

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNÁVANIE REČI V ZJEDNODUŠENOM
ANGLICKOM JAZYKU
BAKALÁRSKA PRÁCA

2019
DÁVID ŠUBA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ROZPOZNÁVANIE REČI V ZJEDNODUŠENOM
ANGLICKOM JAZYKU
BAKALÁRSKA PRÁCA

Študijný program: 2511 Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: prof. Ing. Igor Farkaš, Dr.

Bratislava, 2019
Dávid Šuba



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Dávid Šuba
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Rozpoznávanie reči v zjednodušenom anglickom jazyku
Speech recognition in simplified English

Anotácia: V interakcii človeka s robotickým systémom vzniká prirodzená potreba komunikovať v prirodzenom jazyku, často v nejakej konkrétnej doméne. V súčasnosti existuje niekoľko systémov, najmä pre angličtinu, ktoré sa dajú pre taký účel použiť, prípadne dotrénovať pre potreby užívateľa s cieľom maximalizovať presnosť rozpoznania slov, nezávisle od hovoriaceho.

Cieľ:

1. Naštudujte si problematiku rozpoznávania reči, princíp skrytých markovovských reťazcov a umelých neurónových sietí, a oboznámte sa so systémom HTK Toolkit.
2. Dotrénujte a otestujte systém HTK na vami pripravenej dátovej množine (oznamovanie a príkazové anglické vety týkajúce sa opisu objektov na scéne, viacero hovoriacich).
3. Doprogramujte potrebné skripty potrebné pre nasadenie systému do prevádzky.

Literatúra: Jurafsky D., Martin J. (2008) *Speech and Language Processing*, Pearson International Edition.
HTK Toolkit, Cambridge, UK, <http://htk.eng.cam.ac.uk/>

Vedúci: prof. Ing. Igor Farkaš, Dr.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 24.10.2018

Dátum schválenia: 24.10.2018

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné prehlásenie: čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím uvedenej literatúry.

V Bratislave dňa:

.....
Dávid Šuba

Pod'akovanie: Ďakujem.

Abstrakt

Abstrakt.

Kľúčové slová: rozpoznávanie reči, HTK, skryté markovovské modely

Abstract

Abstract.

Keywords: speech recognition, HTK, hidden markov models

Obsah

Úvod	1
1 Úvod do problematiky	2
1.1 Úvod do rozpoznávania reči	2
1.1.1 Typy rozpoznávania	3
1.2 História	4
1.2.1 Začiatky	4
1.2.2 Dynamic time warping	5
1.2.3 Skryté Markovovské modely	5
1.2.4 Neurónové siete	6
1.3 Podobné práce a systémy	7
2 HTK	8
2.1 HTK toolkit	8
2.2 Podobné nástroje	9
2.3 Inštalácia HTK	10
3 Skryté markovovské modely	11
3.1 Zmesi gausiánov	11

Úvod

Úvod.

1. Úvod do problematiky

V tejto kapitole sa budeme najskôr venovať histórii rozpoznávania reči. Pozrieme sa na rôzne prístupy k nej a na súčasný stav vývoja. Nakoniec si predstavíme rôzne podobné systémy a práce, ktoré sa venovali tejto téme.

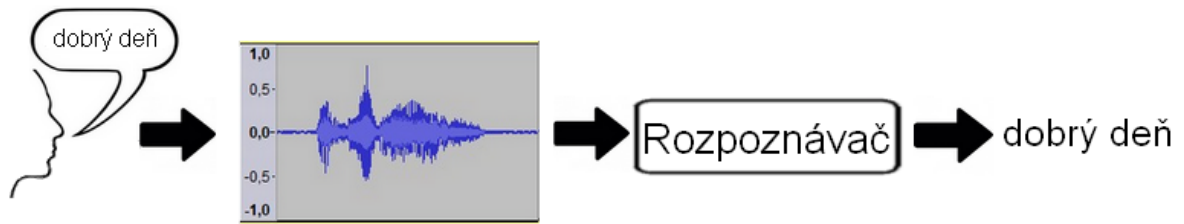
1.1 Úvod do rozpoznávania reči

Problém rozpoznávania reči - ASR(z angl. Automatic speech recognition) vznikol spoločne s vývojom modernej výpočtovej techniky. Hoci ešte stále nie je uspokojivo vyriešený, dnešné systémy sú dostatočne úspešné pre ich použitie v niektorých oblastiach.

Proces rozpoznania reči je náročný a oblastí, ktorými sa pri ňom musíme zaoberať je niekoľko a siahajú do viacerých oborov.

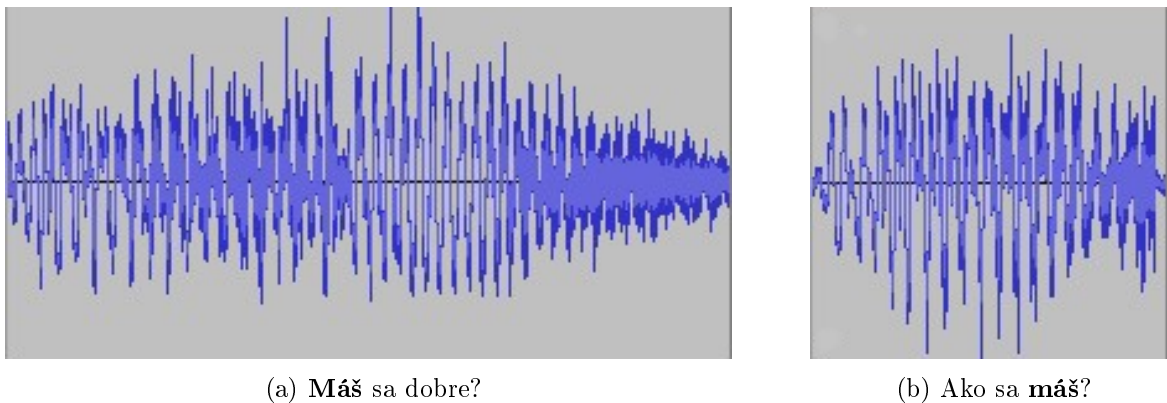
- Akustika - ako vzniká slovo v ľudskom hlasovom trakte, jeho šírenie rôznymi prostrediami a nakoniec spracovanie ušným ústrojenstvom
- Spracovanie signálu - problematika digitalizovania signálu, odstraňovania šumu, vyextrahovania vhodných informácií potrebných pre rozpoznanie reči
- Lingvistika - stavba jazyka, rozdelenie slov na slabiky a hlásky, vzťah medzi zvukom(fonetika) a významom(sémantika)
- Rozpoznávanie vzorov - počítačové algoritmy a metódy na klasifikáciu dát, hľadania podobností

Počítač musí byť schopný zachytiť zvukovú vlnu, šíriacu sa vzduchom, spracovať ju, rozdeliť na slová, prípadne hlásky a poskytnúť používateľovi prepis vyslovenej reči (1.1).



Obr. 1.1: Rozpoznanie slovného spojenia "dobry deň" ASR systémom

Zložitých problémov ktoré nám pri rozpoznávaní reči vyvstanú je niekoľko. Každý človek má iný hlas. Ženy väčšinou vyšší, muži nižší. Dokonca aj jednej osobe sa mení hlas v priebehu dňa. Iný má keď ráno vstane a iný napríklad počas choroby. Naša reč sa líši aj v tempe. Môžeme slovo vysloviť rýchlejšie, pomalšie. Znenie slova sa mení aj v závislosti od emócií, od použitého kontextu. Na obrázku (1.2) vidíme porovnanie zvukovej vlny slova "máš" v dvoch rôznych vetách, vyslovených tým istým rečníkom hneď za sebou. Tvar vlny je síce podobný, ale dĺžka slova vo vete "Máš sa dobre?" je dvojnásobne dlhšia oproti vete "Ako sa máš?". Ďalší faktor vstupujúci do procesu rozpoznávania je vplyv pozadia. Absolútne tiché prostredie je nepoužiteľné, lebo sa nedá dosiahnuť v reálnom nasadení systémov. Vždy musíme počítať so šumom pochádzajúcim z nedokonalosti mikrofónu alebo z prostredia. Rozhovor v pozadí, otvorenie dverí alebo hučanie ventilátora zmenia charakter zvukovej vlny. Modely ASR systémov musia byť robustné a snažiť sa eliminovať tieto nežiaduce javy.

Obr. 1.2: Porovnanie zvukových vln slova **máš** v 2 rôznych vetách.

1.1.1 Typy rozpoznávania

ASR systémy môžeme rozdeliť podľa niekoľkých základných kritérií.

Môžeme rozpoznávať samotné hlásky, izolované slová alebo plynulú reč. Rozpoznanie hlások je náročnejšie, kôli absencii kontextu, ktorý nám môže pomôcť určiť

pravdepodobnosť priradenia hlásky k danej zvukovej sekvencii. Podobný princíp platí aj pri rozoznávaní izolovaných slov a plynulej reči, zloženej z viet. Pri definovanej gramatike jazyka vieme vypočítať pravdepodobnosť výskytu daného slova vo vete. Pri plynulej reči je však problém s rozličnými variantami toho istého slova podľa použitia, ako sme si ukázali na obrázku (1.2).

Hlavne v minulosti boli ASR systémy závislé na rečníkovi. To znamená, že boli schopné rozpoznať reč iba jedného, resp. niekoľkých ľudí. Bolo bežné, že po kúpe komerčného softvéru na transkripciu reči, musel používateľ prečítať a nahráť pripravený text a tak dotrénovať softvér na jeho konkrétny hlas. Dnes je snaha vyvíjať systémy nezávislé na rečníkovi. Je to možné hlavne vďaka dostupnému množstvu dát a zlepšeniu hardvéru.

Zaujímá nás aj veľkosť slovnej zásoby. Koľko slov je rozpoznávač schopný identifikovať. Samozrejme je jednoduchšie rozlišovať medzi niekoľkými povelmi, ako poznať jadro slovnej zásoby jazyka. Väčšina moderných systémov s veľkou slovnou zásobou je preto založená na rozpoznaní foném, resp. viacerých spojených foném - trifónov, bifónov, difónov. Z nich sa potom vyskladajú slová. Rozpoznávač by mal byť teda schopný rozpoznať aj neznáme slovo, na ktoré nebol špeciálne tréňovaný.

Ďalšie kritérium ASR systému je či bude transkripcia prebiehať v reálnom čase. Ak chceme ovládať hlasom nejaké zariadenie, reakcia systému musí byť okamžitá aj za cenu miernych nepresností. Naopak pri generovaní prepisu videa je výhodnejšie nechať systém dlhšie pracovať a získať tak kvalitnejší text.

1.2 História

Použitie rozpoznávania reči je v poslednom čase veľmi skloňované, hlavne kôli produktom veľkých spoločností napr. Alexa, Siri alebo Google Assistant, ktoré sú spoľahlivo ovládané hlasom. O túto problematiku sa však zaujímali informatici už desaťročia dozadu a jej vývoj prešiel rôznymi objavmi a zmenami, ktoré vyústili do dnešného stavu.

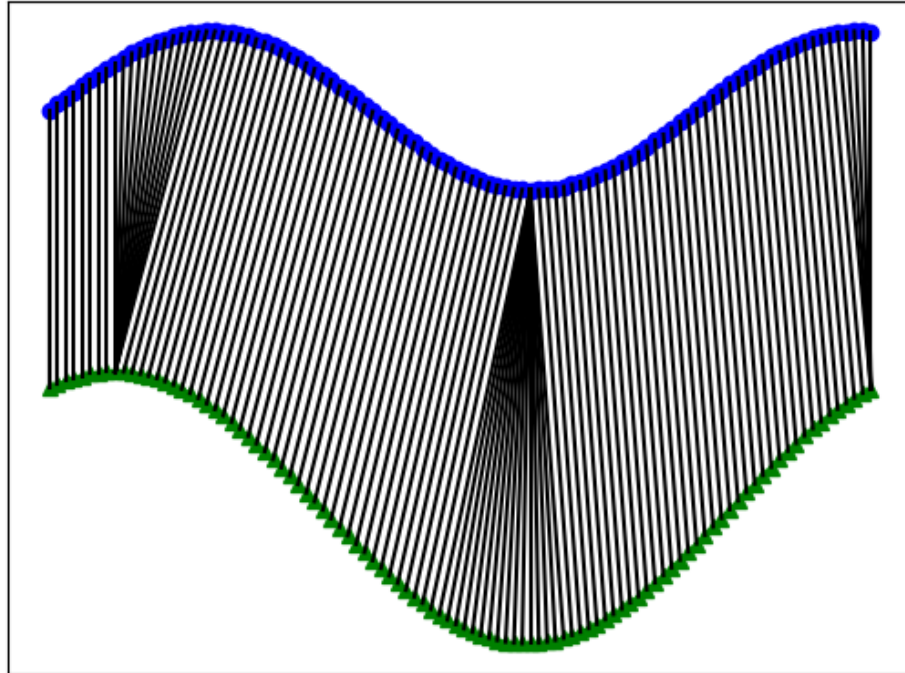
1.2.1 Začiatky

Za prvý pokus o ASR systém môžeme považovať rozpoznávač „Audrey“ z roku 1952 zostrojený v Bell Laboratories. Vedel rozpoznať číslice do desať vyslovené jedným rečníkom. Založený bol na rozpoznaní formantov, špičiek vo zvukovom spektre, počas samohlások každej číslice [3]. Ďalšie výskumy sa zameriavali na rozpoznanie niekoľkých samohlások a spoluhlások, čo sa ukázalo ako dobrý postup, keďže z hlások vieme vyskladať neskôr slová. Slovná zásoba však bola stále obmedzená na desať, dvadsať izolovaných slov.

1.2.2 Dynamic time warping

Posun nastal vymyslením algoritmu Dynamic time warping - DTW, ktorý slúži na hľadanie podobnosti medzi dvoma lineárnymi vstupmi líšiacimi sa v rýchlosti za pomoci dynamického programovania.

Rovnaké slovo vyslovené dvoma rôznymi rečníkmi sa väčšinou líši v tempe. Niektorí rozpráva rýchlejšie a niektorí pomalšie. DTW ohýba časovú os, aby zrovnal vstupné signály.



Obr. 1.3: Ukážka zarovnávania dvoch signálov pomocou ohýbania časovej osi

Neznáme slovo sa samostatne porovnáva so všetkými slovami v danom slovníku. Výberie sa to, s ktorým má najväčšiu zhodu. Prístup je teda klasifikačný, z čoho plynie problém, že hoci sa slovo nemusí nachádzať vôbec v rozpoznávanom slovníku, ani sa na žiadne podobáť, algoritmus ho vždy priradí k najpodobnejšiemu. Tento postup sa osvedčil najmä pri rozpoznávaní menšieho počtu izolovaných slov. Veľkosť slovej zásoby, ktorú ASR systém dokázal rozpoznať ale stúpala na niekoľko desiatok slov.

1.2.3 Skryté Markovovské modely

Ďalší veľký skok v ASR systémoch priniesli skryté markovovské modely - HMM (z angl. Hidden Markov Models). Tomuto riešeniu sa budeme podrobne venovať v ďalších kapitolách. Začali sa používať v osemdesiatych rokoch a tento nový štatistický prístup bol považovaný za najlepší až do nedávnej minulosti. S rozličnými vylepšeniami sa HMM ešte stále používa v niektorých komerčných softvéroch.

1.2.4 Neurónové siete

Používanie umelých neurónových sietí - ANN (z angl. artificial neural networks) v oblasti umelej inteligencie je dnes veľmi moderné. Tento model však bol známy už v päťdesiatych rokoch. Úspešné pokusy o jeho aplikáciu v oblasti rozpoznávania reči sa dosiahli v osemdesiatych a deväťdesiatych rokoch minulého storočia. ANN neboli použité samostatne, ale vylepšovali konkrétne problémy v štatistických HMM systémoch, napríklad odhad rozdelenia pravdepodobnosti alebo vektorizácia vstupného signálu. Predstavený bol HMM/MLP model (MLP - multi layer perceptron) [4], ktorý načrtol budúcnosť použitia ANN v ASR, ale vtedajší hardvér a algoritmy na učenie, nepostačovali aby dokázali konkurovať systémom založených na skrytých Markovovských modeloch.

Odvtedy vývoj pokročil a pomerne dobrú úspešnosť dosahovali aj systémy založené hlavne na neurónových sieťach. Samotný klasifikátor je realizovaný ANN, ale ešte stále je použité predspracovanie signálu alebo vektorizácia signálu algoritmami, aké sa používajú napríklad pri HMM systémoch a budeme si o nich hovoriť v neskorších kapitolách. Nejde teda o rozpoznávanie "od začiatku do konca" (angl. End-to-End) neurónovými sieťami.

Rozpoznávaniu čísel slovenského jazyka za pomoci neurónových sietí sa venuje práca Vojtecha Slovika [5]. Použil trojvrstvovú klasifikačnú ANN s jednou vrstvou skrytých neurónov. Použil aj techniku "okna do minulosti a budúcnosti", ktoré eliminuje problém s časovým kontextom reči, teda že výslovnosť hlásky závisí aj od hlásky pred ňou a po nej. Podarilo sa mu dosiahnuť dobrú úspešnosť okolo 90% pre 16-slovný slovník a 8 rôznych rečníkov.

Skutočnú revolúcia do problému ASR však priniesla metóda hlbokého učenia (angl. deep learning) v posledných pár rokoch. K natrénovaniu takejto DNN siete (z angl. deep neural network) treba veľké množstvo výpočtovej sily a veľké množstvo dát. Dnes to už nie je problém, vďaka vývoju špecializovaných grafických kariet a technologickým gigantom ako napríklad Google, ktorý vie pohodlne zozbierať ohromné množstvá hovorenej reči od svojich používateľov.

DNN "end-to-end" rozpoznávače sú schopné väčšej abstrakcie nad vstupným signálom, teda lepšie využijú kontext časti signálu ako predchádzajúce rozpoznávače, ktoré ho analyzujú po maličkých častiach. Navyše, ako potvrdil tento experiment [6] pri tréningu nie je potrebný fonetický prepis tréningových dát, čo predstavuje veľké ušetrenie práce pri príprave datasetov. DNN systémy úspešnosťou rozpoznávania slov aj plynulej reči, už prevyšujú klasické HMM modely a práve týmto smerom sa ubera aktuálny vývoj ASR.

1.3 Podobné práce a systémy

Automatické rozpoznávanie reči je zaujímavý problém pre komerčnú sféru, kôli možnosti ovládať zariadenia hlasom. Systémy ako Google Assistant, Siri alebo Alexa dosahujú vysokú úspešnosť pre plynulú reč a veľkú slovnú zásobu. Pravdepodobne fungujú na technikách DNN. Je ale potrebný internetový prístup. To ich robí v niektorých situáciách nepoužiteľnými. Systémy bez potreby internetu majú vyššiu nepresnosť alebo obmedzenú slovnú zásobu.

Oblasti rozpoznávania reči sa venovala práca [7]. Cieľ bol rozpoznať slovenské hlásky s využitím rôznych techník, ako napríklad modelovanie šumu pozadia. Rozpoznávanie samostatných hlások je ťažší problém ako rozpoznávanie slov, kôli absencii kontextu. Dosiadnutá úspešnosť sa pohybovala len okolo 50%, ale podarilo sa dosiahnuť mierne zlepšenie po použití optimalizačnej metódy klasteringu. Túto techniku by sme mohli využiť aj v našej práci.

Rozpoznávaním reči založenom na zmesiach gausiánov sa venovala práca [8]. Oproti štandardným gausiánom používa hlbokú architektúru. To znamená, že použijeme viac vrstiev gausiánskych zmesí nad sebou. Ide o kombináciu techník hlbokého učenia a štandardného HMM-GMM prístupu (GMM - z angl. Gaussian Mixture Models). Na vyhodnotenie úspešnosti tohto prístupu, bol systém natrénovaný na rozpoznávanie slovenských číslic. Dosiadol mierne zlepšenie o necelé percento oproti modelu bez hlbokkej architektúry. Nevýhodou tohoto modelu sú zvýšené časové nároky na trénovanie, ktoré môže trvať aj desaťkrát dlhšie. Na základe poznatkov o hlbokých architektúrach môžeme predpokladať, že navrhnutý model bude úspešnejší pri dostatočne veľkej trénovacej množine, avšak pri nedostatku trénovacích dát bude mať väčšiu chybovosť ako štandardný model.

2. HTK

V tejto kapitole si povieme niečo o nástroji na prácu so skrytými markovovskými modelmi HTK [2] (z angl. Hidden markov model ToolKit). Uvedieme si aj iné systémy umožňujúce prácu s HMM.

2.1 HTK toolkit

HTK je nástroj vyvinutý na Cambridge University. Primárne je určený na výskum okolo rozpoznávania reči, ale použitie môže nájsť aj v iných oblastiach používajúcich HMM, napríklad rozpoznávanie znakov alebo DNA sekvenovanie. Poskytuje viacero nástrojov na spracovanie signálu, inicializáciu HMM modelov, ich tréning a analýzu výsledkov. Takisto je k HTK vydaná podrobná dokumentácia aj s príkladmi použitia [2]. Uvedieme si základné a najpoužívanejšie nástroje v rôznych fázach vývoja ASR systému pomocou HMM.

Príprava dát: na začiatku treba pripraviť tréningovú a testovaciu množinu nahrávok spolu s ich transkripciou - HSLAB. Na manipuláciu s nahrávkami ako je ich spájanie, delenie a ich parametrizáciu slúži HCopy. Na konvertovanie a upravovanie súborov s transkripciou reči slúži HLed.

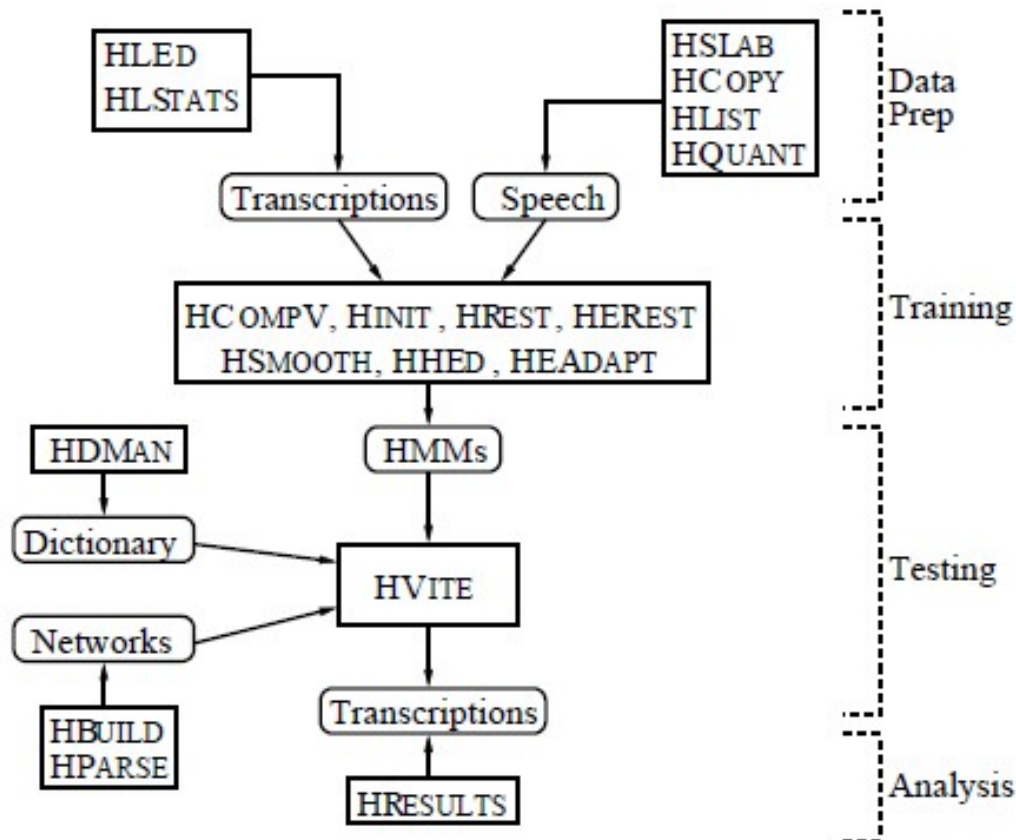
Tréningovanie: Po vytvorení vhodnej HMM architektúry a vytvorení prototypov modelov (ručne editovateľné textové súbory) ich môžeme inicializovať. Na inicializovanie a prvotný odhad parametrov pomocou Viterbiho algoritmu má systém HTK nástroj HInit. Potom môžeme použiť HERest, ktorý používa Baum-Welch algoritmus na úpravu odhadov parametrov modelu. HHed slúži na úpravu jednotlivých HMM, napríklad pridanie gausiánov do jednotlivých stavov. Potom treba celý model pretrénovať znova nástrojom HERest.

Rozpoznávanie: HVite realizuje samotné rozpoznávanie. Vstupom je súbor definujúci gramatiku jazyka (aké sekvencie slov sú dovolené), sieť HMM, slovník výslovnosti slov a neznámy audio súbor. Za pomoci Viterbiho algoritmu sa zisťuje prepis slova, ktorý uloží do súboru. Alternatívou k HVite je HDecode, ktorý sa hodí pre väčšie slovníky.

Analýza: Posledný krok je zistenie úspešnosti rozpoznávania pomocou HResults.

Tento nástroj porovná výstup z HVite a správny prepis audia za pomoci dynamického programovania. Spočíta chyby nahradenia, odstránenia a vloženia.

Na obrázku 2.1 môžeme vidieť spomínané fázy a nástroje poskytované HTK toolkitom.



Obr. 2.1: Rôzne nástroje poskytované HTK toolkitom

2.2 Podobné nástroje

Hoci je už HTK pomerne starý nástroj, neustále zlepšenia (posledná nová verzia je z roku 2015), podrobná dokumentácia a veľa voľne dostupných materiálov robia z neho konkurencie schopný softvér.

O niečo novší je systém Kaldi [11]. Oproti HTK má výhodu voľnejšej licencie a dá sa viac upraviť podľa potrieb. Na druhej strane mu chýba podrobná a zrozumiteľná dokumentácia akou disponuje HTK. Podrobnejšej analýze týchto dvoch nástrojov sa venovala táto práca [8]. Na vykonanom teste dosiahli porovnateľné výsledky, Kaldi o niečo lepšie.

Posledným nástrojom, ktorý spomenieme je CMUSphinx [12]. Jeho výhodou je pod-

pora jazykov Java a Python, čo ho robí použiteľným v mnohých prostriediach. Taktiež ponúka odľahčenú verziu Pocketsphinx, ktorá beží aj na operačnom systéme Android, teda je použiteľná v mobilných aplikáciách. Je však viacej kompaktný a neponúka také podrobné možnosti stavby modelov a ich tréovania ako HTK alebo Kaldi.

2.3 Inštalácia HTK

Návod na inštaláciu HTK na Linuxe aj Windowse je dostupný online [2]. Inštalácia by mala byť pomerne priamočiara, autor práce však mal s ňou problémy na linuxovej distribúcii Debian, preto sa rozhodol uviesť pár bodov, ktoré odstránia prípadé problémy.

Nástroje HTK sú 32-bitová aplikácie, čo spôsobuje problémy s grafickými knižnicami na 64-bitových systémoch. Jediný spôsob, ktorý sa osvedčil bolo vynechať z kompilácie nástroj HSlab, ktorý jediný používa tieto knižnice. Skript **configure** bolo treba spustiť nasledovne:

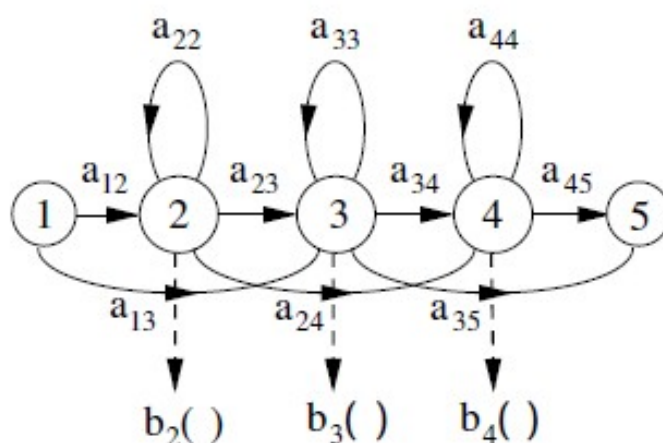
```
$ configure --without-x --disable-hslab
```

HSlab slúži na nahrávanie reči a tvorbu súborov s transkripciou reči na tréovanie. Alternatíva k nemu je napríklad program Wavesurfer [14].

Druhou chybou, ktorá sa vyskytla po spustení príkazu **make install** bolo nesprávne odsadenie v skripte MakeFile. V adresári **htk/HLMTools** ho treba upraviť. Na riadku 77 vymažeme odsadenie tvorené medzerami a nahradíme ich tabelátorom.

3. Skryté markovské modely

Skrytý Markovov model - HMM je štatistický model generujúci postupnosť symbolov. Pomocou HMM sa dajú modelovať postupnosti symbolov, ak predpokladáme, že modelovaný jav sa v rôznych okamihoch nachádza v rôznych stavoch a generovanie symbolov nezáleží na stavoch, v ktorých bol jav predtým a potom. HMM je vlastne konečný automat, ktorého prechody medzi stavmi sú ováňované pravdepodobnosťami a každý vygenerovaný symbol sa generuje pod určitou pravdepodobnosťou. HMM generuje symbol vždy v jednom stave, potom sa presúva do druhého stavu. Príklad jednoduchého modelu môžeme vidieť na obrázku 3.1.



Obr. 3.1: HMM model

3.1 Zmesi gausiánov

DOROBÍŤ.

Literatúra

- [1] Jurafsky D., Martin J., Speech and Language Processing, Third Edition draft, 2018
- [2] HTK, Hidden Markov Model Toolkit, <http://htk.eng.cam.ac.uk/> [navštívené 8.1.2019]
- [3] Juang B. H., Rabiner L. R., Automatic speech recognition—a brief history of the technology development, Georgia Institute of Technology, Atlanta, Rutgers University and the University of California, Santa Barbara
- [4] Bourlard H., Morgan N., Connectionist Speech Recognition: A Hybrid Approach, The Kluwer International Series in Engineering and Computer Science, Boston, MA: Kluwer, 1994
- [5] Rozpoznávanie foném čísel slovenského jazyka neurónovou sieťou, Slovik V., 2007, Diplomová práca, FMFI UK
- [6] Graves A., Jaitly N., Towards End-to-End Speech Recognition with Recurrent Neural Networks, International Conference on Machine Learning (ICML), 2014
- [7] Ritomský O., Zvýšenie úspešnosti rozpoznávania izolovaných hlások využitím techniky modelovania šumu pozadia, 2015, Bakalárska práca, FMFI UK
- [8] Šuppa M., Speech recognition based on deep gaussian mixture models, 2016, Bakalárska práca, FMFI UK
- [9] Young S., Evermann G., Gales M., Hain T., Kershaw D., Liu X., Moore G., Odell J., Ollason D., Povey D., Valtchev V., Woodland P., The HTK Book (for HTK Version 3.4), 2006, Cambridge University Press
- [10] Young S. J., Young S., The HTK hidden Markov model toolkit: Design and philosophy, 1993, University of Cambridge, Department of Engineering
- [11] Kaldi toolkit, <http://kaldi-asr.org/> [navštívené 19.1.2019]
- [12] CMUSphinx speech recognition system, <https://cmusphinx.github.io/> [navštívené 19.1.2019]

- [13] Wavesurfer tool for sound, <http://www.speech.kth.se/wavesurfer> [navštívené 19.1.2019]