da smo morali sami določiti fonetični zapis.

Besedilni korpus smo oblikovali na podlagi korpusa ccGigafida (Logar in sod., 2013). Korpus vsebuje približno 103 milijone besed. Iz korpusa ccGigafida smo izločili vse prazne vrstice, odstranili večkratne presledke, oziroma jih nadomestili z enojnimi ter odstranili vsa ločila. Tako je učenje jezikovnega modela lažje. V nasprotnem primeru model upošteva ločila kot del besede, teh besed pa ni v slovarju. S tem bi se povečalo število besed, ki bi jih morali upoštevati, korpus bi moral biti mnogo večji, učenje pa bi bilo zahtevnejše in počasnejše.

3.1 Obdelava podatkov

Govorni korpusi imajo transkripcijo zapisano v datotekah *.trs, ki so oblika formata XML. Za učenje z uporabo orodij Kaldi moramo najprej iz teh datotek izluščiti potrebne podatke. V ta namen smo napisali skripta, ki preberejo datoteke *.trs in vsakemu izreku pripišejo unikaten identifikator (ID), ID govorca, spol govorca ter datoteko zvočnega posnetka. Ti podatki se shranijo v različne datoteke. Datoteka *text* vsebuje v vsaki vrstici najprej identifikator izreka, nato sam izrek. Datoteka *utt2spk* vsebuje v vsaki vrstici identifikator izreka in identifikator govorca. Datoteka *spk2gender* vsebuje identifikator govorca in spol govorca (m za moški spol, f za ženski). Datoteka *wav.scp* vsebuje identifikator izreka in polno pot do zvočne datoteke. V določenih primerih je lahko v isti zvočni datoteki več izrekov. Vsi uporabljeni govorni korpusi imajo sicer zvočne posnetke razdeljene glede na posamezne izreke, vendar se pri korpusu GosVL ti ne ujemajo popolnoma. Število izrekov ni enako številu zvočnih posnetkov. Ročno ugotavljanje, kje pride do neujemanj, je zelo zamudno. Korpus GosVL ima zvočne posnetke razdeljene tudi na posamezna predavanja. Datoteke .trs vsebujejo podatke o časovni poziciji vsakega izreka znotraj zvočnega posnetka celotnega predavanja. Zato smo uporabili zvočne posnetke celotnih predavanj ter v datoteko *segments* za vsak izrek zapisali začetno in končno mesto (v sekundah) v zvočnem posnetku.

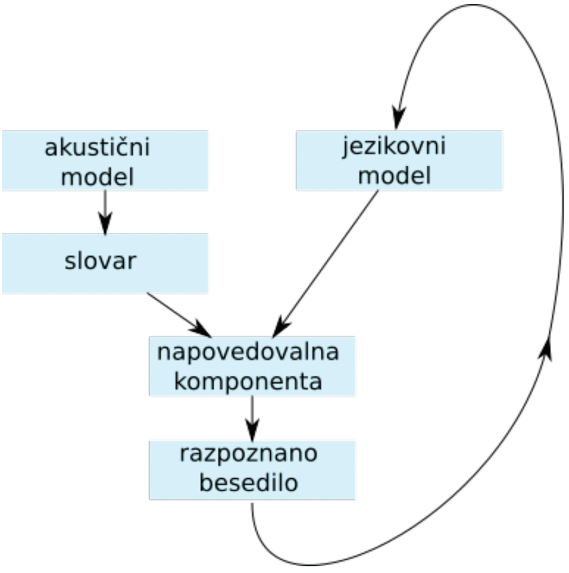
Korpus Gos ima pri nekaterih izrekih spol govorca označen kot »nedoločen«. V teh primerih smo najprej preverili ID govorca. Zadnja črka ID govorca namreč označuje spol govorca (»m« za moškega, »f« za žensko). V večini primerov, ko je spol označen kot nedoločen, je zadnja črka ID govorca »n«. Takrat smo se odločili, da govorcu pripišemo ženski spol »f«, ker so to večinoma posnetki otrok, ki so po višini glasu

bolj podobni ženskam, kot moškim. Nekateri izreki imajo več govorcev. Na primer, en govorec začne stavek, drugi ga dokonča. Ker ne vemo točno, kdaj govori kateri izmed govorcev, lahko pa se zgodi celo, da govorita hkrati, smo cel izrek pripisali enemu govorcu, tistemu, ki je v transkripcijski datoteki zapisan prvi.

Sloleks vsebuje nekaj več kot 100.000 lem, oziroma skupaj 2.791.919 besednih oblik. Odstranili smo podvojene vnose, na primer samostalnik »miza« ima enako obliko v rodilniku ednine ter imenovalniku in tožilniku množine (»mize«). V Sloleksu bi to bili trije vnosi, potrebujemo pa le enega, saj ne uporabljamo podatkov o spolu, številu, sklonu in podobno. Končno imamo 1.129.141 različnih besednih oblik. Izgovarjave teh besed smo tvorili s pomočjo pravil zapisa in pravil izgovarjave v Slovenskem pravopisu. Dodali pa smo še nekaj svojih pravil, ki bolje opisujejo pogovorni jezik, ne zgolj zborni knjižni jezik. Primer takega pravila je izgovarjava končnice »-el« z glasom »-u« (ne »-ew«). Z znakom »w« smo tu označili vse oblike dvoustničnega u.

4 Arhitektura in učenje razpoznavalnika

Sistem za razpoznavanje govora lahko v grobem razdelimo na dva dela, akustični model in jezikovni model. Akustični model naučimo na značilkah pridobljenih iz zvočnih posnetkov govora in pripadajočih transkripcijah govora. Akustični model zvočnemu signalu pripiše pripadajoč fonem. Jezikovni model naučimo iz besedilnega korpusa. Ta model na podlagi predhodnih besed predlaga najbolj verjetno naslednjo besedo. Akustični model prek slovarja, s pomočjo katerega napovedanim fonemom pripišemo najverjetnejšo besedo, povežemo z jezikovnim modelom (slika 1). Napovedovalna komponenta vrača najbolj verjetno naslednjo besedo z uteženim povprečjem napovedi akustičnega in jezikovnega modela. Razpoznavalnik smo naučili z uporabo orodja Kaldi, ki je odprtokodna zbirka orodij za učenje razpoznavanja govora (Povey in sod., 2011).



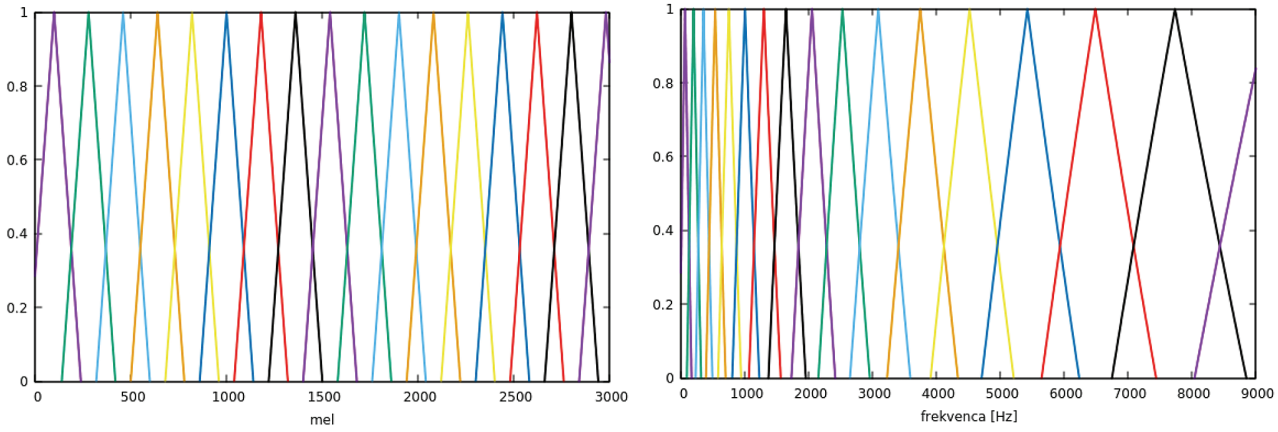
Slika 1: **Groba shema posameznih komponent, oziroma modelov in povezav med njimi.**

4.1 Akustični model

Učenje akustičnega modela razpoznavalnika govora poteka v več zaporednih povezanih fazah. V vsaki fazi smo uporabili drug model učenja, ki je kompleksnejši od prejšnjega. Zvočne posnetke smo najprej razrezali na kratke odseke, oziroma okna, dolga 25 ms, razdalja med sosednima oknoma pa je 10 ms. Signal v vsakem oknu smo transformirali s Fourierjevo transformacijo in nato izračunali značilke MFCC (mel-frekvenčni kepstralni koeficienti) (Davis, Mermelstein, 1980, Charan in sod., 2017). Značilke MFCC dobimo tako, da v mel frekvenčni skali definiramo filtre (slika 2), ki se deloma prekrivajo med seboj. Mel-frekvence (m) dobimo iz frekvenc v hertzih (f) z enačbo:

$$m(f) = 1125ln\left(1 + \frac{f}{700}\right) \qquad (1).$$

Za vsak filter zmnožimo spekter signala s filtrom in izračunamo logaritem spektralne energije znotraj filtra. Nad dobljenim izračunamo kosinusno transformacijo. Prvih 13 koeficientov obdržimo za naše značilke. Na teh značilkah smo učili naš sistem, v kasnejših fazah pa smo jim dodali še druge značilke, predvsem delta in delta-delta značilke, ki predstavljajo prvi, oziroma drugi časovni odvod značilk MFCC, torej njihovo spremembo v času.



Slika 2: **Primer mel-frekvenčnih filtrov, prikazanih v mel skali na levi in isti filtri v normalni**

Osnovo akustičnega modela predstavlja prikriti Markov model. Ta modelira spremembe zvočnega signala v času, to je prehode iz enega fonema v naslednjega. Fonemi predstavljajo skrita stanja v HMM, za njihov zapis pa moramo poznati izgovarjavo vsake besede. Za to uporabimo slovar, v katerem vsakemu geslu (besedi) pripišemo izgovor. Z drugimi besedami, zapišemo besedo in njen fonetični zapis. Opazovana stanja HMM so mešanica Gaussovih porazdelitev (GMM), ki opisujejo spekter posameznega časovnega izseka govornega signala.

Linearna diskriminantna analiza (LDA) je način, s katerim zmanjšamo število dimenzij v vektorjih značilk, obenem pa ohranimo diskriminantne značilnosti množice značilk. Rezultat so nižjedimenzionalni vektorji značilk, ki so manj korelirani in dobro razlikujejo med posameznimi razredi. Tako je učenje akustičnega modela lažje, oziroma hitrejše.

Linearna transformacija z največjim verjetjem (angl. maximum likelihood linear transform - MLLT), ki je poseben primer »delno povezane kovariance« (angl. semi-tied covariance - STC), se pogosto uporablja v kombinaciji z linearno diskriminantno analizo (Gales, 1999). Če imamo opazovana stanja v HMM predstavljena z GMM, bi morali za vsako komponento, torej za vsako Gaussovo porazdelitev, izračunati celotno kovariančno matriko Σ_{jm}. Namesto tega opišemo kovariančno matriko, kot da je sestavljena iz dveh delov. Prvi del predstavlja diagonalna matrika, ki je specifična za vsako komponento (Σ_{jm}^{diag}). Drugi del je delno povezana matrika $\mathbf{H}^{(r)}$, ki ni specifična za vsako komponento, ampak za vsak razred komponent. Delno povezana matrika $\mathbf{H}^{(r)}$ lahko predstavlja poljubno število komponent (Stuttle, 2003). Kovariančno matriko zapišemo kot:

$$\Sigma_{jm} = H^{(r)}\, \Sigma_{jm}^{(diag)}\, H^{(r)T} \qquad (2).$$

Za od govorca neodvisen sistem za razpoznavanje govora nujno potrebujemo veliko število govorcev v učni množici. Akustični modeli naučeni na taki učni množici zato vsebujejo tudi parametre, ki razlikujejo med posameznimi govorci. Naš cilj pa je razlikovati med različnimi besedami, ne glede na govorca. Problem rešujemo z učenjem s prilagajanjem govorcu (angl. speaker adaptive training - SAT). Začnemo z modelom neodvisnim od govorca, definiramo afino transformacijo

$$\mathrm{x'} = \mathbf{A}\mathrm{x} + \mathbf{b} \qquad (3),$$

kjer postavimo začetna **A=I** in **b=0**. S podatki le enega govorca (**o**(t)), kjer značilke transformiramo z dano transformacijo (3), učimo akustični model eno iteracijo. Shranimo povprečja (**P**), variance (μ) in posteriorne verjetnosti (γ) za vsako Gaussovo porazdelitev (m) in vsak časovni okvir (t). Ocenimo novi vrednosti **A** in **b**, z maksimizacijo spodnjega izraza (4).

$$K - \frac{1}{2} \Sigma\Sigma \gamma_m(t) \left(K^{(m)} + log(|P^{(m)}|) - log(|A|^2) + (Ao(t) + b - \mu^{(m)})^{T\, p(m)-1} (Ao(t) + \mathrm{b} - \mu^{(m)}) \right) \qquad (4)$$

Ponovimo učenje z novimi vrednostmi **A** in **b**, in ponavljamo opisani postopek dokler vrednosti ne konvergirajo (Gales, 1998). Konstanta K v izrazu (4) je

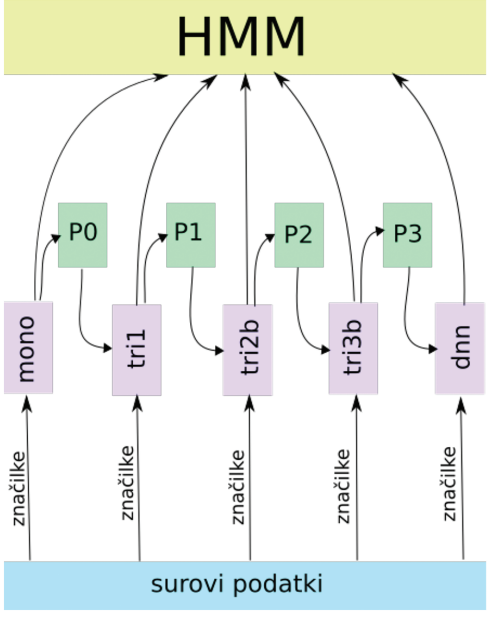
$$1 + \frac{f}{700}$$

odvisna le od verjetnosti prehodov med stanji HMM, $K^{(m)}$ pa je normalizacijska konstanta za vsak m. **A** in **b** sta različna za vsakega govorca, tako izraz (3) predstavlja transformacijo med kanoničnim modelom in modelom specifičnim za posameznega govorca.

Najprej smo uporabili različne modele GMM-HMM, povzete v tabeli 1. Uporabili smo monofonski model učenja (mono), trifonski model z delta in delta-delta značilkami (tri1), trifonski model z LDA in MLLT značilkami (tri2b), trifonski model z LDA, MLLT in dodanim prilagajanjem govorcu (SAT) (tri3b). Rezultate modela GMM-HMM smo uporabili za osnovo učenja hibridnega modela DNN-HMM. Skrita stanja v HMM in verjetnosti za prehode med njimi smo zamrznili. Za napovedovanje fonemov smo, namesto GMM, uporabili globoke nevronske mreže (DNN). Uporabili smo več različnih konfiguracij globokih nevronskih mrež.

GMM-HMM

Monofonski model smo učili le na podmnožici učnih podatkov. Vzeli smo le 20.000 najkrajših izrekov. Razlog za učenje na krajših izrekih je, da smo zagotovili boljše ujemanje med zapisom besed in zvočnim posnetkom le-teh. V učnih podatkih nimamo zapisa, kje natančno znotraj zvočnega posnetka se nahajajo posamezne besede, niti kje natančno so posamezni fonemi znotraj besede. Za dober model moramo vsak fonem opisati zgolj s tistim delom posnetka, kjer je fonem izrečen. To skušamo ugotoviti med samim učenjem, kar je bolj zanesljivo pri krajših posnetkih. Po vsaki fazi učenja smo z Viterbijevim algoritmom čim bolje poravnali zvočne posnetke s fonemi, oziroma v naslednjih fazah s trifoni. Te poravnave uporabimo kot osnovo pri učenju naslednje faze (slika 3).



Slika 3: **Shema posameznih faz učenja, za vsako izluščimo značilke in učimo verjetnosti skritih stanj v HMM. Hkrati določimo poravnave fonemov z značilkami (Pn), s katerimi poravnamo foneme in značilke pred učenjem naslednje faze, kjer na novo učimo vejretnosti istega HMM.**

Pri preostalih fazah učenja akustičnega modela smo uporabili celotno učno množico. V učni množici ne nastopajo vsi možni trifoni, prav tako je teh veliko, zato jih uredimo v odločitveno drevo, glede na lastnosti posameznih fonemov v trifonu (samoglasnik ali soglasnik, in podobno). Z vsako naslednjo fazo smo povečali število Gaussovih porazdelitev (komponent v GMM) in število stanj, oziroma listov v našem odločitvenem drevesu, saj so modeli čedalje kompleksnejši. Poizkusili smo več različnih parametrov in primerjali rezultate na validacijski množici. Rezultati so bili zelo podobni, občutno izboljšanje smo dosegli le pri močno povečanem številu stanj, predvsem pa številu komponent v GMM. Zaradi bojazni, da bi s tem model prenaučili, smo se odločili za manjše število Gaussovih porazdelitev in manj listov v drevesu. Uporabljeni parametri so navedeni v tabeli 2. Pri monofonskem modelu je odločitveno drevo trivialno, navedemo le število Gaussovih porazdelitev za primerjavo.

Tabela 2: **Parametri odločitvenega drevesa pri različnih akustičnih modelih**

Učni model	Število komponent v GMM	Število listov	Razmerje med št. komponent in št.listov
mono	1000	/	/
tri1	12000	2000	6,00
tri2b	20000	3000	6,67
tri3b	30000	3500	8,57

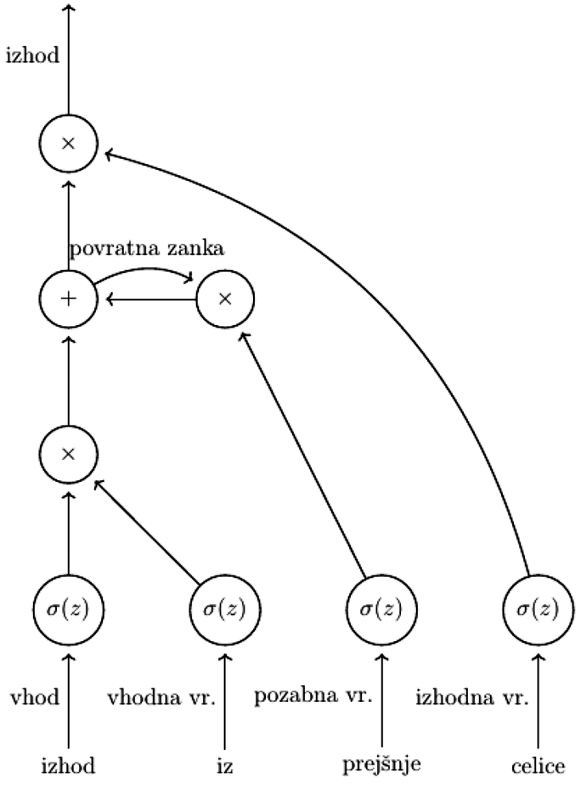
DNN-HMM

Pred učenjem globoke nevronske mreže smo umetno povečali količino učnih podatkov tako, da smo spremenili hitrost posnetkov. Originalno hitrost smo pomnožili s faktorji 0,9, 1,0 in 1,1 ter dobili trikrat toliko podatkov. Dodatno smo zmanjšali izhodno vzorčno frekvenco modela na tretjino, tako da je vsak vzorec sestavljen iz treh podvzorcev. Efektivno

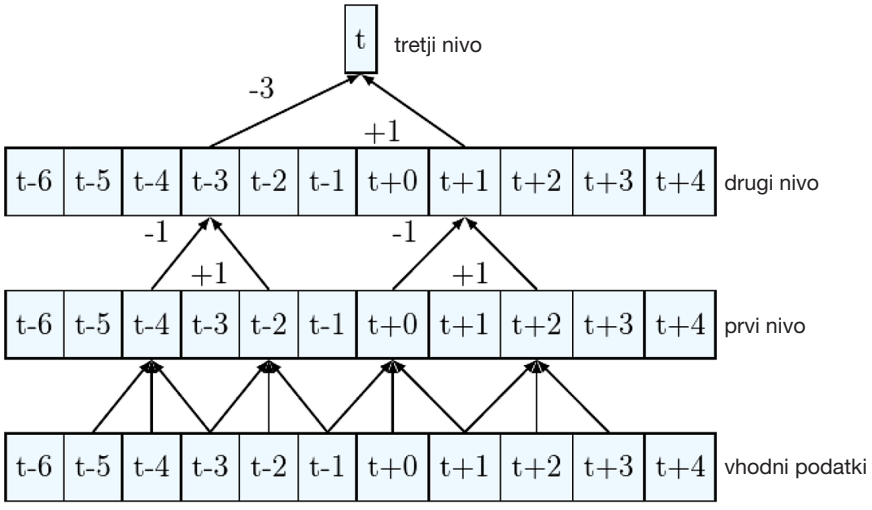
izhod nevronske mreže ovrednotimo na vsakem tretjem (originalnem) vzorcu oziroma časovnem oknu. Nato zamaknemo podatke za en podvzorec in učimo ponovno. Končno zamaknemo podatke za še en podvzorec (skupno za dva podvzorca) in spet učimo. Nevronske mreže smo učili 5 dob, kar z zamikanjem vzorcev skupno nanese 15 dob. Za učenje nevronskih mrež smo povečali spektralno ločljivost. Število značilk MFCC smo dvignili s 13 na 40. Vse mreže imajo tako vhodni nivo dimenzije 40. Napovedujemo stanja v HMM, ki predstavljajo trifone, oziroma konkretneje liste odločitvenega drevesa. Teh stanj je po zadnji GMM-HMM fazi 2848, zato je taka tudi dimenzija izhodnega nivoja. Za optimizacijski algoritem smo uporabili stohastični gradientni spust, za funkcijo izgube pa logaritem verjetnosti pravilne sekvence fonemov. Za potrebe regularizacije smo dodali še en izhodni nivo enake dimenzije, kjer smo uporabili križno entropijo za funkcijo izgube. Pri učenju nastopata oba izhodna nivoja, pri napovedovanju pa zgolj prvi.

V zadnjem času so se v praksi uveljavile mreže LSTM, ki rešujejo težavo izginjajočega gradienta (Hochreiter & Schmidhuber, 1997). Mreža LSTM je sestavljena iz posameznih celic, ki so med seboj povezane s povezavami naprej in s povratnimi povezavami. Vsaka posamezna celica je sestavljena iz več enot (slika 4). Poleg vhoda, aktivacijske funkcije in izhoda ima celica tudi povratno zanko znotraj celice ter troje vrat, ki utežujejo posamezne dele celice. Vhodna vrata dajo utež vhodnim podatkom celice. Izhodna vrata dajo utež izhodu iz celice. Pozabna vrata utežijo povratno zanko znotraj celice. Vsa vrata imajo sigmoidno aktivacijsko funkcijo, podatki na vhodu pa imajo poljubno nelinearno aktivacijsko funkcijo (Goodfellow, Bengio, & Courville, 2016); v našem primeru smo uporabili hiperbolični tangens (tanh).

TDNN so vrsta nevronskih mrež s povezavami naprej, vendar se učijo na časovno širšem kontekstu. Skriti nivoji v globoki nevronski mreži združujejo informacijo iz prejšnjega nivoja, tako da nevroni v vsakem naslednjem nivoju upoštevajo večji časovni razpon. Na primer, če na vhodnem nivoju vsak nevron predstavlja eno časovno okno, bo nevron na prvem skritem nivoju predstavljal pet časovnih oken. Združil bo informacijo enega okna s po dvema predhodnima in dvema naslednjima oknoma (slika 5). Nevron na drugem skritem nivoju na primer združi



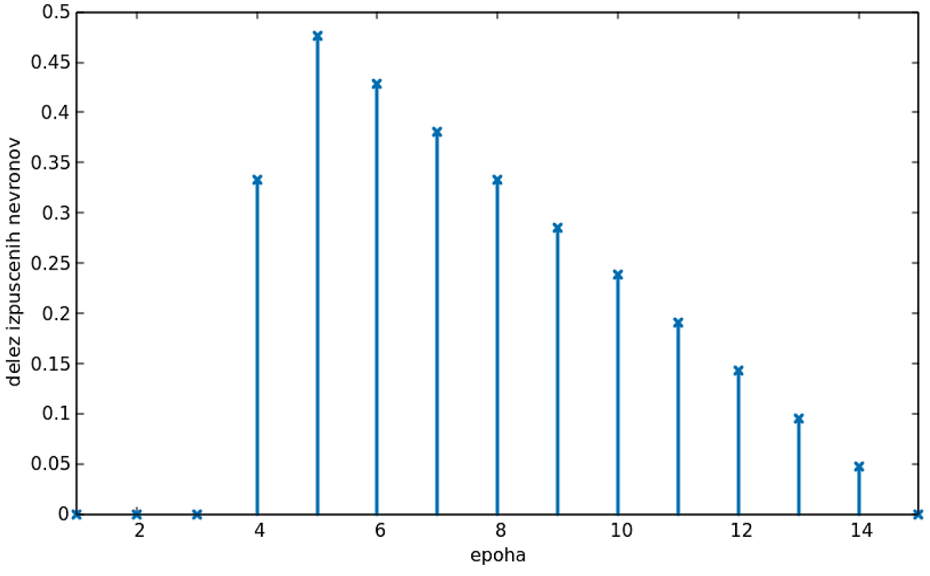
Slika 4: **Sestava ene celice mreže LSTM**.



Slika 5: **Primer časovnega združevanja v TDNN. Na prvem nivoju se združi informacija pri časovnih indeksih t-1, t in t+1, glede na vhod. Na drugem nivoju se združi informacija pri časovnih indeksih t-1 in t+1 prvega nivoja. Na tretjem nivoju pri indeksih t-3 in t+1. Vsak nevron v tretjem nivoju vsebuje informacije devetih časovnih okvirjev vhodnih podatkov (med t-5 in t+3).**

informacijo štirih nevronov iz prvega skritega nivoja: enega pri istem časovnem indeksu ter še enega z večjim in dva z manjšim časovnim indeksom (ali obratno) (Peddinti, Povey, Khudanpur, 2015).

Preizkusili smo več različnih konfiguracij mrež z različnim številom skritih nivojev in z različno povezanimi nivoji. Osredotočili smo se na arhitekturi TDNN in LSTM. Prva časovno zakasnjena mreža (*tdnn_1a*) na prvem skritem nivoju (nivo lda) združi časovni kontekst petih okvirjev pri časovnih indeksih, ki se razlikujejo za -2, -1, 0, 1 in 2 glede na opazovan časovni indeks. Nivo je polno povezan in je dimenzije 40. Sledi osem polno povezanih nivojev (nivoji tdnn1-8) dimenzije 512, kjer uporabimo izpuščanje nevronov (angl. dropout) (Hinton in sod., 2012). Za dano moč izpuščanja α definiramo masko dimenzije 512, kjer ima vsak element maske naključno vrednost iz intervala [1-2α, 1+2α]. V vsakem časovnem okvirju pomnožimo izhode nevronov z enako masko (Povey, 2018). Moč izpuščanja α se spreminja tekom učenja in je prikazana na sliki 6. Izpuščanje nevronov je prisotno v nivojih tdnn1-8. Za aktivacijsko funkcijo uporabimo ReLU. V petih izmed osmih nivojev TDNN združimo časovni kontekst treh različnih okvirjev iz



Slika 6: **Spreminjanje »deleža izpuščenih nevronov« oziroma vrednosti α tekom učenja.**

Tabela 3: **Združevanje po času pri mreži tdnn_1a - upoštevani so našteti časovni indeksi prejšnjega nivoja, glede na trenuten časovni indeks *t***

Nivo	Časovni indeksi
lda	t-2, t-1, t+0, t+1, t+2
tdnn2	t-1, t+0, t+1
tdnn4	t-1, t+0, t+1
tdnn6	t-3, t+0, t+3
tdnn7	t-3, t+0, t+3
tdnn8	t-6, t-3, t+0

Slika 7: **Shema konfiguracija tdnn_1a z dvema izhodoma pri učenju, xent pomeni, da ta veja uporablja križno entropijo za funkcijo izgube.**

prejšnjega nivoja, kot je prikazano v tabeli 3. Sledita dva vzporedna polno povezana nivoja dimenzije 512 z aktivacijsko funkcijo ReLU, brez izpuščanja in nato izhodna nivoja (slika 7). Konfiguracijo smo povzeli po Kaldijevem projektu *vystadial_cz* za češki jezik (Denisov, 2018).

Časovno zakasnjena mreža *tdnn_1c* je podobna mreži *tdnn_1a*, le da vsebuje šest nivojev TDNN. Časovno združevanje je enako, le da pri tej konfiguraciji poteka pri nivojih lda, tdnn2, tdnn3, tdnn4, tdnn5 in tdnn6. Časovno zakasnjena mreža *tdnn_1b* je identična mreži *tdnn_1c*, le da ne uporabljamo izpuščanja nevronov. Časovno zakasnjeni mreži *tdnn_1d* in *tdnn_1e* vsebujeta 10 nivojev TDNN, kjer uporabljamo izpuščanje nevronov. Pri mrežah *tdnn_1d* in *tdnn_1e* smo preizkusili asimetrično združevanje podatkov po času še v dveh nivojih poleg zadnjega (glej tabelo 4). Nevronske mreže *tdnn_1a*, *tdnn_1b*, *tdnn_1c* in *tdnn_1d* smo učili 5 dob. Uporabili smo privzete Kaldijeve parametre paralelizacije, nastavili smo le število začetnih vzporednih nalog na 2 in število končnih nalog na 12. Pri nevronski mreži *tdnn_1e* smo oba parametra števila nalog nastavili na 1, kar naj bi bilo bolj optimalno, saj smo za učenje uporabljali le eno grafično kartico. Število dob smo zmanjšali na 3, tako da smo približno ohranili število iteracij in čas učenja. Število iteracij je bilo pri konfiguraciji *tdnn_1e* nekoliko večje kot pri konfiguraciji *tdnn_1d*, čas učenja pa daljši kljub manjšemu številu dob in enakim skritim nivojem.

Tabela 4: **Združevanje po času pri mrežah tdnn_1d in tdnn_1e - upoštevani so našteti časovni indeksi prejšnjega nivoja, glede na trenuten časovni indeks t.**

Nivo	Časovni indeksi
lda	t-1, t+0, t+1
tdnn2	t-2, t+0, t+1
tdnn4	t-1, t+0, t+2
tdnn6	t-3, t+0, t+3
tdnn8	t-3, t+0, t+3
tdnn10	t-6, t-3, t+0

Prva mreža LSTM (*lstm_1b*) ima enak vhodni nivo kot mreže TDNN, prav tako ima enak prvi skriti nivo (*lda*). Sledijo štirje nivoji tipa LSTMP, dimenzije 1024, in izhodni nivo. Nivo LSTMP se od LSTM razlikuje v tem, da izhod nivoja ne pelje nazaj na vhod istega nivoja, ampak sta vmes dva vzporedna projekcijska nivoja. Izhod enega projekcijskega nivoja pelje na vhod nivoja LSTM, izhod drugega projekcijskega nivoja pa na vhod naslednjega skritega nivoja. Dimenzija projekcijskih nivojev je 256. Druga mreža LSTM (*lstm_1c*) ima po prvem skritem nivoju šest nivojev LSTM (ne LSTMP) dimenzije 512. Pri *lstm_1b* in lstm_1c ni izpuščanja nevronov. Obe mreži LSTM smo učili 4 dobe.

Globoke nevronske mreže smo učili na grafični kartici Nvidia Tesla P100, ostale modele pa na procesorju z 8 nitmi. Učenje posamezne TDNN je trajalo približno 16 ur, LSTM približno 20 ur, vseh predhodnih faz na procesorju pa skupno 2-3 dni.

4.2 Jezikovni model

Jezikovni modeli so v obliki končnih pretvornikov (angl. finite-state transducer - FST) (Mohri, 1997). Naučimo jih z uporabo n-gramov, to so skupine n besed, ki se zaporedno pojavijo v besedilu. Na primer, v stavku: »Članek potrebuje pozornega bralca«, imamo štiri unigrame (posamezne besede), tri bigrame (»članek potrebuje«, »potrebuje pozornega«, »pozornega bralca«), dva trigrama in en štirigram.

N-gramski jezikovni model ocenjuje verjetnost, da neka beseda w_i sledi danim n-1 predhodnim besedam. Uporabili smo 3-grame z Witten-Bellovim glajenjem (Witten, Bell, 1991). Witten-Bellovo glajenje ocenjuje verjetnost pojavitve besede w_i, glede na predhodnih n-1 besed v n-gramu, $p(w_i | w_{i-n+1}^{i-1})$ z enačbo:

$$p(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) + N_{1+}(w_{i-n+1}^{i-1}\bullet)p(w_i|w_{i-n+2}^{i-1})}{\sum_{w_i} c(w_{i-n+1}^{i}) + N_{1+}(w_{i-n+1}^{i-1}\bullet)} \qquad (5),$$

kjer je $c(w_{i-n+1}^{i-1})$ število pojavitev besed w_{i-n+1}^{i-1}, • poljubna beseda, N_{1+} pa število besed •, ki se v n-gramu w_{i-n+1}^{i-1} • pojavi vsaj enkrat (Chen, Goodman, 1998). Slednje lahko drugače zapišemo kot:

$$N_{1+}(w_{i-n+1}^{i-1}\bullet) = |\{w_i : c(w_{i-n+1}^{i} w_i) > 0\}| \qquad (6),$$

torej velikost množice besed w_i, za katere velja, da je število pojativev n-grama $w_{i-n+1}^{i} w_i$ (tj. n-1-gram w_{i-n+1}^{i}, ki mu sledi beseda w_i) večje od nič.

Uteži končnih pretvornikov ponovno ocenimo z uporabo globokih nevronskih mrež. Preizkusili smo več različnih konfiguracij časovno zakasnjene mreže in mreže LSTM. Model s 3-grami smo naučili na besedilnem korpusu ccGigafida, modele z nevronskimi mrežami pa na besedilnem korpusu ccKres (Logar in sod., 2013)). Za korpus ccKres smo se odločili, ker so različni viri besedil (internet, revije, časopisi, leposlovje, ...) bolj enakomerno zastopani kot pri korpusu ccGigafida. Prav tako je korpus ccKres manj obširen (približno 10 milijonov besed), kar je pohitrilo učenje z več kot en dan na nekaj ur.

V razdelku 5 ovrednotimo tri konfiguracije nevronskih mrež, ki smo jih uporabili za učenje jezikovnega modela. Konfiguracijo *rnnlm_1a* sestavlja pet skritih nivojev, od tega trije nivoji TDNN in dva nivoja LSTM. Dimenzija vsakega nivoja je 800. Prvi skriti nivo je TDNN, kjer združimo časovna indeksa t in t-1. Sledi nivo LSTMP. Dimenzija projekcijskih nivojev je 200. Tretji skriti nivo je TDNN, kjer združimo časovna indeksa t in t-2, sledi še en enak nivo LSTMP. Zadnji skriti nivo je TDNN z združevanjem časovnih indeksov t in t-1. Opisano nevronsko mrežo smo učili 10 dob.

Konfiguracijo *rnnlm_1b* sestavljajo trije skriti nivoji tipa TDNN. V prvem in drugem skritem nivoju združimo časovna indeksa t in t-1, v tretjem skritem nivoju pa časovna indeksa t in t-2. Vsi nivoji so dimenzije 800. Nevronsko mrežo smo učili 15 dob. Konfiguracijo *rnnlm_1c* sestavljajo trije skriti nivoji tipa LSTM. Vsi nivoji so dimenzije 512. Nevronsko mrežo smo učili 15 dob.

Nevronske jezikovne modele smo učili na enaki grafični kartici kot akustične. Učenje je trajalo 3 do 5 ur.

5 REZULTATI IN ANALIZA

Uspešnost modelov pri razpoznavanju govora smo izmerili na validacijski in testni množici. Uspešnost podajamo kot delež besed, ki jih moramo dejanskemu besedilu dodati (vriniti), izbrisati in zamenjati, da bi dobili razpoznano besedilo. Ta mera za uspešnost se imenuje WER (besedna stopnja napak, angl. word error rate) in jo izračunamo kot:

$$WER = (V + I + Z) / B, \qquad (7)$$

kjer V predstavlja število vrivanj, I število izbrisov, Z število zamenjav in B število vseh izgovorjenih besed v dejanskem besedilu.

Validacijska množica je sestavljena iz dveh korpusov, Gos (dev_gos) in Sofes (dev_sofes). Zaradi posebnosti obeh validacijskih množic, rezultate ločeno predstavljamo za vsak korpus posebej. Množica

dev_sofes vsebuje veliko tujih lastnih imen, predvsem imen letališč in mest, zaradi česar pričakujemo slabše rezultate. Množica dev_gos vsebuje nekaj izrekov, kjer nastopata dva govorca hkrati. Transkripcija korpusa Gos zapiše dva ločena stavka, ki pa se navezujeta na isti zvočni posnetek. Ločena stavka moramo združiti v en izrek, pri čemer ne vemo, v kakšnem vrstnem redu sta stavka izrečena. Možno je tudi, da govorca govorita hkrati. V teh primerih se transkripcija ne ujema z zvočnim posnetkom. Primer takega izreka je v tabeli 5. Rezultati v teh primerih

Tabela 5: **Primerjava treh transkripcij izreka iz korpusa Gos, brez ločil. Primerjamo transkripcijo zapisano v korpusu Gos (T1), našo ročno transkripcijo (T2) in transkripcijo modela tdnn_1a (T3)**

T1	»to je to je še pred volitvami bilo ja in sem takrat deloval kot državni sekretar na Ministrstvu za notranje zadeve kar v bistvu ni bila politična funkcija«
T2	»in sem takrat deloval kot državni sekretar na ministrstvu za notranje zadeve kar TO TO JE ŠE PRED VOLITVAMI JA politična funkcija«
T3	»in sem takrat deloval kot državni sekretar na ministrstvu za notranje zadeve kako je biti še pred volitvami je policija«

prikažejo slabše stanje od dejanskega.

Če ovrednotimo transkripcijo T3 iz tabele 5 glede na transkripcijo T1 dobimo napako WER=85,0%, glede na transkripcijo T2 pa WER=35,0%. Pri tem nismo upoštevali malih in velikih začetnic.

Metode, opisane v razdelku 4, smo ovrednotili na omenjenih validacijskih množicah in na testni množici. Primerjali smo rezultate pri različnih utežeh med jezikovnim in akustičnim modelom. Z večanjem uteži jezikovnega modela, narašča število vrivanj, vendar pada število izbrisov. Število zamenjav se pri množici dev_sofes ne spreminja dosti v odvisnosti od uteži, pri množici dev_gos pa rahlo pada. Rezultate navajamo pri uteži, ki da najboljše rezultate na posamezni validacijski množici, rezultate na testni množici pa pri uteži, ki da najboljše rezultate na uniji obeh validacijskih množic. Najboljši rezultat je dosegla kombinacija akustičnega modela *tdnn_1a* in jezikovnega modela *rnnlm_1a*, kjer na testni množici dosežemo WER=27,16%.

V primerjavi z GMM-HMM, so vsi akustični modeli, naučeni z nevronskimi mrežami, dosegli boljše rezultate, razen konfiguracije *tdnn_1e*, kjer je rezultat na validacijski množici dev_sofes slabši. Glede na to, da je zgradba nevronske mreže v *tdnn_1e* enaka tisti v *tdnn_1d*, sklepamo, da je nismo dovolj dolgo učili, čeprav je bil čas učenja daljši. Mreža *tdnn_1d* doseže za 8,21% boljši rezultat na množici dev_gos kot mreža *tdnn_1e*. Najboljša časovno zakasnjena mreža je mreža *tdnn_1a*, ki na validacijskih množicah pravilno razpozna 7,42%, oziroma 17,39% več besed kot model GMM-HMM. Na testni množici je izboljšanje 14,41% (tabela 6). Mreže LSTM so prinesle manjše izboljšanje od časovno zakasnjenih mrež. Najboljša mreža LSTM, v primerjavi z najboljšo mrežo TDNN, pravilno razpozna 2,95% manj besed na množici dev_gos in 4,16% manj besed na testni množici. Pravilno pa razpozna 0,46% več besed na množici dev_sofes (tabela 6). Tu je razlika zelo majhna, zato ne moremo sklepati o prednosti mreže LSTM za praktično rabo.

Razlike v uspešnosti med jezikovnimi modeli naučenimi z nevronskimi mrežami so zelo majhne. Pri obeh validacijskih množicah in pri testni množici so razlike v deležu pravilno razpoznanih besed med nevronskimi jezikovnimi modeli manjše od 0,5%. Najboljši nevronski jezikovni model *rnnlm_1a* v primerjavi s 3-gramskim jezikovnim modelom pravilno razpozna 1,96% več besed na množici dev_gos, 4,71% več besed na množici dev_sofes in 2,61% več besed na testni množici.

Tabela 6: **Rezultati (WER) akustičnih modelov GMM-HMM (tri3b), najboljše mreže TDNN (tdnn_1a) in najboljše mreže LSTM (lstm_1b) pri 3-gramskem jezikovnem modelu ter akustičnega modela TDNN pri najboljšem nevronskem jezikovnem modelu (rnnlm_1a).**

model	dev_gos	dev_sofes	test
tri3b	70,69 %	36,72 %	44,18 %
tdnn_1a	53,30 %	29,30 %	29,77 %
lstm_1b	56,25 %	28,84 %	33,93 %
tdn_1a+rnnlm_1a	51,34 %	24,59 %	27,16 %

Rezultate našega sistema smo primerjali z Googlovim vmesnikom Google Cloud speech-to-text (2018). Naključno smo izbrali pet izrekov iz testne množice, jih prepisali z Googlovim vmesnikom in transkripcije primerjali z transkripcijami našega razpoznavalnika. Napaka pri izbranih izrekih je WER=33,33% za Googlov sistem in WER=21,28% za naš sistem, kjer smo uporabili akustični model *tdnn_1a* in jezikovni model *rnnlm_1a*. Primerjava ni ravno reprezentativna, ker smo uporabili le 5 izrekov iz testne množice. Preliminarni rezultati kažejo, da naš model deluje bolje, vendar je potrebno več dodatnega testiranja, da

bi dobili statistično zanesljive rezultate. Glede na trenutno videno ima Googlov sistem manj vrivanj kot naš sistem, kar je prednost pri obotavljanju govorca. Mašila, kot sta »hmm« ali »eee« in prekinjene besede Googlov razpoznavalnik izpusti, medtem ko jih naš razpoznavalnik napačno zazna kot neke druge besede. Prednost našega razpoznavalnika v primerjavi z Googlovim je, da ima manjše število izbrisov in zamenjav. Torej zazna večje število izgovorjenih besed in te tudi bolj pravilno prepozna.

Podrobneje je Googlov vmesnik preizkusil David Čefarin (2016) v svojem diplomskem delu. Ugotovil je, da sistem dosega 61,72% točnost, oziroma WER=38,28% pri prostem govoru. Naš razpoznavalnik ima napako WER=27,16%, vendar omenjenih točnosti, oziroma napak Googlovega in našega sistema tudi tu ni moč neposredno primerjati med seboj, saj smo našega ovrednotili na drugi testni množici.

Naš razpoznavalnik je bil izdelan z namenom razpoznavanja splošnega, vsakdanjega govora. Pri učenju smo uporabili široko besedišče in veliko število govorcev, tako da razpoznavalnik ni specializiran za specifično področje uporabe (npr. za medicino ali za glasovno upravljanje nekega programa) niti ni prilagojen na posameznega govorca. Prednost tega je, da razpoznavalnik dosega podobne rezultate ne glede na vsebino govora ali na to, čigav govor razpoznavamo. Slabost te splošnosti pa je, da dosega slabše rezultate od specializiranih razpoznavalnikov, ki so omejeni zgolj na ozko besedišče ali enega govorca. Pred razpoznavanjem moramo govor shraniti v zvočno datoteko, obdelava posameznih datotek pa je daljša, kot če bi sproti brali zvočni signal z mikrofona.

6 SKLEPNE UGOTOVITVE

Predstavili smo sistem za razpoznavanje slovenskega govora z metodami globokih nevronskih mrež. Za akustični model smo uporabili hibridni sistem DNN-HMM, kjer uporabimo globoke nevronske mreže za napovedovanje skritih stanj v HMM. Časovno zakasnjene nevronske mreže so se izkazale kot bolj uporabne od mrež z dolgim kratkoročnim spominom, saj dosegajo nekoliko boljše rezultate, obenem pa sta učenje in prepis hitrejša. Jezikovni model smo naučili z metodo n-gramov in z različnimi globokimi nevronskimi mrežami. Model z n-grami dosega slabšo natančnost, vendar je mnogo hitrejši pri prepisu. Napovedovanje (s procesorjem, brez uporabe grafične kartice) z akustičnim nevronskim modelov in n-gramskim jezikovnim modelom traja približno tako dolgo, kot je dolžina posnetka, ki ga prepišemo. Če dodamo nevronski jezikovni model, se čas napovedovanja podaljša za do 100%. Med modeli z globokimi nevronskimi mrežami ni velikih razlik v uspešnosti. Naš sistem dosega boljše rezultate pri razpoznavanju tekočega govora od primerljivih razpoznavalnikov za slovenščino.

Trenutna verzija razpoznavalnika je uporabna na vseh področjih, kjer visoka pravilnost razpoznavanja ni ključnega pomena. Pri učenju smo uporabili široko besedišče, zato je uporaben pri različnih tematikah, tako strokovnih kot pri vsakdanjem govoru. Slabost razpoznavalnika je njegova hitrost, saj nima možnosti sprotnega razpoznavanja govora.

Kljub uspešnemu razpoznavanju je možnosti za izboljšave še precej. Na točnost razpoznavanja vplivata velikost učnih podatkov in arhitektura razpoznavalnika. Smiselno bi bilo vključiti več govornih in besedilnih korpusov, prav tako bi lahko drugače obravnavali težave korpusa Gos. Problematične izreke bi lahko popolnoma izpustili iz učne množice ali jih transkribirali na novo. Pri učenju jezikovnega modela bi uspešnost lahko izboljšali z upoštevanjem morfoloških podatkov in z uporabo večjega besedilnega korpusa. Obstaja veliko različnih možnih konfiguracij nevronskih mrež. V našem delu gotovo nismo našli najbolj optimalne, saj je možnosti preveč, učenje pa dolgotrajno.

Naš sistem bi lahko izboljšali z uporabo značilk iVector, ki nekoliko izboljšajo točnost razpoznavanja. Največja prednost teh značilk je, da se uporabljajo pri tekočem razpoznavanju govora. To pomeni, da ni potrebno vnaprej posneti celotnega zvočnega posnetka, ampak govor razpoznavamo sproti, med snemanjem.

Razpoznavalnik bi lahko razširili s sistemom, ki v besedilu avtomatsko postavi ločila. Tako besedilo bi bilo lažje berljivo in manj dvoumno, če bi tak sistem dobro naučili. Prepisan govor bi potreboval manj ročnega urejanja, saj moramo pri uporabi našega razpoznavalnika ločila ročno dodati.

Uporabniško izkušnjo bi lahko izboljšali z aplikacijo, ki z mikrofonom snema govor, sproti shranjuje posnetke in jih obdela za uporabo razpoznavalnika govora. Aplikacijo bi lahko uporabili tudi za že obstoječe posnetke govora.

Programska koda našega razpoznavalnika je dostopna na spletnem naslovu https://github.com/

MatejUlcar/kaldi/tree/slovenscina/egs/slovenscina. Načrtujemo, da bomo vsaj nekatere od zgornjih možnosti za izboljšave implementirali v sistem in izboljšano verzijo skupaj z najboljšimi naučenimi modeli objavili na repozitoriju Clarin.si.

LITERATURA

[1] Alam, J., Kinnunen, T., Kenny, P., Ouellet, P., O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech Communications*, 55:237–251.

[2] Bolka, A. (2016). *Samodejno razpoznavanje fonemov slovenskega govora z uporabo zbirke orodij Kaldi*. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko.

[3] Chen, S. F., Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. *Harvard Computer Science Group Technical Report TR-10-98.*

[4] Charan, R., Manisha, A., Karthik, R., Kumar, M. R. (2017). A Text-independent Speaker Verification Model: A Comparative Analysis. *2017 International Conference on Intelligent Computing and Control.*

[5] Čefarin, D. (2016). *Preizkus Googlovega govornega programskega vmesnika za slovenski govorni jezik*. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko.

[6] Davis, S., Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllaic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4)*, strani 357-366.

[7] Denisov, P. (2018). https://github.com/kaldi-asr/kaldi/blob/master/egs/vystadial_cz/s5b/local/chain/tuning/run_tdnn_1a.sh, dostopano 6. 8. 2018.

[8] Dobrišek, Simon; Žganec Gros, Jerneja; Žibert, Janez; Mihelič, France and Pavešić, Nikola, 2017, *Speech Database of Spoken Flight Information Enquiries SOFES 1.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1125.

[9] Dobrovoljc, Kaja; Krek, Simon; Holozan, Peter; Erjavec, Tomaž and Romih, Miro, 2015, *Morphological lexicon Sloleks 1.2*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1039.

[10] Donaj, G. (2015). *Avtomatsko razpoznavanje govora za pregibni jezik z uporabo morfoloških jezikovnih modelov s kontekstno odvisno strukturo*. Doktorska disertacija, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.

[11] Gales, M. (1998). Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. *Computer Speech and Language*, vol. 12, strani 75-98.

[12] Gales, M. (1999). Semi-tied Covariance Matrices for Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing,* vol. 7, strani 272-281.

[13] Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research*, 57:345–420.

[14] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[15] Google. *Google cloud text-to-speech*. https://cloud.google.com/speech-to-text/, dostopano 18. 9. 2018.

[16] Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Estève, Y. (2018). TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. *20th International Conference, SPECOM 2018*, strani 198-208.

[17] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. (2012). Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. *ArXiv preprint arXiv:1207.0580.*

[18] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9, strani 1735-1780.

[19] Logar, Nataša; Erjavec, Tomaž; Krek, Simon; Grčar, Miha and Holozan, Peter, 2013, *Written corpus ccGigafida 1.0*, Slovenian language resource repository CLARIN.SI, *http://hdl.handle.net/11356/1035*.

[20] Logar, Nataša; Erjavec, Tomaž; Krek, Simon; Grčar, Miha and Holozan, Peter, 2013, *Written corpus ccKres 1.0*, Slovenian language resource repository CLARIN.SI, *http://hdl.handle.net/11356/1034*.

[21] Mohri, M. (1997). Finite-state Transducers in Language and Speech Processing. *Computational Linguistics*, vol. 23, issue 2, strani 269-311.

[22] Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. *Sixteenth Annual Conference of the International Speech Communication Association*, strani 3214-3219.

[23] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Catalog No.: CFP11SRW-USB.

[24] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., Khudandpur, S. (2018). Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. *Interspeech 2018* 3743-3747.

[25] Rao, K., Peng, F., Sak, H., Beaufays, F. (2015). Grapheme-to-Phoneme Conversion Using Long Short-Term Memory Recurrent Neural Networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, strani 4225–4229.

[26] Sak, H., Senior, A., Rao, K., Beaufays (2015a). Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. *16th Annual Conference of the International Speech Communication Association* (INTERSPEECH 2015), strani 1468-1473.

[27] Sak, H., Senior, A., Rao, K., Beaufays, F., Schalkwyk, J. (2015b). Google voice search: faster and more accurate. *Google AI Blog.* https://ai.googleblog.com/2015/09/google-voice-search-faster-and-more.html, dostopano 10. 9. 2018.

[28] Stuttle, M. N. (2003). *A Gaussian Mixture Model Spectral Representation for Speech Recognition*. Doktorska disertacija, Hughes Hall and Cambridge University Engineering Department.

[29] Verdonik, Darinka; Potočnik, Tomaž; Sepesy Maučec, Mirjam and Erjavec, Tomaž, 2017, *Spoken corpus Gos VideoLectures 2.0 (transcription)*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1158.

[30] Witten, I. H., Bell, T. C. (1991). The Zero-Frequency Problem: Estimating the probabilities of Novel Evenets in Adaptive Text Compression*. IEEE Transactions on Information Theory, vol. 37, no. 4*, julij 1991.

[31] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G. (2017). The Microsoft 2016 conversational speech recognition system. *IEEE International Conference on Acoustics, Speech and Signal Processing*, strani 5255–5259.

[32] Zwitter Vitez, Ana; Zemljarič Miklavčič, Jana; Krek, Simon; Stabej, Marko and Erjavec, Tomaž, 2013, *Spoken corpus*

Gos 1.0, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1040.

[33] Žgank, A., Donaj, G., Sepesy Maučec, M. (2014). Razpoznavalnik tekočega govora UMB Broadcast News 2014: kakšno vlogo igra velikost učnih virov? V: *Zbornik 9. konference Jezikovne tehnologije, Informacijska družba - IS 2014*, strani 147-150.

[34] Žgank, A., Verdonik, D., Sepesy Maučec, M. (2016). Razpoznavanje tekočega govora v slovenščini z bazo predavanj SI TEDx-UM. V: *Zbornik konference Jezikovne tehnologije in digitalna humanistika*, strani 186-189.

▪

Matej Ulčar je leta 2018 magistriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je zaposlen kot raziskovalec.Ukvarja se z razpoznavanjem govora in medjezikovnimi tehnologijami, predvsem z vektorskimi vložitvami besed.

▪

Simon Dobrišek je izredni profesor in predstojnik Laboratorija za strojno inteligenco na Fakulteti za elektrotehniko Univerze v Ljubljani. Raziskovalno deluje na širšem področju tehnologij govorjenega jezika, razpoznavanja vzorcev, biometrije in umetnih inteligentnih sistemov. Posveča se tudi interdisciplinarnim raziskavam, ki segajo na področje jezikoslovja in glasoslovja ter varstva zasebnosti pri uporabi informacijskih in komunikacijskih tehnologij. V zadnjih letih je sodeloval pri več nacionalnih in mednarodnih raziskovalnih projektih s področja razvoja biometričnih tehnologij in tehnologij govorjenega jezika ter s področja etike in pravnega urejanja uporabe nadzornih tehnologij.

▪

Marko Robnik-Šikonja je redni profesor in predstojnik Katedre za umetno inteligenco Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno se ukvarja s področji umetne inteligence, strojnega učenja, obdelave naravnega jezika in analize omrežij. Je avtor več kot 100 znanstvenih publikacij, ki so bile citirane več kot 4000-krat.