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 rozpoznávannia rečí, 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: Jurasfky 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

Poďakovanie: Ďakujem.

Abstrakt

Abstrakt.

Kľúčové slová: rozpoznávanie reči, HTK, skryté markovovské modely

Abstract

Abstract.

Keywords: speech recognition, HTK, hidden markov models

Obsah

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 súčastný stav vývoja. Nakoniec si predstavíme rôzne podobné systémy a práce, ktoré sa venovali tejto téme.

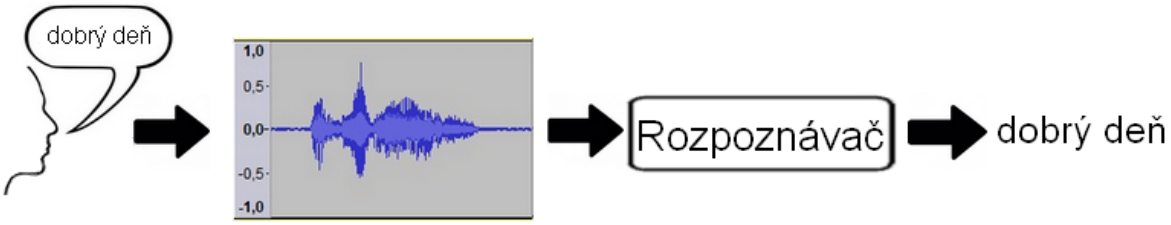
1.1 Úvod do rozpoznávania reči

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.

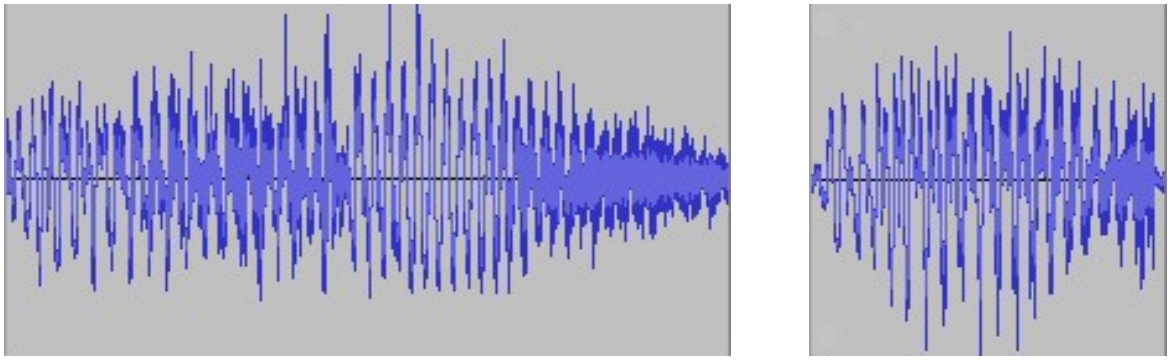
- Akustika - ako vzniká slovo v ľudskom hlasovom trakte, jeho šírenie rôznymi prostrediami a nakoniec spracovanie ušným ústrojenstvom
- Spracovanie signálu - problematika digitalizovania signálu, odstraňovania šumu, vyextrahovania vhodných informácií potrebných pre rozpozanie reči
- Lingvistika - stavba 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 "dobrý deň ÄSR 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 líší aj v tempe. Môžme slovo vysloviť rýchlejšie, pomalšie. Znenie slova sa mení aj v závislosti od emócii, 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 Äko sa máš?". Ďalší faktor vstupujúci do 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.



(a) **Máš** sa dobre? (b) Ako sa **máš**?

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

1.1.1 Typy rozpoznávania

ASR systémy môžme rozdeliť podľa niekoľkých základných kritérií.

Môžme rozpoznávať samotné hlásky, izolované slová alebo plynulú reč. Rozpoznávanie 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 nahrať 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.

Zaujíma 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 trénovaný.

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 Assistent, 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.

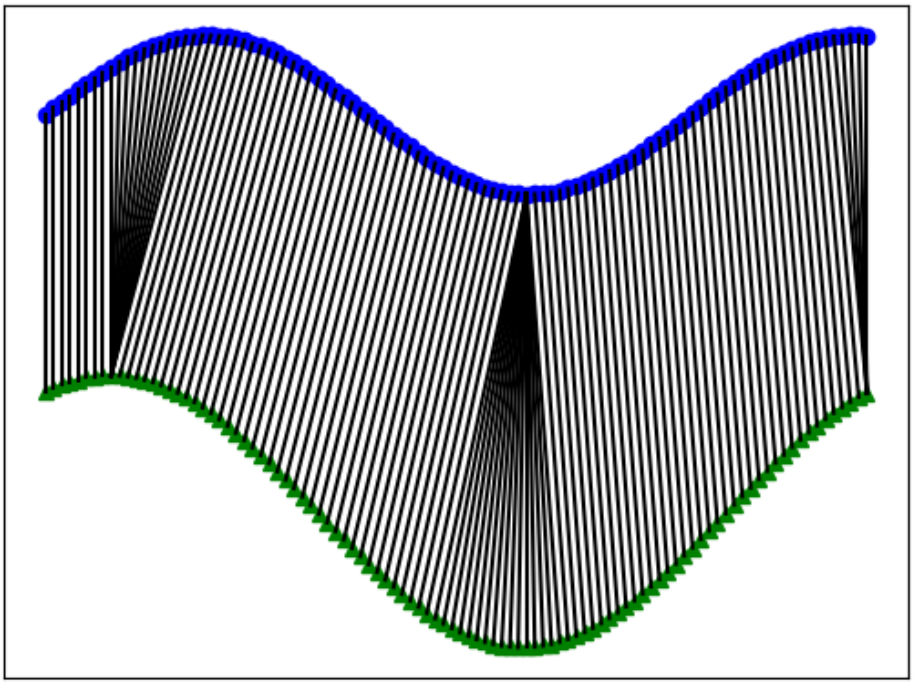
1.2.1 Začiatky

Za prvý pokus o ASR systém môžme považovať rozpoznávač Äudrey"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 ukázal 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. Niekto rozpráva rýchlejšie a niekto pomalšie. DTW öhýba"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. Vyberie 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 podobať, algoritmus ho vždy priradí k najpodobnejšiemu. Tento postup sa osvedčil najme pri rozpoznávaní menšieho počtu izolovaných slov. Veľkosť slovnej zásoby, ktorú ASR systém dokázal rozpoznať ale stúpla 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 softvérch.

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 pravdepodonosti 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 öd 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 ökna 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 revolúciu 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 vývojú špecializovaných grafických kariet a technologickým gigantom ako napríklad Google, ktorý vie pohodlné zozbierať ohromné množstvá hovorenej reči od svojich používateľov.

DNN ënd-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 trénovaní 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 uberá 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 Assistent, 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. Dosiahnutá 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. Dosiahol mierne zlepšenie o necelé percento oproti modelu bez hlbokej architektúry. Nevýhodou tohoto modelu sú zvýšené časové nároky na trénovanie, ktoré môže trvať aj desaťkrát dlhšie. Na základe poznatkov o hlbokých architektúrach môžme 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 trénovaciu a testovaciu množinu nahrávok spolu s ich transkripciou - HSLAB. Na manipuláciu s nahrávkami ako je ich spájenie, delenie a ich parametrizáciu slúži HCopy. Na konvertovanie a upravovanie súborov s transkripciou reči slúži HLed.

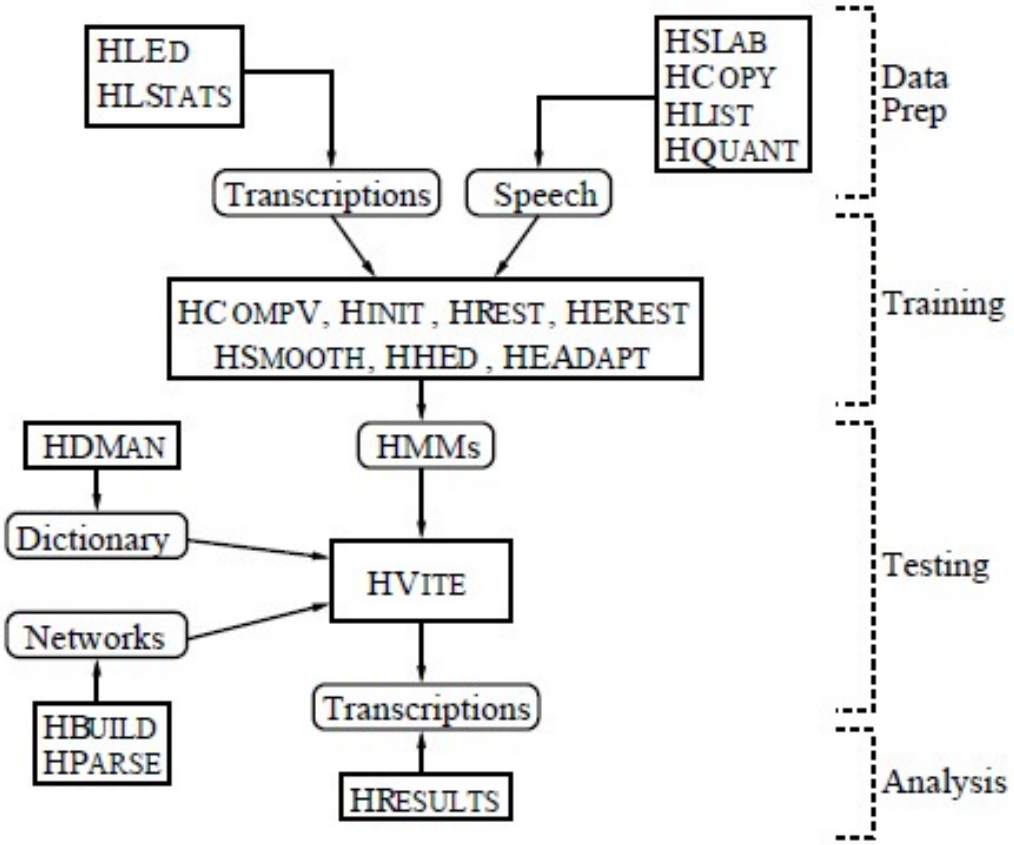
Trénovanie: Po vytvorení vhodnej HMM architektúry a vytvorení prototypov modelov(ručne editovateľné textové súbory) ich môžme inicializovať. Na inicializovanie a prvotný odhad parametrov pomocou Viterbiho algoritmu má systém HTK nástroj HInit. Potom môžme použiť HERest, ktorý používa Baum-Welch algoritmus 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 zistiťuje 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 môžme 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 dokuntácia 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 trénovania 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
```

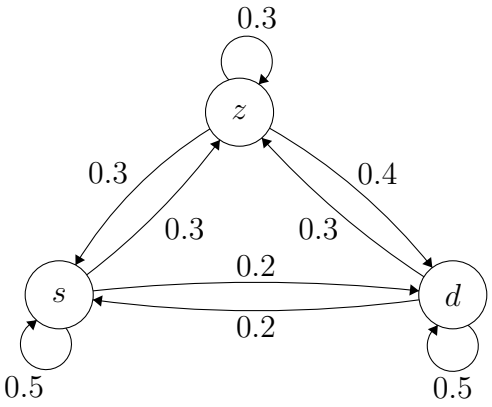
HSlab slúži na nahrávanie reči a tvorbu súborov s transkripciou reči na trénovanie. Alternatíva k nemu je napríklad program Wavesurfer [13] alebo Audacity .

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 markovovské modely

Skryté markovovské modely(HMM - z angl. hidden markov models) sú rozšírením markovovských reťazcov. Markovovské reťazce sa snažia modelovať náhodné procesy ako postupnosti stavov a pravdepodobnosti prechodov medzi týmito stavmi. Založené sú na predpoklade, že budúcnosť závisí len na aktuálnom stave, nie na minulosti.
Klasický príklad práce markovovských reťazcov je predpoveď počasia. Predpokladajme, že počasie je vždy definované jedným z nasledujúcich stavov: slnečno(s), zamračené(z), dážď(d). Ďalej predpokladajme, že pravdepodobnosti prechodov medzi jednotlivými stavmi sú nasledovné:

Takto definovaný model si môžme predstaviť ako konečný automat zobrazený na obrázku 3.1.



Obr. 3.1: markovovský reťazec M na predpoveď počasia

Skúsme vypočítať pravdepodobnosť, že najbližších 5 dní bude počasie nasledovné: s,s,z,d,z, ak dnes prší. Chceme vypočítať pravdepodobnosť sekvencie stavov $X = \{d, s, s, z, d, z\}$ z modelu M. Budeme vychádzať z vyššie spomenutého predpokladu,

že nasledujúci stav, závisí iba od aktuálneho.

$$\begin{aligned} P(X|M) &= P(d,s,s,z,d,z|M) \\ &= P(d) \cdot P(s|d) \cdot P(s|s) \cdot P(z|s) \cdot P(d|z) \cdot P(z|d) \\ &= \pi_d \cdot a_{ds} \cdot a_{ss} \cdot a_{sz} \cdot a_{zd} \cdot a_{dz} \\ &= 1 \cdot 0.2 \cdot 0.5 \cdot 0.3 \cdot 0.4 \cdot 0.3 \\ &= 0.0036 \end{aligned} \tag{3.1}$$

π_d - pravdepodobnosť, že dážď bude začiatočný stav. V našom prípade je to 1, lebo vieme, že dnes prší.

V tomto jednoduchom príklade vieme určiť aké je počasie jednoznačne, tak, že sa pozrieme z okna. Niektoré procesy ale nevieme pozorovať priamo (sú nám skryté) a informácie o nich získavame skrze iné procesy, ktoré produkujú nejaké pozorovania. Napríklad si predstavme, že sme v miestnosti bez okien a o tom aké je počasie nás informuje osoba, ktorá niekedy klame. Skutočná postupnosť stavov počasia nám je skrytá a poznáme len pozorovania od danej osoby, ktoré sú s určitou pravdepodobnosťou pravdivé. Teda jednotlivé stavy nebudú reprezentovať skutočný stav počasia, ale generujú pravdepodobnosti s akými bolo v príslušnom stave odpozorované dané pozorovania. Takto sme rozšírili markovovské reťazce na skryté markovovské modely.

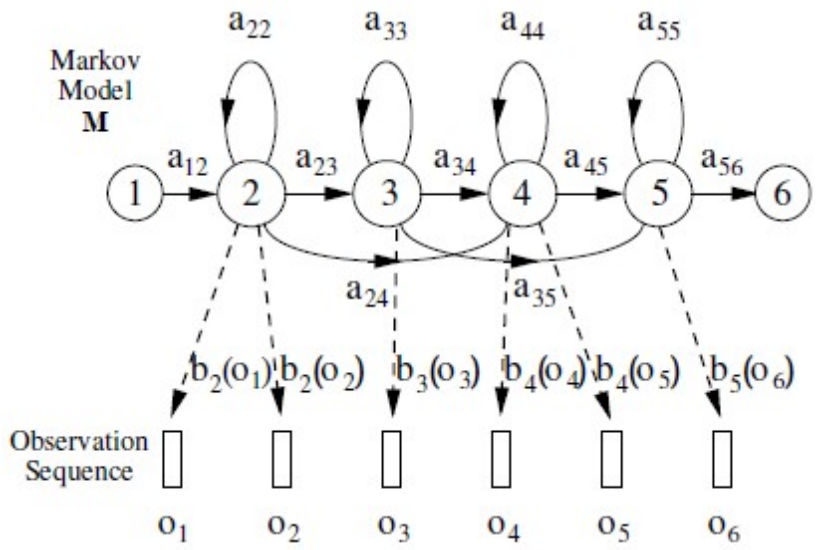
Podľa [1] môžme zadefinovať HMM ako nasledovnú päticu:

$$M = (Q, A, O, B, \pi) \tag{3.2}$$

$Q = q_1, q_2 ... q_n$ - množina všetkých stavov
$A = a_{11} ... a_{ij} ... a_{nn}$ - matica pravdepodobností prechodov, a_{ij} - pravdepodobnosť prechodu zo stavu i do stavu j, z toho vyplýva $\sum_{j=1}^{n} a_{ij} = 1, \forall i$
$O = o_1, o_2 ... o_t$ - sekvencia t pozorovaní zo slovníka $V = v_1, v_2 ... v_v$
$B = b_i(o_j)$ - pravdepodobnosť generovania pozorovania o_j v stave i
$\pi = \pi_1, \pi_2 ... \pi n$ - počiatočné rozdelenie pravdepodonosti, π_i - pravdepodobnosť, že počiatočný stav bude i, musí platiť, že $\sum_{i=1}^{n} \pi_i = 1$

Z praktických dôvodov sa v nástroji HTK [2] pridáva ešte jeden špeciálny stav na začiatok a na koniec modelu. Tieto stavy negenerujú pozorovania a prechody medzi nimi sa dejú v nulovom čase. Výhoda je, že umožňujú spájať samostatné HMM do väčších celkov. Z toho vyplýva, že π_i stráca zmysel, lebo sa vždy začne v špeciálnom stave a tieto pravdepodobnosti sa presunú do prechodov medzi p;vodnými stavmi a špeciálnym začiatočným stavom.
Na obrázku 3.2 môžme vidieť takto upravený model ako prechádza cez sekvenciu stavov $X = 1, 2, 2, 3, 4, 4, 5, 6$ a počíta pravdepodobnosť vygenerovania postupnosti pozorovaní $O = o_1, o_2 ... o_6$.

$$P(O, X|M) = a_{12} b_2 o_1 a_{22} b_2 o_2 a_{23} b_3 o_3 ... \tag{3.3}$$



Obr. 3.2: príklad HMM modelu [9]

Postupnosť stavov nám pri HMM ale nie je známa, poznáme len pozorovanie O a pravdepodobnosť vygenerovania tohto pozorovania modelom M vypočítame ako sumu pravdepodoností prechodov cez všetky možné sekvencie stavov $X = x(1), x(2), x(3)...x(T)$. [9]

$$P(O|M) = \sum_{X} a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)} \tag{3.4}$$

$x(0)$ je špeciálny vstupný stav a $x(T+1)$ bude špeciálny koncový stav.
Jednoduchšia možnosť, ktorá sa v praxi používa, je aproximovanie výsledku a to tak, že nájdeme len najpravdepodobnejšiu sekvenciu stavov, teda takú, ktorej pravdepodobnosť vygenerovania O je maximálna.

$$\hat{P}(O|M) = \max_{X}\{a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)}\} \tag{3.5}$$

3.1 HMM a rozpoznávanie reči

Vstupom pre rozpoznávanie reči je síce meniaci sa rečový audio signál, ale keď ho rozdelíme na dostatočne malé úseky(10-30ms) môžme ho považovať za stacionárny vzhľadom k tomu, že náš hlasový aparát nemení konfiguráciu v takomto krátkom časovom úseku. Teda môžme považovať reč za postupnosť stavov. Rozpoznanie slova

Pravdepodobnosť $b_i(o_j)$ môže byť diskrétna alebo spojitá.

3.2 Zmesi gausiánov

DOROBIŤ.

4. Spracovanie signálu

Spracovanie audio signálu je prvým článkom v každom ASR systéme, či už je založený na HMM, ANN, DTW alebo inom princípe. Jeho úloha je prekonvertovať zvukovú vlnu do nejakej parametrickej reprezentácie pre ďalšie spracovanie a analýzu.
Vstupom pre rozpoznávanie reči je síce meniaci sa rečový audio signál, ale keď ho rozdelíme na dostatočne malé úseky(10-30ms) môžme ho považovať za stacionárny vzhľadom k tomu, že náš hlasový aparát nemení svoju konfiguráciu v takomto krátkom časovom úseku. Teda môžme považovať rečový signál za postupnosť stavov, ktorých je konečný počet. Parametre získané z týchto úsekov nazývame príznaky.
Spôsobov akým sa signál parametrizuje, aby sme z neho získali čo najviac informácii o reči, je viacero, ale najpoužívanejšie - LPC a MFCC sa zakladajú na spektrálnej analýze signálu. Najskôr si ale povieme niečo o Hammingovom okienku.

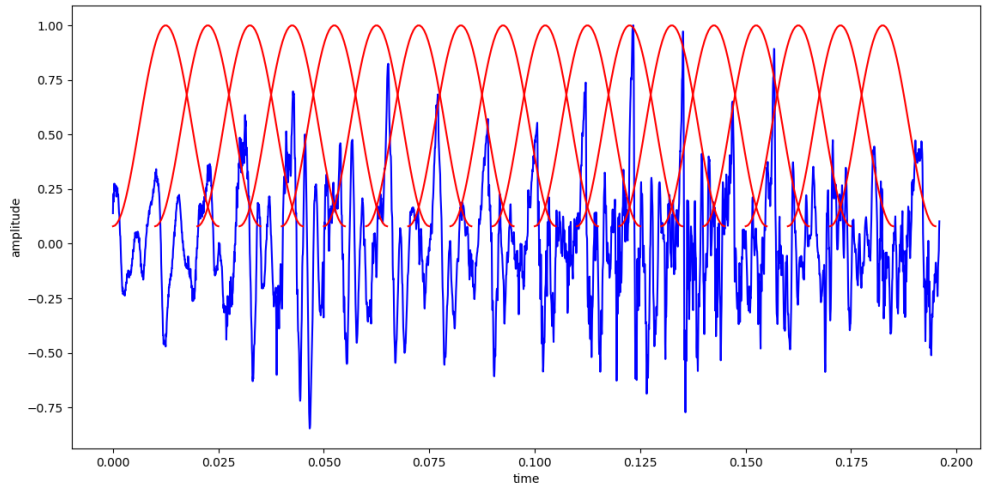
4.1 Hammingovo okienko

Ako sme si už povedali, rečový signál analyzujeme po krátkych úsekoch, rádovo v desiatkach milisekúnd. Ak by sme len jednoducho rozdelili zvukovú vlnu na pravidelné časti požadovanej dĺžky, stratili by sme kontext a mohli by sme prísť o časť informácií. Preto sa používajú prekrývajúce sa ökienkové"funkcie [1].
V oblasti spracovania signálu pre ASR systém je najpoužívanejšie Hammingovo okienko. Je to funkcia definovná predpisom 4.1.

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{M-1}) \quad 0 \leq n \leq M-1 \tag{4.1}$$

M - veľkosť okienka

Okienko "kĺza"po signále o hodnotu menšiu ako je jeho dĺžka, čiže sa bude prekrývať a ováhuje nám jednotlivé hodnoty amplitúdy vo výseku. Ukážku práce takéhoto okienka môžme vidieť na obrázku 4.1.



Obr. 4.1: Ukážka kĺzajúceho sa Hammingovho okienka

4.2 LPC

LPC (z angl. Linear Predictive Coding) je metóda založená na predpoklade, že rečová vzorka môže byť odhadnutá ako lineárna kombinácia predchádzajúcich vzoriek. LPC modeluje rečový signál ako zdroj, ktorý je generovaný hlasivkami a filter, tvorený krkom a ústami(4.2). Snaží sa oddeliť tieto rezonancie od seba a lokalizovať už spomínané formanty - špičky vo zvukovom spektre.

$$H(z) = \frac{1}{A(z)} \tag{4.2}$$

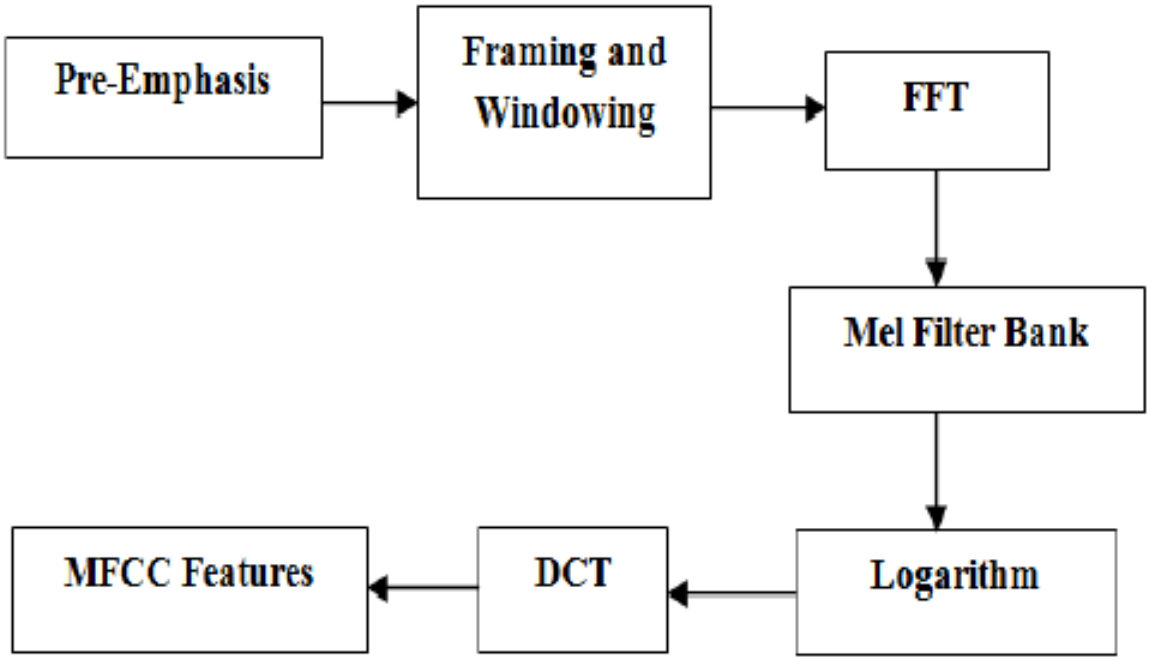
$A(z) = 1+a_1z^{-1}+...+a_pz^{-p}$ - polynóm rádu p, kde $p = 2k+1$, k je počet formantov Model sa nazývame, že je rádu p. Typické hodnoty p sú 10 pre vzorkovaciu frekvenciu 8kHz, pre vyššie frekvencie napr. 16. [15]

Ak máme model reči a chyby predikovaného signálu oproti skutočnému signálu, vieme reprodukovať signál, z ktorého boli tieto parametre vypočítané. Preto sa LPC používa aj na kompresiu rečového signálu, hlavne pri systémoch s nízkou prenosovou rýchlosťou. Na parametrizáciu signálu pre potreby rozpoznávania reči sa používajú parametre vymodelovaného filtra a_i.

[1] z angl. window functions

4.3 MFCC

MFCC(z angl. Mel-Frequency Cepstrum Coefficients) je v súčastnosti asi najpoužívanejšia technika získania príznakov z rečového signálu. Jednotlivé fázy tohto procesu môžme vidieť na obrázku 4.2.



Obr. 4.2: Blokový diagram extrahovania MFCC príznakov [16]

Prvá fáza *zvýraznenie signálu* je filter, ktorý zvýrazní vyššie frekvencie, ktoré boli potlačené počas vytvárania zvuku v ľudskom hlasovom trakte[16].

$$S(n) = X(n) - a \cdot X(n-1) \tag{4.3}$$

$S(n)$ - zvýraznená vzorka signálu, $X(n)$ - pôvodná vzorka, $X(n-1)$ - predchádzajúca vzorka, a - koeficient zvýraznenia, väčšinou v intervale $<0.95, 1>$ [16]

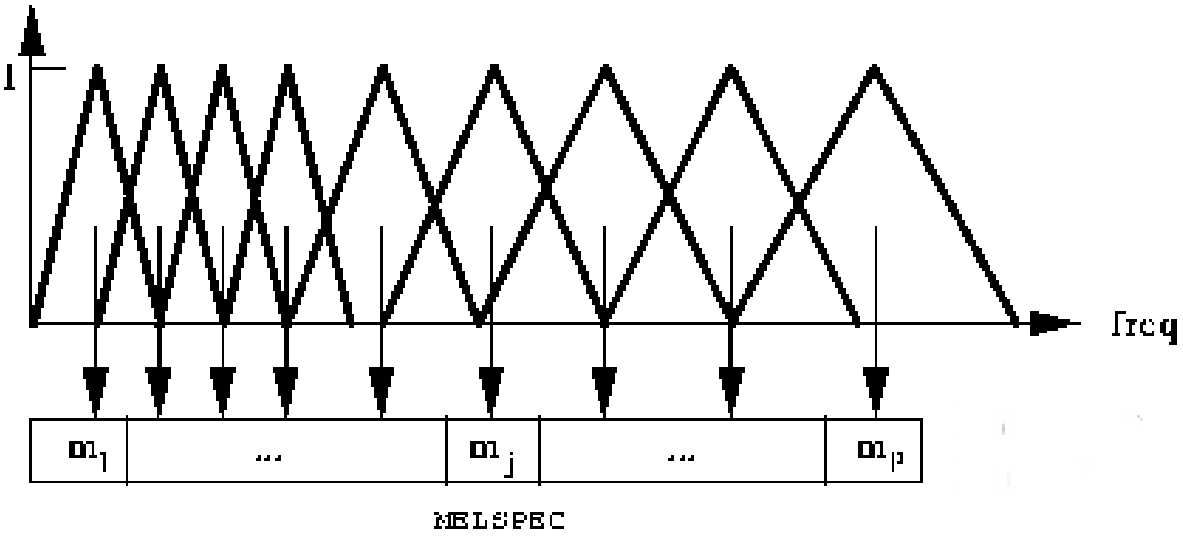
Druhá fáza je rozdelenie vstupného signálu na rámce. Tu sa používa Hammingovo okienko spomínané v časti 4.1.

Tretia fáza je rýchla fourierova transformácia (FFT - z angl. Fast fourier transform). FFT je rýchla implementácia Diskrétnej fourierovej transformácie(DFT - z angl. Discrete fourier transform), ktorá mení časovú doménu signálu do frekvenčnej domény. DFT je veľmi dôležitý nástroj v spracovaní signálu, často je výhodnejšie pracovať s frekvenčným spektrom ako so zvukovou vlnou vo forme sínusoidy. FFT vypočíta magnitúdu frekvencie vstupného rámca.

Štvrtou fázou je analýza pomocou banky filtrov v Melovej škále. Pozorovaním sa zistilo, že ľudské ucho vníma frekvencie nelineárne, preto sa frekvencie získané FFT prevedú do Melovej škály, ktorá sa viac snaží priblížiť k ľudskému vnímaniu frekvencií. Prevod frekvencie f sa realizuje nasledujúcim výpočtom:

$$M = 2595 \log_{10}(1 + \frac{f}{700}) \tag{4.4}$$

Ďalej vytvoríme banku prekrývajúcich sa trojuholníkových filtrov, ktoré sú rozložené pozdĺž Melovej škály(obr **??**). To znamená, že filtre pri vyšších frekvenciách budú širšie, ako tie pri nižších. Výstup každého filtra je príslušne ováhovaná suma spektrálnych komponentov.



Obr. 4.3: Analýza frekvenčného spektra pomocou banky filtrov [9]

Na výstup banky filtrov sa aplikuje logaritmus a následne diskrétna kosínusová transformácia (DCT - z angl. Discrete cosine transform). Tento proces slúži na prevod z Melovej spektrálnej domény naspať na časovú. DCT je definovaná vzorcom 4.5 [17].

$$X_k = \alpha \sum_{i=0}^{N-1} x_i \cdot \cos\{\frac{(2i+1)\pi k}{2N}\} \tag{4.5}$$

α je konštanta závislá na N.
Týmto spôsobom dostaneme MFCC vektory príznakov pre rečový signál.

Záver

Na záver už len odporúčania k samotnej kapitole Záver v bakalárskej práci podľa smernice [**?**]: „V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom. Rozsah záveru je minimálne dve strany. Záver ako kapitola sa nečísluje.“

Všimnite si správne písanie slovenských úvodzoviek okolo predchádzajúceho citátu, ktoré sme dosiahli príkazmi `\glqq` a `\grqq`.

V informatických prácach niekedy býva záver kratší ako dve strany, ale stále by to mal byť rozumne dlhý text, v rozsahu aspoň jednej strany. Okrem dosiahnutých cieľov sa zvyknú rozoberať aj otvorené problémy a námety na ďalšiu prácu v oblasti.

Abstrakt, úvod a záver práce obsahujú podobné informácie. Abstrakt je kratší text, ktorý má pomôcť čitateľovi sa rozhodnúť, či vôbec prácu chce čítať. Úvod má umožniť zorientovať sa v práci skôr než ju začne čítať a záver sumarizuje najdôležitejšie veci po tom, ako prácu prečítal, môže sa teda viac zamerať na detaily a využívať pojmy zavedené v práci.

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 Networkss, 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]

[14] Audacity audio software, https://www.audacityteam.org/ [navštívené 10.5.2019]

[15] Černocký H., Zpracování řečových signálů — studijní opora, 2006, Speech@FIT, Ústav počítačové grafiky a multimédií, Fakulta informačních technologií, Vysoké učení technické v Brně

[16] Deshmukh R.R., Saksamudre S.K., Isolated Word Recognition System for Hindi Language, JCSE-International Journal of Computer Sciences and Engineering, vol.4, Issue 7, July 2015

[17] Das B. P., Parekh R., Recognition of Isolated Words using Features based on LPC, MFCC, ZCR and STE, with Neural Network Classifiers, International Journal of Modern Engineering Research , Vol.2, Issue.3, May-June 2012