VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ADAPTACE ROZPOZNÁVAČE ŘEČI NA DATECH BEZ PŘEPISU

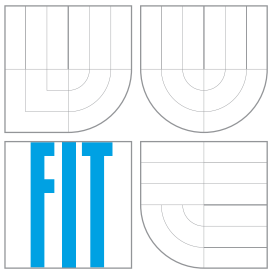
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JÁN ŠVEC

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ADAPTACE ROZPOZNÁVAČE ŘEČI NA DATECH BEZ PŘEPISU

UNSUPERVISED ADAPTATION OF SPEECH RECOGNIZER

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JÁN ŠVEC

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR SCHWARZ, Ph.D.

BRNO 2015

Abstrakt

Cílem práce je vytvořit a otestovat techniky pro adaptaci rozpoznávače řeči na audionahrávkach bez slovního přepisu. Nejprve připravíme data pro trenovaní rozpoznavače řeči a natrénujeme počáteční systém. Tímto rozpoznavačem přepišeme neznáma data a zaměříme se na experimentování s výběrem kvalitních adaptačních dat na základě míry kvality přepisu. Systém na nově vytvořené sadě přetrénujeme a vyhodnotíme úspešnost. Dále experimentujeme s množstvím adaptačních dat.

Abstract

The goal of this thesis is to design and test techniques for unsupervised adaptation of speech recognizers on some audio data without any textual transcripts. A training set is prepared at first, and a baseline speech recognition system is trained. This sistem is used to transcribe some unseen data. We will experiment with an adaptation data selection process based on some speech transcript quality measurement. The system is re-trained on this new set than, and the accuracy is evaluated. Then we experiment with the amount of adaptation data.

Klíčová slova

rozpoznávaní řeči, akustický model, jazykový model, konfidence, adaptace

Keywords

speech recognition, acustic model, language model, confidence, adaptation

Citace

Ján Švec: Adaptace rozpoznávače řeči na datech bez přepisu, diplomová práce, Brno, FIT VUT v Brně, 2015

Adaptace rozpoznávače řeči na datech bez přepisu

Prohlášení

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Petra Schwarza Ph.D.

.....
Ján Švec
3. júna 2015

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Petru Schwarzovi Ph.D. za odborné vedení, pomoc a cenné rady při vývoji této práce a pochopení dané problematiky. Dále chci poděkovat panu Ing. Martinu Karafiatovi Ph.D. za vstřícný přístup při vysvětlování dané problematiky na konzultacích.

Obsah

Kapitola 1

Úvod

Ľudská reč je najprirodzenejší prostriedok komunikácie aký si ľudstvo vytvorilo. Na počiatku si ľudia predávali informácie len touto formou. Postupom času prišlo na to, ako svoju reč a informácie v nej uchovávať. Žijeme v multimediálnej dobe, kde každý deň prebiehajú tisícky hodín telefónnych hovorov, televízneho a rádiového vysielania. Každá z týchto častí obsahuje množstvo informácií, ktoré je možné len ťažko získať. Pri takomto veľkom objeme dát nie je v ľudských silách všetky tieto dáta prezerať a získavať informácie. Človek, ale tieto informácie požaduje a preto si pomocou počítača stvoril pomocníka. Týmto pomocníkom je trieda počítačových programov nazvaných rozpoznávače. Aj keď tieto rozpoznávače nie sú bezchybné, už dnes uľahčujú človeku hľadanie informácií v takýchto typoch dát. Avšak, aby bolo možné takéto programy či systémy vytvoriť, je nutné získať adekvátne dáta. Pre rozpoznávač reči je nutné nazhromaždiť dostatočný počet audio súborov s rečou a k týmto súborom zodpovedajúce prepisy. V dnešnej dobe, nie je zväčša problém získať audio dáta, ktoré spĺňajú naše kritéria. Opačné je to však s prepismi takýchto dát. Existujú firmy, ktoré sa špecializujú na tvorbu a predaj sád dát pre rozpoznávače. Tieto sady dát sú však veľmi drahé. Taktiež je možné vytvoriť si takéto sady dát vo vlastných podmienkach. To však prináša zamestnanie tímu annotátorov, ktorí budú doslovne prepisovať slová z nahrávok. Časová náročnosť tvorby takýchto dát je však veľmi vysoká. Na 1 hodinu prepísaného audia pripadá 6 hodín práce annotátora. Ako je možné usúdiť, ani tento spôsob získavania dát nie je lacný.

Preto vznikajú snahy tieto náklady redukovať. Jedným z takýchto spôsobov je použiť existujúci rozpoznávač reči, prepísať neznáme dáta a prepisy použiť ako anotácie. Nie je však isté, ako takýto upravený rozpoznávač bude fungovať. A presne toto je cieľom tejto práce, zistiť ako sa jeden z týchto upravených rozpoznávačov bude chovať.

Práca je členená na kapitoly. Kapitola 2 popisuje základy štruktúry bežného rozpoznávača reči. Je tu popis akustického a jazykového modelu. Kapitola 3 predstaví použité sady dát a ich úpravy pre použitie v rozpoznávači tejto práce. Popis trénovania a adaptácie je načrtnutý v kapitole 4. Predposledná kapitola 5 podrobne predstavuje získané výsledky z experimentov. V záverečnej kapitole sú zhrnuté dosiahnuté výsledky.

Kapitola 2

Systém pre rozpoznávanie reči

Systém pre rozpoznávanie reči[1] (angl. ASR – Automatic Speech Recognition) je založený na prevode hovorenej reči na textovú reprezentáciu alebo nejakú jej formu (prepis). Komplexnosť takýchto systémov vo veľkej miere najčastejšie ovplyvňujú:

- *Variabilita rečníka.* Hlas jedného rečníka sa spravidla líši od hlasu iného rečníka. Je to spôsobené hlavne parametrami hlasového ústrojenstva a rečníkovou artikuláciou. Systémy, ktoré na základe týchto skutočností vznikajú sa delia na systémy na rečníkovi závislé (trénujú sa na hlas jediného rečníka, alebo malej skupiny) a systémy na rečníkovi nezávislé (model je trénovaný za pomoci hlasov stoviek až tisícov rečníkov)

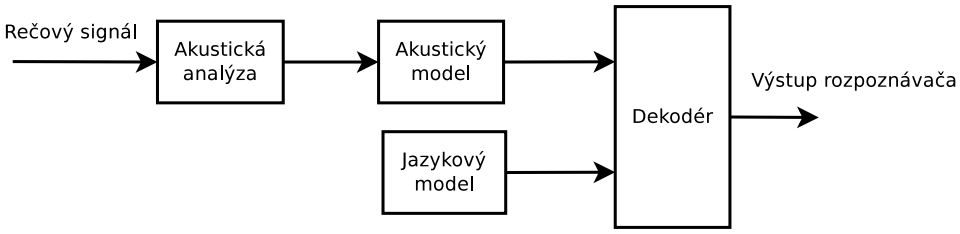
- *Variabilita hlasu.* Hlas jedného rečníka môže byť rôzny na základe situácie v ktorej sa rečník nachádza. Mení sa, ak povieme jednu frázu potichu, nahlas, v strese, alebo ak sme prechladnutý. Je preto nemožné aby jeden rečník jednu frázu v rôznych situáciach povedal rovnako.

- *Akustické prostredie a vlastnosti kanálu.* Akustické prostredie má významný dopad na kvalitu audio signálu, ktorý má byť prepísaný a taktiež na kanál, do ktorého sa hlas nahráva. Vznikajúci šum môže sťažovať odhad začiatku a konca slova. Systémy určené pre prácu v rušnom prostredí musia byť preto schopné odlíšiť hluk od reči.

- *Štýl reči v úlohe.* Úlohy rozpoznávania izolovaných slov (čísloviek, povelov ...) patria určite medzi najjednoduchšie. O niečo komplikovanejšie sú úlohy pri rozpoznávaní niekoľkých slov oddelených jasnými pauzami. Najzložitejšiu kategóriu tvoria úlohy rozpoznávania prirodzenej, spontánnej reči. Tá býva plná páuz medzi slovami, ktoré sú nesúmerné, ale i slová sú často nedokončené a taktiež rečník vkladá do svojho prejavu aj tzv. nerečové akustické udalosti (nádychy, myšlienkové váhania, smiech, kašeľ ...). Napríklad v slovenskom jazyku býva prejav plný nespisovných a hovorových slov. Taktiež voľné gramatické väzby, v tomto jazyku, medzi slovami sťažujú rozpoznávanie.

- *Veľkosť slovníka.* Rozpoznávač pri svojej práci využíva slovník pre výber slov, ktoré sa majú objaviť na jeho výstupe. Úlohy na rozpoznávač kladené využívajú malé slovníky (rozpoznávanie izolovaných slov) obsahujúce jednotky až desiatky slov, avšak úlohy používajúce veľké slovníky (rozpoznávanie plynulej reči) môžu používať až desiatky tisíc slov.

[1] Táto kapitola je založená na publikáciách [13] [12] [2] [8]



Obr. 2.1: Príklad štruktúry rozpoznávača.

Takéto typy rozpoznávačov najčastejšie vychádzajú z postupov porovnávania vzorov (angl. template matching), alebo štatistických metód. Porovnávanie vzorov bolo populárne v sedemdesiatych a osemdesiatych rokoch minulého storočia. Boli veľmi úspešné v úlohách rozpoznávania izolovaných slov. Štatistické metódy používajú techniky, kde celé slová, dokonca aj celý prejav môže byť modelovaný pomocou tzv. skrytých Markovových modelov. Každé slovo môže byť modelované jedným Markovovým modelom, avšak bežnejšie sa modely vytvárajú pre menšie rečové jednotky (slabiky, fonémy, trifóny ...). Pre modelovanie slova či sekvenciu slov sa tieto jednotky reťazia. V procese trénovania sa pre každú jednotku získavajú parametre Markovovho modelu. Rozpoznávanie neznámej postupnosti pozorovaní je vyhodnotené ako postupnosť slov získaná zreťazením modelov, pre obdržanie čo najväčšej aposteriórnej pravdepodobnosti.

2.1 Extrakcia príznakov

V počiatočnej fáze rozpoznávania do systému vstupuje diskretizovaný signál, ktorý je však príliš variabilný a pre samotné rozpoznávanie sa nehodí. Tu je možné použiť niektorú z techník extrakcie príznakov (angl. feature vectors), ktoré transformujú priebeh reči na reprezentáciu vhodnú pre rozpoznávač, ktorou býva postupnosť nízko dimenzionálnych vektorov. Extrakcia príznakov na začiatku vstupný signál rozdelí na segmenty(rámce) najčastejšie dlhé 10 – 25ms, u ktorých sa predpokladá, že modifikačné ústrojenstvo sa nachádza v jednej z konečného počtu artikulačných konfigurácií. Priebeh výpočtu vychádza z určenia spektrálnych vlastností segmentu. To súvisí zo snahou vytvoriť model ľudského vnímania zvuku. Postupom času sa však prichádzalo na metódy a postupy ako tieto príznaky využiť a získať ich spojením príznaky s lepšími vlastnosťami pre klasifikáciu.

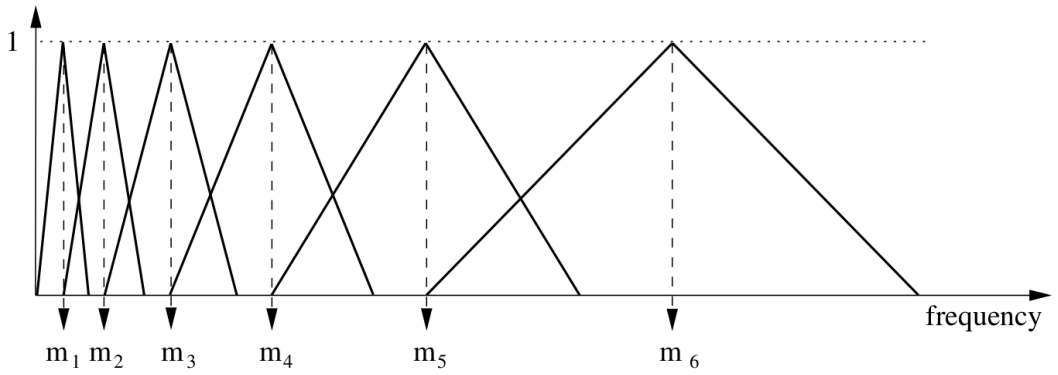
2.1.1 Zakladné príznaky

Jedna z najbežnejších techník pre extrakciu príznakov ktorú môžeme použiť je Melovského frekvenčná škála definovaná vzťahom:

$$f_m = 2595\ log_{10}\left(1 + \frac{f}{700}\right)\ , \tag{2.1}$$

kde f[Hz] je frekvencia v lineárnej škále a f_m[mel] predstavuje frekvenciu v nelineárnej škále. Táto škála aproximuje spôsob, akým ľudský sluch vníma zmeny vo frekvenciách. Proces výpočtu melovských kepstrálnych koeficientov(MFCC)[4], ktoré túto škálu využívajú, začína aplikovaním rýchlej Fourierovej transformácie na rámec rečového signálu vybraného najčastejšie Hammingovým oknom. Hlavnou časťou extrakcie je melovská filtrácia. Algoritmus takejto filtrácie je realizovaný bankou trojuholníkových pásmových filtrov z rovnomerným

rozložením stredných frekvencií jednotlivých trojuholníkových filtrov popri frekvenčnej osi, s merítkom v Melovského škále. Ukážka rozloženia je na obrázku 2.2. Počet pásiem je vhodné voliť podľa počtu a umiestnenia kritických pásiem. Trojuholníky sú zväčša rozmiestnené po celom frekvenčnom pásme od nuly až po vzorkovaciu frekvenciu.



Obr. 2.2: Rozmiestnenie filtrov na frekvenčnej ose. Prevzaté z prednášok ZRE.

Ďalším používaným prístupom je perceptuálna lineárna predikcia (PLP)[6]. Podobne ako metóda MFCC používa poznatky o vnímaní zvuku ľudským sluchom (odtiaľ perceptuálna). Začiatok spracovávania je rovnaký, výber rámca rečového signálu pomocou Hammingového okna a následná aplikácia rýchlej Fourierovej transformácie, pretože ľudské vnímanie zmien zvuku nie je lineárne, ale logaritmické, čo je ovplyvnené tzv. maskovaním zvukov. Šírka pásma, kde toto maskovanie prebieha sa nazýva šírka kritického pásma a je premenlivá s frekvenciou. Preto PLP tento jav modeluje nelineárnou transformáciou pôvodnej osi frekvencie ω[rad/s] na osu frekvencie $\Omega(\omega)$ meranú v jednotkách bark podľa vzorca

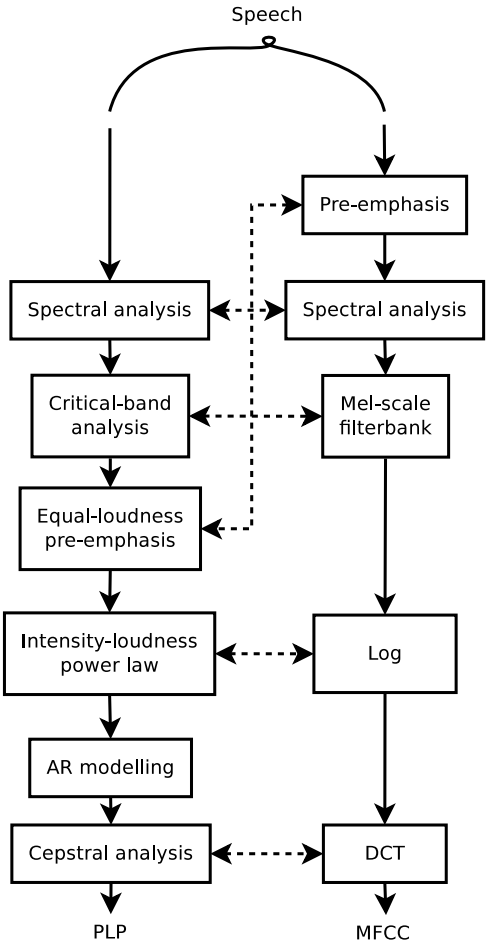
$$\Omega(\omega) = 6\ ln\left(\frac{\omega}{1200\pi} + \sqrt{\left(\frac{\omega}{1200\pi}\right)^2 + 1}\right), \tag{2.2}$$

kde $\omega = 2\pi f$ [rad/s] a $\Omega(\omega)$ [bark] a konštrukciou maskujúcich kriviek ktoré simulujú kritické pásmo počuteľnosti. Človek taktiež vníma intenzitu zvuku v závislosti na frekvencii ako hlasitosť. Aby bolo možné prispôsobiť výkonové spektrum $P(\omega)$ tejto vlastnosti sluchu prevedieme v ďalšom kroku preemfázu diskrétnych vzoriek kriviek predstavujúce pásmový filter m–tého kritického pásma a zodpovedajúcich si hodnôt aproximujúcich krivky $E(\omega)$

$$E(\omega) = K\ \frac{\omega^4(\omega^2 + 56,9).10^6}{(\omega^2 + 6,3.10^6)^2(\omega^2 + 379,4.10^6)(\omega^6 + 9,6.10^26)}, \tag{2.3}$$

kde $\omega = 2\pi f$ a konštantu K je vhodné nastaviť tak, aby kritický pásmový filter dosiahol hodnotu 0 dB práve v najvyššej hodnote intenzity. Funkcia $E(\omega)$ je teda navrhnutá pre aproximáciu citlivosti ľudského sluchu v odlišných frekvenciách. Ďalšími krokmi je vážená spektrálna sumarizácia vzoriek výkonového spektra a uplatnením vzťahu o vnímaní intenzity zvuku a jeho hlasitosti. Posledný krok je aproximácia spektrom celopoľového modelu.

Záverom je možné tieto metódy porovnať. Ako ukazuje obrázok 2.3, obidve metódy používajú rámcovaný vstupný signál, ktorý je spracovávaný FFT. Obidve metódy používajú poznatky o vnímaní zvuku človekom a simulujú ich pomocou filtrov, avšak metóda PLP používa pokročilejšie postupy ako napr. aplikácia experimentálne zisteného vzťahu pre intenzitu a hlasitosť.

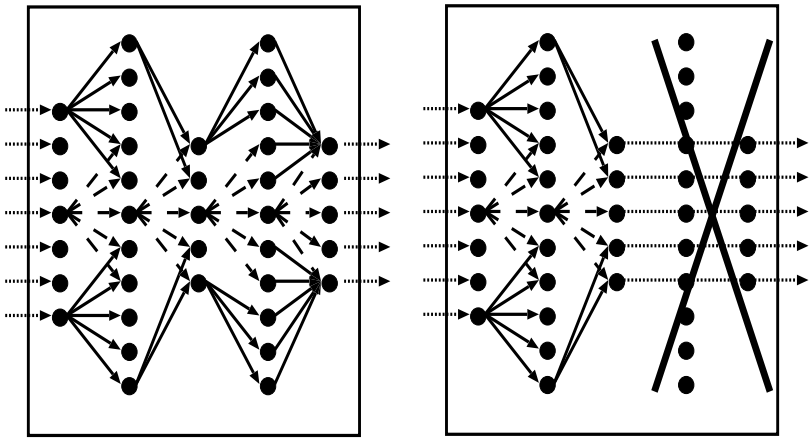


Obr. 2.3: Porovnanie výpočtu MFCC a PLP. Prevzaté a upravené z [9].

2.1.2 Hybridné príznaky

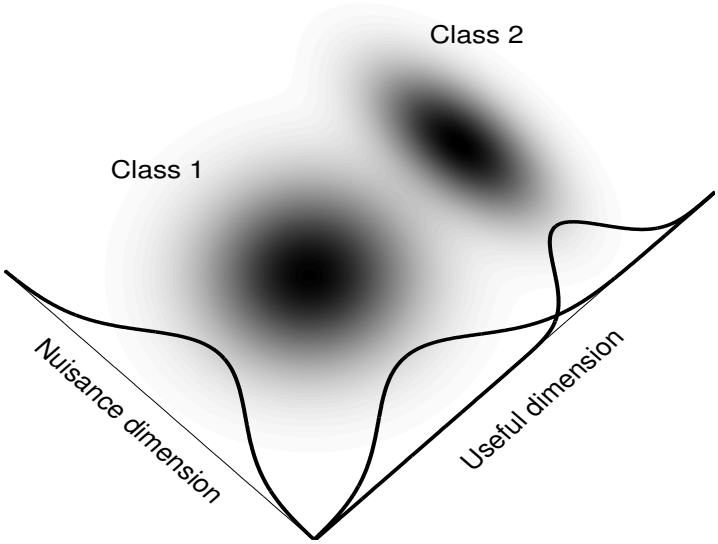
V predchádzajúcej časti boli načrtnuté príznaky, ktoré sa bežne používajú, ale od systémov pre rozpoznávanie sa požaduje stále vyššia presnosť a robustnosť. Práve preto sú snahy tieto príznaky vylepšiť. Jedným z takýchto vylepšení je použiť tzv. bottleneck príznaky (bottleneck = hrdlo fľaše). Bottleneck príznaky získame priamo z neurónovej siete. Konštrukčne je táto sieť klasickou doprednou neurónovou sieťou s back–propagation učiacim algoritmom, avšak jedna z jej skrytých vrstiev je zúžená. Sieť sa pri trénovaní pokúša nájsť nelineárnu transformáciu medzi vstupným vektorom príznakov a množinou rečových jednotiek. Po ukončení trénovania sa koncová časť siete odstráni a ako výstup zostane zúžená vrstva. Potom sa príznaky dekódujú takto upravenou sieťou na nové.

Ako ďalšie postupy sa uplatňujú lineárne transformácie v príznakovom priestore. Jednou z takých je *Lineárna diskriminačná analýza* (LDA), ktorá sa pokúša nájsť lineárnu transformáciu, čím spôsobí redukciu dimenzií. Transformácia sa vykonáva v smeroch, kde je možné triedy čo najlepšie oddeliť a tým zachovať diskrimináciu medzi týmito triedami. Tiež sa predpokladá, že rozloženie príznakov je gaussovské a využíva jednu kovariančnú maticu. Keď však pre každú triedu máme rozdielnu kovariančnú maticu, takúto transformáciu nazývame *Heteroskodická lineárna diskriminačná analýza* (HLDA). Táto transformácia sa používa u rozpoznávačov reči tam, kde chceme pokryť kontext napr. pre aktuálny prízna-



Obr. 2.4: Ukážka bootleneck siete. Vľavo konfigurácia počas trénovania. Vpravo konfigurácia pre dekódovanie. Prevzaté z [11].

kový vektor PLP o rozmeroch 13 koeficientov vypočítame prvú, druhú a tretiu deriváciu, čím spolu získame vektor príznakov o veľkosti 52 dimenzií. Na takýto vektor sa aplikuje HLDA, čim môžeme získať nový príznak HLDA–PLP o veľkosti 39 dimenzií[5].



Obr. 2.5: Porovnanie výpočtu MFCC a PLP. Prevzaté a upravené z [3].

2.2 Klasifikácia

Úloha rozpoznávania reči sa dá vnímať ako problém dekódovania, zo snahou maximalizovať aposteriórnou pravdepodobnosť. Majme postupnosť slov $W = \{w_1, w_2 \dots w_M\}$ kde M je počet slov v postupnosti a $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \dots \mathbf{o}_N\}$ je postupnosť príznakových vektorov zís-

kaných z rečového signálu z ktorých sa systém pokúša rozpoznať, aké slová boli rečníkom vyslovené. Cieľom je nájsť takú postupnosť slov $\hat{W}$, ktorá maximalizuje podmienenú pravdepodobnosť $P(W|\mathbf{O})$ t.j. najpravdepodobnejšiu postupnosť slov W pre danú postupnosť príznakových vektorov $\mathbf{O}$. Túto skutočnosť môžeme zapísať pomocou Bayesova pravidla:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ P(W|\mathbf{O}) = \underset{W}{\operatorname{argmax}}\ \frac{P(\mathbf{O}|W)P(W)}{P(\mathbf{O})} \tag{2.4}$$

kde

- $P(\mathbf{O}|W)$ je podmienená pravdepodobnosť, ak vyslovíme postupnosť slov W, obdržíme postupnosť príznakových vektorov $\mathbf{O}$.
- $P(W)$ je apriórna pravdepodobnosť W, predstavujúca vyslovenie rečníkom postupnosť slov W .
- $P(\mathbf{O})$ je apriórna pravdepodobnosť $\mathbf{O}$.

a keďže $P(\mathbf{O})$ nieje závislá na W môžeme upraviť vzorec:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ P(W|\mathbf{O}) = \underset{W}{\operatorname{argmax}}\ P(\mathbf{O}|W)P(W) \tag{2.5}$$

z toho vyplýva, že rozpoznávanie je možné riešiť pomocou dvoch oddelených pravdepodobností, ktoré je možne trénovať a vyhodnocovať taktiež samostatne. Podmienená pravdepodobnosť $P(O|W)$ predstavuje informáciu o akustickom modele. Pravdepodobnosť $P(W)$ zasa informáciu o jazykovom modele.

V nasledujúcich kapitolách bude načrtnuté prevedenie akustického a jazykového modelu.

2.3 Akustický model

Úlohou akustického modelu je čo najpresnejšie a najrýchlejšie odhadnúť podmienenú pravdepodobnosť $P(\mathbf{X}|W)$. Model by mal spĺňať čo najlepšiu flexibilitu, presnosť a účinnosť. Flexibilita systému je nutná, keďže podmienky trénovania a nasadenia systému nebývajú rovnaké. Presnosť je vyžadovaná pre odlíšenie akusticky podobných, ale lingvisticky odlišných jednotiek. Posledná vlastnosť–účinnosť, je občas kľúčová pri nasadení v aplikáciach, kde je nutná odozva systému v reálnom čase.

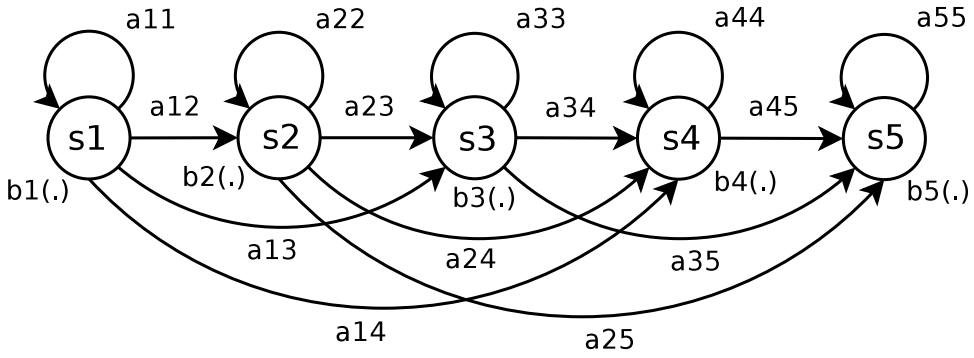
Ako veľmi efektívny spôsob riešenia tejto úlohy sa ukázalo použitie tzv. skrytých Markovových modelov (angl. Hidden Markov Model, zkr. HMM). Skrytý Markovov model je model stochastického procesu, ktorý je možné chápať ako pravdepodobnostný konečný automat, ktorý v diskrétnych časových okamihoch generuje náhodnú postupnosť pozorovaní $\mathbf{O} = \mathbf{o}_1, \mathbf{o}_2 \ldots \mathbf{o}_T$. Príklad modelu je možné vidieť na obrázku 2.6.

Pravdepodobnosť prechodu a_{ij} je podmienená a určuje s akou pravdepodobnosťou prechádza model zo stavu s_i, v čase t do stavu s_j v čase $t+1$ vyjadrenú:

$$a_{ij} = P(s(t+1) = s_j | s(t = s_i)) \tag{2.6}$$

Predpokladajme, že prechodová pravdepodobnosť a_{ij} je v čase konštantná a pre všetky stavy je splnená podmienka

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad i = 1, 2, \ldots N \tag{2.7}$$



Obr. 2.6: Príklad 5–stavového skrytého Markovovho modelu slova.

Funkcia rozdelenia výstupnej pravdepodobnosti $b_j(\mathbf{o}_t)$ popisuje rozdelenie pravdepodobnosti pozorovania $\mathbf{o}_t$ produkovaného v stave s_j v čase t. Ak pozorovanie nadobúda konečný počet diskrétnych hodnôt hovoríme o pravdepodobnosti pre hodnotu spojitej náhodnej veličiny, ktorá funkciou $b_j(\mathbf{o}_t)$ hustoty pravdepodobnosti javu. Všeobecne

$$b_j(\mathbf{o}_t) = P(\mathbf{o}_t|s(t) = s_j) \tag{2.8}$$

Pre diskrétne a spojité rozdelenie pritom platí

$$\sum_o b_j(\mathbf{o}_t) = 1 \qquad \int_o b_j(\mathbf{o}_t)d\mathbf{o} = 1 \tag{2.9}$$

a to pre všetky emitujúce stavy, t.j stavy, ktoré sú schopné generovať výstupný vektor pozorovania.

Rozdelenie výstupnej pravdepodobnosti musí byť dostatočne špecifické, aby bolo možne od seba oddeliť rôzne zvuky a zároveň dostatočne robustné, aby pokrylo variabilitu reči. Na tento účel sa v bežných prípadoch používa zmes normálnych rozložení (angl. Gaussian Mixture Models, zkr. GMM). Ďalším možným riešením je použiť umelú neurónovú sieť, kde vstupom sú príznakové vektory a výstupom je aposteriórna pravdepodobnosť $P(\mathbf{O}|S)$ jednotlivých stavov HMM.

Pre získanie parametrov HMM sa používa štatistická indukcia, inak trénovanie (odhad, estimácia) z presne „popísaných“ (anotovaných) trénovacích akustických dát. Najčastejšia metóda pre odhad sa využíva metóda maximálnej vierohodnosti (angl. Maximum Likelihood, zkr. ML). Pri úlohe modelovania reči HMM existuje veľmi efektívna metóda pre trénovanie parametrov modelu, ktorá má základ v kritériu maximálnej vierohodnosti. Pre maximalizáciu vierohodnosti funkcie sa v takom prípade používa iteratívna procedúra tzv. Baum–Welch algoritmus, ktorý je špeciálnym prípadom algoritmu očakávanie–maximalizácia (angl. Expectation–Maximalization, zkr. EM).

Pre výpočet pravdepodobnosti generovania prejavu modelom, ak predpokladáme, že máme odhadnuté parametre, platí

$$P(\mathbf{O}|\lambda) = \sum_S P(\mathbf{O}, S|\lambda) = \sum_S a_{s(0)s(1)} \prod_{t=1}^{T} b_s(t)(\mathbf{o}_t) a_{s(t)s(t+1)} \tag{2.10}$$

λ ako parametre HMM. Týmto spôsobom z hľadiska operácií je to často nerealizovateľné (až $2TN^T$ kde N je celkový počet stavov modelu a T je počet vektorov pozorovania).

Z tohoto dôvodu bol navrhnutý omnoho efektívnejší algoritmus spôsobu výpočtu, tzv. algoritmus forward–backward, ktorý vyžaduje len N^2T. Takto prevádzaný priamy výpočet podľa tohoto algoritmu vedie obvykle k numerickému podtečeniu a preto sa pri rekurzívnom výpočte využívajú normalizačné koeficienty, alebo logaritmus pravdepodobnosti, ktoré tento nedostatok eliminujú. Výpočet podmienenej pravdepodobnosti je možné alternatívne aproximovať pravdepodobnosťou $P_S(\mathbf{O}|\lambda)$, ako najpravdepodobnejšiu postupnosť stavov, ktorými prejde modelom λ postupnosť $\mathbf{O}$

$$P_S(\mathbf{O}|\lambda) = \max_S P(\mathbf{O}, S|\lambda) = \max_S \left[a_{s(0)s(1)} \prod_{t=1}^{T} b_{s(t)}(\mathbf{o}_t) a_{s(t)s(t+1)}\right] \tag{2.11}$$

Túto pravdepodobnosť i optimálnu postupnosť stavov je možné určiť pomocou tzv. Viterbiova algoritmu, ktorý rieši problém rekurzívne využitím techniky dynamického programovania.

Popri trénovaní pomocou ML existujú aj iné metódy. Jednou z takýchto metód je *Minimum Phone Error* (zkr. MPE) patriaca do triedy diskriminačného trénovania. Tento typ trénovania namiesto maximalizácie likelihoodu, používa maximalizáciu hodnoty objektívnej funkcie. Tá obsahuje aj meranie počtu správne rozpoznaných fonémov, alebo celých slov ktorú zohľadňuje pri trénovaní. Potom hovoríme o Minimum Word Error(zkr. MWE).

Pre jednoduché úlohy rozpoznávania niekoľkých slov je možné HMM vytvoriť pre každé slovo samostatne, ale pre úlohy s veľkým slovníkom, ktorý obsahuje desiatky tisíc slov, nie je možné nazhromaždiť dostatočné množstvo trénovacích dát pre každé slovo. Preto častejšie rozpoznávače používajú HMM modelujúce subslovné jednotky (ako napríklad fonémy). Tie však nepokrývajú akustickú rôznorodosť a z toho dôvodu je vhodné akustické elementy modelovať viac komplexnejšie. Takýmto možným vylepšením bývajú tzv. kontextovo závislé fonémy, ktoré môžu byť modelované kontextovo závislými HMM. Najbežnejší takýto kontextový foném sa používa trifón štruktúry zápisu $L - F + P$, kde foném F leží medzi ľavým L a pravým P fonémom.

2.4 Jazykový model

Jazykový model (zkr. LM) je ďalšia časť systému pre rozpoznávanie reči. Úlohou tejto časti je čo najrýchlejšie a najpresnejšie odhadnúť apriórnu pravdepodobnosť $P(W)$ pre ľubovoľnú postupnosť slov. Pri vytváraní takéhoto modelu musíme myslieť na určité zákonitosti, ktoré sú pre každý jazyk špecifické. Prvá je slovník, ktorý obsahuje slová používané jazykom. Každé slovo má priradenú výslovnosť, môže ich byť aj viac, popisovanú ako postupnosti zvukov (fonémov) modelu jazyka. Druhou sú pravidlá, podľa ktorých sú slová reťazené do vetných celkov. Tu je snaha modelu vystihnúť gramatiku daného jazyka, avšak u jazykov ako slovenčina a čeština je to veľmi ťažké. V takýchto prípadoch sa LM zostavuje pre doménu pôsobenia daného rozpoznávača.

Jazykový model by mal byť schopný určiť pravdepodobnosť postupnosti W, ktorá zahŕňa K slov, ktorú je možné určiť všeobecne podľa vzťahu

$$P(W) = P(w_1^K) = P(w_1 w_2 \dots w_K) = \prod_{i=1}^{K} P(w_i|w_1^{i-1}) \tag{2.12}$$

a pre ľubovolný začiatok $w_1 w_2 \dots w_k \quad (k \leq K)$ podobne platí

$$P(w_1^k) = P(w_1^{k-1}) P(w_k|w_1^{k-1}) \qquad k = 2, \dots, K. \tag{2.13}$$

Všetky tieto pravdepodobnosti je veľmi ťažké a takmer nemožné vyčísliť. Preto sa v praxi používa ich aproximácia, pre všetky histórie $w_1 \dots w_{i-2}w_{i-1}$ pričom pri zhodnej dĺžke histórie posledných $n-1$ slov sa zaradia do rovnakej triedy. Takto aproximované modely sa nazývajú n–gramové modely. Slovom n–gram sa rozumie postupnosť n za sebou idúcich slov v pozorovaní napr. v trénovanom textovom korpuse. Používajú sa najčastejšie varianty takýchto modelov ako unigramy($n = 1$), bigramy($n = 2$) a trigramy($n = 3$).

2.5 Rozpoznávacia sieť

Všetky predchádzajúce časti akustického a jazykového modelu môžu byť reprezentované tzv. *váhovanými konečnými transducermi WFST* (anglicky Weightet finite-state transducer). Vďaka tejto reprezentácii je možné tieto časti spájať pomocou operácie kompozície do jednej rozpoznávacej siete a tú ďalej optimalizovať. Váhový transducer je konečný automat, ktorý má na svojich prechodoch aj vstupné a výstupné symboly. Každý prechod má taktiež váhu, ktorá môže reprezentovať pravdepodobnosť alebo cenu prechodu. S váhovými transducermi je možné vykonávať operácie determinizácie, minimalizácie a stlačovanie. Výslednú sieť je potom možné prehľadávať Viterbiovým algoritmom. Typický váhovaný konečný transducer používaný pre rozpoznávanie reči pozostáva z častí:

- **G** – je transducer predstavujúci gramatiku daného jazyka, popisom všeobecného usporiadania n–tic slov v jazyku.
- **L** – výslovnostný slovník, popisujúci výslovnosť slov a taktiež pravdepodobnosť výskytu jednotlivých fonémov v slove.
- **C** – kontextový rozklad slov na trifóny. Popisuje ľavý a pravý kontext k aktuálnemu fonému. Kontext taktiež postihuje i medzi slovami.
- **H** – reprezentuje akustický model(zväčša pomocou HMM).

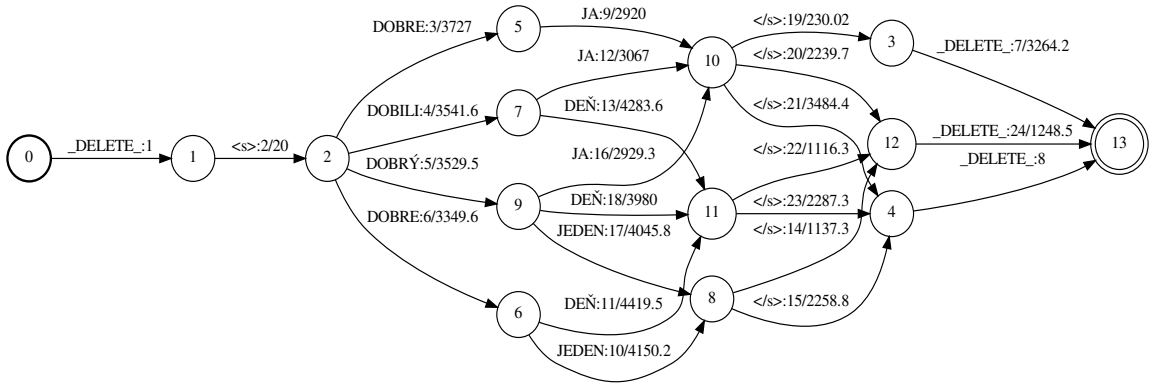
Výslednú sieť je možné komponovať:

$$H \circ C \circ L \circ G \tag{2.14}$$

Operátor $\circ$ značí kompozíciu, ktorá sa prevádza sprava doľava. V jednotlivých krokoch tejto kompozície pre vytvorenie celkového automatu prebieha determinizácia a minimalizácia pre redukciu veľkosti celého automatu a jeho efektívneho behu.

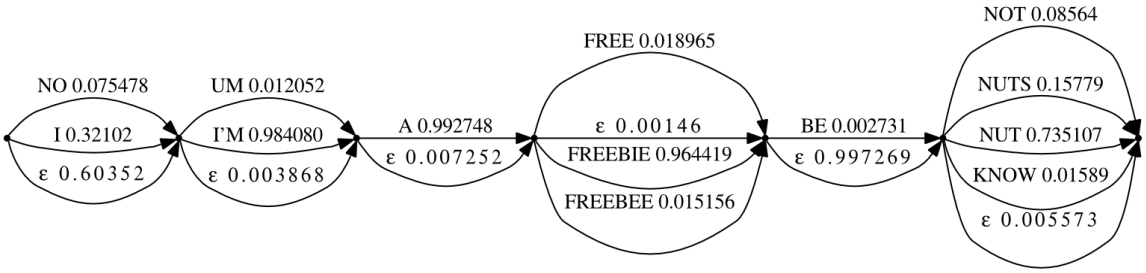
2.6 Výstupy

Výstupy rozpoznávača reči najčastejšie spadajú do jednej z 3 kategórií formátu výstupu, ktoré spolu navzájom súvisia. Najvšeobecnejším typom výstupu je *mriežka hypotéz*(angl. lattice). *lattice* je komplexná grafová štruktúra, vytvorená z hypotéz zostavených počas priebehu spracovávania rečových dát rozpoznávačom. Štrukturálne sú to orientované, acyklické grafy pozostávajúce z uzlov a hrán. Hrany vedú cesty v hypotetickej sekvencii slov. Nesú taktiež informáciu o vierohodnosti tejto cesty, podľa akustického a jazykového modelu k danému slovu. Uzol uchováva informáciu o svojom začiatočnom a koncovom čase a o hypotetickom slove. Tento výstup je plný dodatočných informácií o prepise a je preto často používaný pre dodatočnú analýzu. Príklad lattice pre krátky prejav je na obrázku 2.7.



Obr. 2.7: Ukážka lattice pre krátky prejav ziskaná z rozpoznávača firmi Phonexia[1].

Ďalšou možnou reprezentáciou vychádzajúcou z lattice je *Sieť zámien* (angl. Confusion Network). Je to taktiež grafová reprezentácia, ktorá sa získava z lattice, jej zarovnaním do časových okien, v ktorých sa každé slovo môže vyskytovať iba raz, čo je vhodné k indexácii Pre každé slovo je možné stanoviť aposteriórnu pravdepodobnosť príslušnosti k danému časovému oknu. Tento výstup už neobsahuje toľko dodatočných informácií k čistým prepisom ako lattice. „Stratené“ informácie však bývajú transformované na informácie iné.



Obr. 2.8: Ukážka siete zámien, prevzaté z [13].

Posledný výstup má charakter *N–Najlepších prepisov* (angl. N–Best), kde najčastejšie $N = 1$, čo predstavuje najpravdepodobnejšiu sekvenciu slov vyslovených v rečových dátach. Tento výstup býva používaný najčastejšie

2.7 Hodnotenie

Stále vznikajú nové systémy pre rozpoznávanie reči. Aby bolo možné tieto systémy medzi sebou porovnávať, museli byť stanovené jasné kritériá a metriky pre ich hodnotenie. V tejto

kapitole bude popísaná jedna z najčastejších metrík, akými bývajú systémy hodnotené. Taktiež tu bude načrtnuté meranie vierohodnosti (tzv. confidence) používané pre určenie správnosti hypotézy vzhľadom k ostatným hypotézam.

2.7.1 Word Error Rate

Pre hodnotenie výkonu rozpoznávača reči sa používajú metriky založené na samotnej výstupnej postupnosti slov. Najbežnejšia takáto metrika je *Word Error Rate* (WER)[7]

$$WER = \frac{S + D + I}{N} \tag{2.15}$$

alebo

$$WER = \frac{S + D + I}{S + D + C} \tag{2.16}$$

kde

- **Substitučné chyby** (angl. Substitution error, S) nastávajú, ak referenčné slovo a slovo z hypotézy na neho zarovnané sa nezhodujú.
- **Chyby zmazania** (angl. Deletion error, D) sú prípady, kedy referenčné slovo nie je možné zarovnať na slovo v hypotéze.
- **Chyby vloženia** (angl. Insertion error, I) ktoré sa objavujú, keď sa v hypotéze vyskytne slovo, ktoré nie je možné zarovnať na slovo v referencii.
- Premenná **C** predstavuje počet správne rozpoznaných slov v hypotézach.
- Premenná **N** predstavuje počet slov v referenčnom prepise ($N = S + D + C$).

Taktiež sa používajú hodnotiace metriky presnosť a správnosť, ktoré z WER vychádzajú

$$Acc = \frac{hits}{N} \tag{2.17}$$

$$Corr = 1 - WER \tag{2.18}$$

Táto metrika je založená na vyhodnotení počtu slov, ktoré sú rozdielne medzi hypotézami prepísanými rozpoznávačom a referenčnými prepismi. Avšak takéto prepisy sa nemusia zhodovať. Nie je ich možné porovnávať jednoducho podľa indexu slova v sekvencii, ale je potrebné použiť sofistikovanejší prístup pre zarovnanie, pomocou techník dynamického programovania. To vykonáva procedúra minimalizácie Levenshtein vzdialenosti medzi dvoma sekvenciami, definované ako vážená suma výskytov chýb popísaných vyššie. WER je potom vypočítaná ako podiel súčtu jednotlivých chýb k počtu referenčných slov.

Na „oskórovanie“ touto metrikou je použitá najlepšia hypotéza, známa ako 1–Best prepis, akú nám systém môže poskytnúť. Ďalšia metrika pre hodnotenie prepisov môže byť napríklad *oracle error rate*, ktorý je najnižší možný WER cez zoznam N uvažovaných prepisov (uvažuje sa N–Best).

2.7.2 Meranie dôveryhodnosti

Meranie dôveryhodnosti (ang. confidence) prepisov rozpoznávača reči je neoceniteľným zdrojom informácií pre iné systémy, ktoré s nimi spolupracujú. Techniky zachytávania slov (ang. keyword spotting) môžu byť celé postavené na meraní dôveryhodnosti. Takéto aplikácie poskytovali prvotnú motiváciu pre vývoj techník merania dôveryhodnosti. Toto meranie získalo popularitu v dialógových systémoch, v systémoch pre prepis s vysokými dátovými tokmi, taktiež u systémoch v nich kombinovaných a pre adaptáciu akustického modelu.

Problém presného merania dôveryhodnosti pre LVCSR systémy je však veľmi náročný a nie je úplne vyriešený. Zlepšovanie kvality merania dôveryhodnosti máva nespochybniteľný význam pre rozpoznávač, pretože účinnosť väčšiny teoretických aplikácií tohoto skóre a jeho spoľahlivosť bola obmedzená nedostatkom presného skóre.

Najčastejšie sa využívajú postupy postavené na aposteriórnej pravdepodobnosti. Moderné ASR systémy vykonávajú svoju činnosť pomocou aplikácie štatistických princípov. Ak by rozpoznávače boli perfektné, nepotrebovali by meranie dôveryhodnosti. Z toho vyplýva, že základným predpokladom pri takomto meraní musí byť predpoklad, že typický rozpoznávač robí chyby. Avšak v dôsledku štatistickej povahy systémov rozpoznávania reči, môže byť ním samým odhadnutá dôvera v hypotézy, čo je už dobrým ukazovateľom skutočnej presnosti výsledného najlepšieho prepisu.

Je možné formulovať hypotézu sekvencie slov s maximálnou aposteriórnou pravdepodobnosťou. Táto aposteriórna pravdepodobnosť už môže byť interpretovaná ako dôveryhodnosť. Tu je možné použiť metodiky z HMM systémov. Bežná praktika u týchto systémov smeruje predovšetkým k tomu, presne modelovať distribúciu pozorovaní cez všetky možné hypotézy. Táto úloha u systémov s veľkým slovníkom je však neriešiteľný problém. Preto táto pravdepodobnosť musí byť odhadnutá inak.

Vhodnými metódami sú tie, ktoré táto pravdepodobnosť aproximuje. Lattice predstavujú kompaktnú reprezentáciu najviac zohľadňovaných hypotéz generovaných počas rozpoznávania. Hypotézy v latticiach predstavujú sadu možných riešení v danom časovom priestore. Výpočet aposteriórnej pravdepodobnosti nad latticami je preto dobrou aproximačnou metódou. To ale stále predstavuje početne náročnú úlohu, tak sa používa prepočet lattic na N-best hypotézy.

Prvým krokom takého algoritmu je výpočet aposteriórnej pravdepodobnosti každej hrany lattice. Likelihoody jazykového modelu (LM) a akustického modelu (AM) ktoré sú uložené v latticiach a sú použité pre výpočet pravdepodobnosti. Pri definícii cesty q cez latticu, ktorá obsahuje slovo w v sekvencii slov $\mathbf{W}$, a obdržanej sekvencii pozorovaní $\mathbf{X}$, môže byť pravdepodobnosť vyčíslená:

$$p(q, \mathbf{X}) = \underbrace{p(\mathbf{X}|q)^{\frac{1}{\gamma}}}_{AM} \underbrace{P(\mathbf{W})}_{LM} \tag{2.19}$$

kde koeficient merítka γ sa používa na potlačenie hodnoty akustickej pravdepodobnosti skôr ako pre navýšenie pravdepodobnosti jazykového modelu. Následná aposteriórna pravdepodobnosť na hrane môže byť určená sumou ciest latticou, ktoré prechádzajú danou hranou:

$$p(a, \mathbf{X}) = \frac{\sum_{\mathbf{Q}_a} p(q, \mathbf{X})}{p(\mathbf{X})}$$

kde $\mathbf{Q}_a$ predstavuje množinu ciest danou hranou. Takáto suma môže byť efektívne vypočítaná pomocou forward-backward algoritmu. Lattice obsahujú hrany, v ktorých sa vyskytujú rovnaké slová, avšak rôzne časovo segmentované a v rôznych n–gramových kontextoch. Pre

výpočet pravdepodobnosti slova však tieto okolnosti zanedbávame a nezohľadňujeme ich vo výpočtoch. V ďalšej fáze sa rieši agregáciou takýchto hrán obsahujúcich rovnaké slovo v príhovore.

Kapitola 3

Data

Hlavným cieľom tejto práce bolo postaviť rozpoznávač reči a previesť jeho adaptáciu. Tento cieľ by nebolo možné dosiahnuť bez zodpovedajúcich dát.

3.1 Sady dát

Pre trénovanie, testovanie a adaptáciu boli použité 2 sady dát. Prvá obsahuje nahrávky rozhovorov zväčša dvoch osôb (operátor, klient), ktorí medzi sebou komunikovali. Obsah dát bol orientovaný na oblasť telekomunikácií (mobilné paušály, ich tarify, akcie na služby mobilných operátorov, internet), nahrávaných call-centrom, v ktorých je primárny slovenský jazyk. Spolu sada obsahuje 897 nahrávok o priemernej dĺžke 3 minúty, vzorkovacou frekvenciou 8 kHz, s veľkosťou vzorku 16 bitov a jedným kanálom. Ku každému dátovému súboru bol priložený textový prepis obsahu. Transkripcie k týmto akustickým dátam boli vyhotovené v programe Transcriber a boli uložené pomocou jeho interného formátu v súboroch s koncovkou *trs*. Túto sadu vlastní firma Phonexia s.r.o. a jej presný obsah je klasifikovaný ako citlivý. V častiach textu budeme túto sadu nazývať *Sada1*. Táto sada bola použitá ako primárna. Bola z nej vyčlenená testovacia podsada používaná počas celého priebehu práce. Zvyšná časť sady sa používa pre trénovacie a adaptačné účely v pomere ako to požadovali experimenty. Podrobnejšie informácie, ktoré je možné o sade dát verejne uverejniť a jej delenie možno nájsť v tabuľke 3.1.

Druhá poskytnutá sada bola veľmi podobná. Obsah pozostával z rozhovorov medzi 2 osobami a témy týchto rozhovorov boli rôzne a týkali sa bežného života (nakupovanie, cestovanie, telefonovanie, ...). U tejto sady neboli ku každému akustickému súboru poskytnuté zodpovedajúce prepisy. Prepisy boli taktiež vo formáte programu Transcriber. Spolu sada obsahuje 861 nahrávok o priemernej dĺžke 2 minúty. Formát súboru bol totožný ako u Sady1 (8kHz, 16lin, 1ch). V častiach textu budeme túto sadu nazývať *Sada2*. Dáta v tejto sade boli použité výhradne pre účely adaptácie. Podrobnejšie informácie je možné nájsť v tabuľke 3.2.

	časť pre trénovanie / adaptáciu	testovacia podsada
Dĺžka audia	cca 52 hodín	cca 0.5 hodiny
Dĺžka reči	cca 30 hodín	cca 0.3 hodiny
Segmenty	41296	444

Tabuľka 3.1: Vlastnosti dát Sada1.

Dĺžka audia	cca 25 hodín
Dĺžka reči	cca 14 hodín
Segmenty	39184

Tabuľka 3.2: Vlastnosti dát Sada2.

Typ textov	Počet textov	Obsah vo výslednom modeli
všeobecné	3	23%
anotácie	1	23%
telekomunikácie	6	55%

Tabuľka 3.3: Prehľad rozloženia textov vo výslednom jazykovom modeli.

V počiatočnej fáze tejto práce bolo uvažované o navýšenie počtu akustických dát pre adaptáciu, metódou získavania dát z rádiového vysielania. Postup a nástroje pre túto operáciu boli popísané v článku [10]. Pre charakter tejto práce nebolo vhodné tento postup aplikovať, pretože pri testovacom nahrávaní bolo nahratých okolo 2 000 hodín rádiového vysielania, z ktorého sa podarilo touto metódou získať len 3,3 hodiny reči, čo je zisk 0,1%. Takýto zisk spolu s časovou a priestorovou náročnosťou bol vyhodnotený ako nevyhovujúci pre túto prácu. Preto od tohto spôsobu získavania akustických dát bolo upustené.

3.2 Príprava jazykového modelu

Firmou Phonexia s.r.o. neboli žiadne dáta pre prípravu jazykového modelu poskytnuté, preto ich bolo nutné získať z iných zdrojov. Informačným obsahom akustických dát Sady1, ktorý bol použitý pre trénovanie, bola telekomunikácia a preto bolo nutné zhromaždiť texty k tejto téme. Vybrané texty museli spĺňať charakter spontánnej hovorovej reči. Ako vhodným zdrojom sa ukázali diskusné fóra na internete, v ktorých diskutujúci komunikovali napríklad o typoch mobilných telefónov, o výhodnosti a nevýhodnosti poskytnutých paušáloch, ich tarifách, akciách, ktoré poskytujú jednotliví mobilní operátori alebo o poskytovateľoch internetu. Pri analýze bolo vytypovaných niekoľko takýchto stránok a bol navrhnutý a zhotovený skript pre automatické sťahovanie v jazyku Python využívajúci modul Selenium. Tento modul umožňoval sťahovať „čisté“ texty bez prídavného HTML obsahu. Toto riešenie sa z hľadiská výkonu osvedčilo len pre malé weby, alebo pri sťahovaní len z určitej časti webu. Preto pre sťahovanie textov z veľkých webov bol použitý nástroj *wget*. K týmto dátam boli ešte v tejto téme telekomunikácií pridané texty prepisov Sady1.

Výsledný jazykový model však musí pokrývať i oblasť bežnej ľudskej komunikácie, preto sa pozornosť opäť obrátila na hľadanie vhodných textov. Opäť boli texty získavané z diskusných fór. Aby sa rôznorodosť textov zvýšila, boli pridávané titulky z filmov a blogy. Takýmto spôsobom bolo získané veľké množstvo surových textových dát, ktoré bolo nutné ďalej spracovávať. Vyskytlo sa u nich aj niekoľko problémov. Pri sťahovaní stránok, tieto obsahovali veľké množstvo neslovných znakov, ktoré však bolo možné ľahko odfiltrovať. Väčším problémom boli texty u ktorých bolo pri sťahovaní určené že sú určitého kódovania, avšak pri ich kontrole bolo zistené odlišné kódovanie,čo spôsobilo nečitateľnosť písmen s diakritikou. U týchto textov bolo manuálne zistené kódovanie a boli následne prevedené do jednotného kódovania pre všetky texty, ktorým bolo UTF–8.

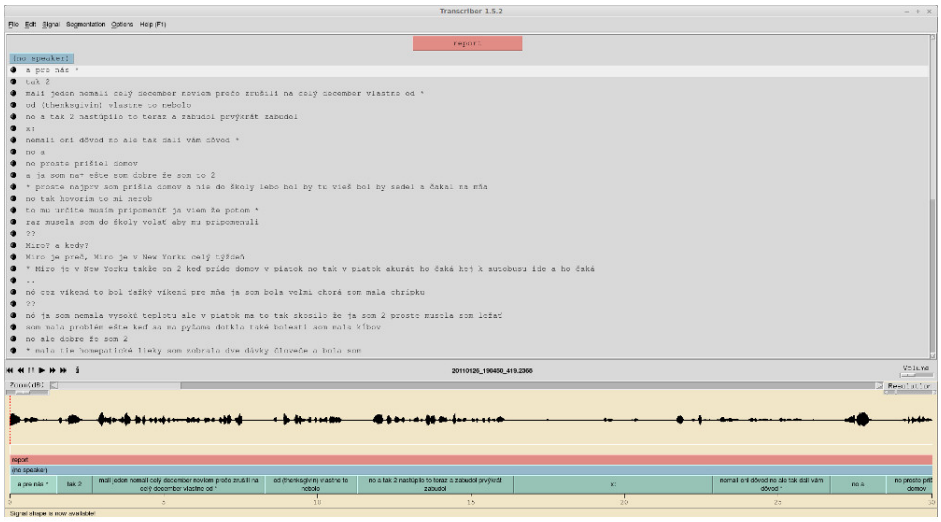
Posledný problém textov bol z formálnej časti spôsobený kultúrou na internete a bohatou slovenskou slovo tvorbou. Myslí sa tým nepoužívanie interpunkčných znamienok, použí-

vanie skrátených, nárečových a regionálnych slov, čím strácajú texty gramatickú celistvosť. Tento problém bol riešený použitím programu *Hunspell*, čím sa slovný priestor pre tvorbu jazykového modelu značne zredukoval. Do výslednej redukcie bolo vstúpené len v prípade textov setu1, u ktorých sa predpokladal vysoký prínos modelu ručným pridávaním odmietnutých slov na základe skúseností s týmto jazykom. Texty boli taktiež zbavené vulgárnych a nevhodných slov, pre zabránenie ich výskytu na výstupe rozpoznávača. Z každého textu bol zostrojený jazykový model, pomocou programu *ngram-count*, ktorý tento text popisuje. Formát týchto modelov bol ARPA. Pre modely bol stanovený odhad v akom pomere by bolo vhodné ich zmiešať. Tento odhad bol korigovaný a následne použitý pre zhotovenie jazykového modelu popisujúceho vybrané texty programom *ngram*. Tabuľka 3.3 ukazuje výsledný pomer zmiešania. Tento jazykový model bol však príliš veľký, pretože obsahoval uni-, bi-, trigramy s početnosťou až niekoľko miliónov, čím by bol z hľadiska výpočtového a výkonnostného prakticky nepoužiteľný. Model bol preto zredukovaný, odstránením všetkých tri gramov a počet bi gramov bol znížený na približne 1.5 milióna.

K takémuto jazykovému modelu bolo taktiež nutné zostaviť výslovnostný slovník. Ako základ poslúžil slovník od firmy Phonexia s.r.o o veľkosti 13 tisíc slov. Ten však neobsahoval všetky slová v jazykovom modeli obsiahnuté. Problém bol vyriešený použitím systému Graphem-to-phoneme (tzv. G2P). Pomocou základného slovníku bol tento systém natrénovaný a zoznam chýbajúcich slov bol následne dogenerovaný a k slovníku pridaný.

3.3 Príprava dát trénovania akustického modelu

Po obdržaní dát Sady1 nebolo nutné s audio súbormi prevádzať úpravy, pretože sa už nachádzali vo formáte z akého bolo možné extrahovať príznaky. Opačná situácia bola u prepisov. Program Transcriber je nástroj pre manuálnu anotáciu rečových signálov. Poskytuje grafické užívateľské rozhranie umožňujúce segmentovanie a popis takýchto segmentov a taktiež pridávanie dodatočných informácií. Príklad užívateľského rozhrania je možné vidieť na obrázku 3.1.



Obr. 3.1: Program Transcriber

Formát v ktorom Transcriber tieto prepisy ukladá vo formáte XML, avšak takýto formát je pre ďalšie spracovávanie nevhodný pretože môže jeden segment uložiť na viac textových

riadkov. Preto bol zvolený čitateľný a ľahšie spracovateľný formát vychádzajúci z formátu *NIST STM* nazvaný *PHX_STM*. Pre prevod bol napísaný skript *trs2stm* v jazyku Python ktorý súbor s transkripciou formátu TRS prevedie na novú podobu. Skript bol testovaný na viac formátoch TRS, pretože formát sa často odvíja od práce anotátora. Text segmentu neobsahoval len samotný prepis rečníka v danom segmente hovoriaceho, ale i pomocné značky predstavujúce sa meniace akustické prostredie tzv. nerečové udalosti (váhanie, nepriamy súhlas, nesúhlas, kašeľ, smiech). Nakoniec boli na prepisy aplikované anotačné a čistiace pravidlá pre dané texty. Tieto výstupy boli prevedené do štruktúry Master label file (tzv. MLF) a zoznamu použiteľných segmentov pre trénovanie rozpoznávača reči. Takto štruktúrovaný súbor je používaný HTK toolkit-om. MLF bol ešte pomocou slovníka kontrolovaný na počet fonémov v ňom sa vyskytujúcich. Dôvod prečo bola táto kontrola prevedená je že trénovacia sada nemusí obsahovať dostatočný počet fonémov. Tu bolo zvolené pravidlo, že ak sa foném v súbore nevyskytuje aspoň 100 krát bude v slovníku premapovaný. Výslovnosti slovník bol získaný rovnakým spôsobom ako bolo popísané v časti 3.2.

U Setu2 nebolo určené akým spôsobom má byť rozdelený na segmenty, preto bola použitá technológia *voice activity detection* (zkr. VAD), pomocou ktorej sme mohli zostaviť odpovedajúci adaptačný list.

Kapitola 4

Popis vytvorenenia systému pre rozpoznávanie reči

Ďalšou úlohou bolo postaviť systém pre rozpoznávanie reči a vykonať jeho adaptáciu. Technologická časť rozpoznávača, nástroje, akustické dáta a ich presné prepisy boli poskytnuté firmou Phonexia s.r.o[1]. Akustický a jazykový model, výslovnostný slovník, ktoré popisujeme v tejto kapitole boli vytvorené v podmienkach tejto firmy a Fakulty informačných technológií v Brne (zkr. FIT).

4.1 Trénovanie akustického modelu

Pre základ trénovania akustického modelu bola použitá trénovacia pipeline LVCSR výskumnej skupiny Speech@FIT, ktorá bola upravená pre firmu Phonexia. Pipeline je pri správnom nastavení automatizovaná a každý svoj krok zaznamenáva. Taktiež využíva Sun Grid Engine (zkr. SGE), čo je systém pre správu dávkového spracovávania úloh. Zložité úlohy je tak možné deliť na sadu menších, ktoré sa počítajú paralelne. Pipeline je prioritne určená pre trénovanie akustického modelu, ale umožňuje plynule prejsť k jeho adaptácii. Trénovanie je rozdelené do etáp (tzv. stage) 1 až 45, kde hlavnými etapami trénovania sú:

- príprava príznakov
- trénovanie mono–fónnych modelov a ich časové zarovnanie
- trénovanie tri–fónových modelov a ich časové zarovnanie
- odhad HLDA a jej aplikácia na príznaky
- trénovanie neurónovej siete pre bottle–neck príznaky
- trénovanie kontextových HMM pomocou maximum likelihood prístupu
- trénovanie kontextových HMM pomocou MPI prístupu

Ak sa vyskytne chyba pri trénovaní, po oprave je možné sa vrátiť do etapy, kde trénovanie skončilo chybou a pokračovať v trénovaní od tohoto bodu. Systém na svojom vstupe pre trénovanie požaduje:

- zoznamy trénovacích a testovacích segmentov

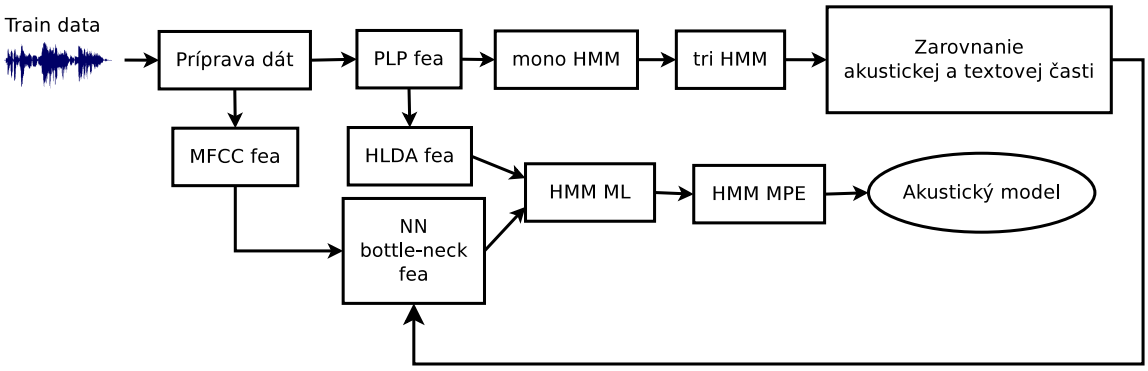
- prepisy vo formáte MLF, viazaný k segmentovaným listom
- jazykový model
- výslovnostný slovník pre slová v trénovacích segmentoch a v jazykovom modeli
- zoznam tzv. questions, čo je skupina fonémov s podobnými akustickými vlastnosťami

Ak bude požadovaná i adaptácia je potrebné predložiť systému aj zoznam segmentov nahrávok pre túto adaptáciu. Na začiatku sú akustické súbory prevedené na príznaky PLP a MFCC, pomocou programu *HCopy* z toolkitu HTK a programu *fextract* z knižnice firmy Phonexia. Tieto príznaky je nutné získať v prostredí firmy Phonexia, pretože tá tieto dáta vlastní a sú klasifikované ako citlivé. Príznaky sú následne presunuté do prostredia FIT VUT. Tu sú umiestnené do pripravenej adresárovej štruktúry. Pomocou príznakov PLP sa trénujú modely HMM pre jednotlivé fonémy. Trénovanie prebieha v štyroch iteráciách pre jednu konfiguráciu počtu GMM, nastáva tu odhad nových parametrov HMM. Následne je pridaný zvolený počet komponentov zmesi gaussoviek modelu a trénovanie pokračuje. Maximálny počet komponentov je 64. Po natrénovaní týchto modelov je prevedené zarovnanie, pre získanie času štartu a konca jednotlivých fonémov v trénovacej sade. Takéto zarovnanie sa anglicky nazýva force–alignment. Toto zarovnanie je následne použité v ďalšom kroku pre trifónové zarovnanie, ktoré prebieha trénovaním HMM pre trifóny. Na začiatku sa prevedie pár iterácií pre odhad, odozvy nových trifónových HMM a následne je použitá technika zhlukovanie pre redukciu počtu modelov trifónov. Trénovanie potom prebieha rovnako ako pri jednoduchých fonémoch v štyroch iteráciách pre jednu konfiguráciu počtu GMM, avšak trénovanie končí, keď je dosiahnutý počet komponentov 24. Až potom je vykonané samotné trifónové časové zarovnanie. Tu si systém prevedie a prepočíta jazykový model do jeho interného formátu. Nad príznakmi PLP je vypočítaná transformácia HLDA. Po nájdení tejto transformácie je táto aplikovaná na všetky trénovacie a testovacie dáta a sú získané nové príznaky – plphlda. Pomocou trifónového časového zarovnania je trénovaná neurónová sieť typu bottleneck. Neurónová sieť je trénovaná pomocou príznakov MFCC získaných pri extrakcii. Po natrénovaní tejto neurónovej siete sú koncové vrstvy odstránené až po bottleneck vrstvu s 30 neurónmi. MFCC koeficienty sú transformované pomocou tejto neurónovej siete na NN_MFCC príznaky. Ako ďalší krok sú vytvorené nové príznaky spojením plphlda a NN_MFCC pod menom plphlda–NN_MFCC. Pomocou týchto združených koeficientov je trénovaný rozpoznávač založený na kontextových HMM pomocou metódy ML. Tento systém je ďalej použitý pre trénovanie kontextových HMM technikou diskriminačného trénovania MPE. Trénovanie pomocou techniky MPE prebiehalo v 15 iteráciách. Po každej iterácii bola vyhodnotená presnosť rozpoznávania reči na základe testovacej sady. Pred začatím tohto trénovania bolo dôležité nastaviť kľúčový parameter I–Smooting, ktorý predstavuje interpoláciu medzi MPE a ML odhadom v závislosti na pokrytí dát Gaussianom. Na konci trénovania bola potom vybraná jedna z týchto iterácii ako víťazná.

4.2 Adaptácia systému

Etapy vykonané pipeline pri priebehu adaptácie:

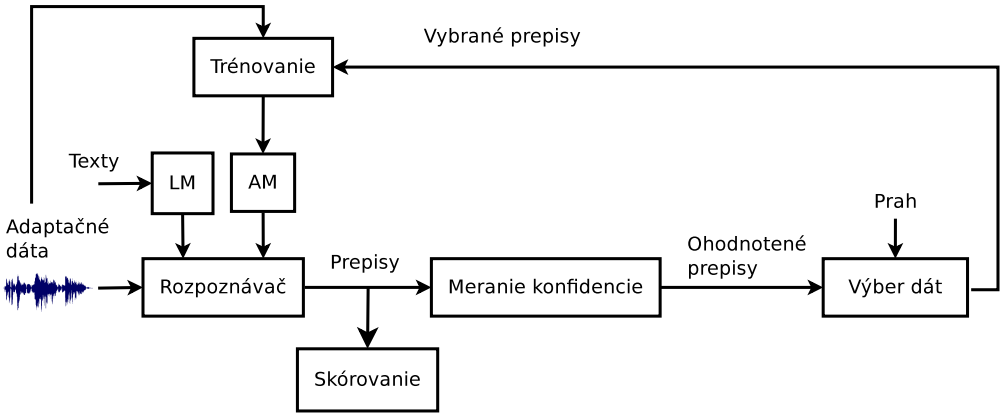
- príprava dát
- prepis adaptačných dát
- odhad konfidencie prepisov



Obr. 4.1: Blokové schéma trénovania akustického modelu

- výber prepisov, ktoré budú použité pre adaptáciu
- spojenie trénovacích a adaptačných dát
- pretrénovanie neurónovej siete pre bottle–neck príznaky
- pretrénovanie kontextových HMM pomocou maximum likelihood prístupu
- pretrénovanie kontextových HMM pomocou MPI prístupu

Bloková schéma takejto adaptácie je na obrázku 4.2. Experimenty s adaptáciou budú predstavené v kapitole 5.



Obr. 4.2: Bloková schéma priebehu adaptácie.

Po vytvorení výsledného rozpoznávača, ktorý bol natrénovaný ako sme uviedli v predchádzajúcej sekcii, bol tento systém použitý pre sadu dát, ktoré chceme použiť na adaptáciu. Tieto dáta boli prepísané týmto systémom do výstupu lattíc. Odhad konfidencie hypotéz začína transformáciou lattice do confusion network, kde je možné zistiť konfidenciu jednotlivého slova v danom časovom úseku v tejto hypotéze. Jednotlivé slová takto hodnotené sú však príliš krátke pre adaptáciu. Získali by sme krátke segmenty a strácal by sa kontext medzi slovami. Preto sa používajú metriky pre spájanie takto ohodnotených slov a prepočítanie konfidencie na celé frázy. Jednou z nich môže byť priechod sieťou od začiatku až do konca tak, že prechádzame najlepšie ohodnotené uzly obsahujúce slová a akumulovať ich skóre konfidencie. Nakoniec podeliť počtom slov, ktorými sme prešli a tým získať

konfidenciu celej frázy. Takáto konfidencia má hodnoty od 0 – 1. Ďalej nasleduje výber segmentov. Tu sa môžeme rozhodovať na základe prahu dvoma spôsobmi. Prah nastavíme tak, že vyberieme všetky segmenty nad zvoleným prahom, čiže nám nezáleží na presnom počte adaptačných dát. A taktiež je možné prah zvoliť tak, že presne percentuálne vyberieme počet adaptačných dát. Tu môže prebiehať ešte dodatočná filtrácia pre krátke segmenty, či už podľa počtu slov, alebo písmen v skúmanom segmente. U vybraných adaptačných segmentov sa prevedie extrakcia základných príznakov (PLP, MFCC), aj zostavenie príznakov z nich vychádzajúcich (plphlda, bottleneck NN, plphlda_NN). Následne sa pridajú k pôvodným trénovacím segmentom. Nasleduje opätovné pretrénovanie bootle–neck neurónovej siete. Po dokončení trénovania, je sieť ešte dotrénovaná bez adaptačných dát, čo má za následok korekciu siete. Postup ktorý potom pokračuje, prebieha ako u klasického trénovania fázami: tvorbou kontextových HMM pomocou maximum likehood a diskriminačným trénovaným kontextových HMM metódou MPI.

Kapitola 5

Experimenty

V tejto kapitole budú preskúmané vlastnosti rozpoznávača trénovaného klasickým spôsobom a následne po prevedení jeho adaptácie. Budeme skúmať presnosť systému na základe veľkosti trénovacej a adaptačnej sady. Taktiež bude skúmaná voľba prahu pre výber adaptačných dát.

5.1 Priebeh experimentov

Na začiatku každého experimentu bolo stanovené rozdelenie dostupných sád dát na trénovaciu a adaptačnú časť. Boli skontrolované a prípadne upravené parametre prostredia trénovania. Samotné trénovanie, testovanie a adaptácia potom prebiehali čiste autonómne. Záznam o priebehu trénovania bol počas behu kontrolovaný, či nenastal problém. Každý experiment tak predstavuje samostatné natrénovanie celého akustického modelu a vyhodnotenie jeho úspešnosti pomocou testovacej sady.

5.2 Systém trénovaný klasickým spôsobom

Tento systém bol trénovaný na Sade1 celej časti, ktorú bolo možné použiť pre trénovanie. Tento systém bol natrénovaný ako prvý pre overenie funkčnosti spôsobu trénovania. Sada pozostávala z viac ako 30 hodín reči a systém tu dosahoval najlepších výsledkov z celého priebehu experimentov.

V tabuľke 5.1 je uvedená úspešnosť vytvoreného systému na testovacích dátach.

train [h]	adapt [h]	WER [%]	D [%]	S [%]	I [%]
30.2	0	35.5	13.5	19.1	2.8

Tabuľka 5.1: Úspešnosť systému s plnou trénovanou sadou, bez adaptácie.

Hodnota 35.5 WER je prijateľnou chybou u systému takéhoto druhu, preto mohlo byť pristúpené k ďalším experimentom.

5.3 Systém trénovaný klasickým spôsobom s adaptáciou

Pri tomto experimente bol prevzatý systém z predchádzajúcej časti 5.2, u ktorého bola prevedená adaptácia pomocou Sady1. Dáta z tejto sady boli vybraté z tých, ktoré neboli

požitá pre trénovanie (145 nahrávok) a celej Sady2. Spolu adaptačná sada obsahovala viac ako 15 hodín reči. Presnosť však stúpla len 0.7%. To môže byť spôsobené tým, že model už dosiahol na testovacích dátach takmer svojho maxima a takýmto typom adaptačných dát ho už nie je možné výrazne vylepšiť, alebo predložená adaptačná sada bola príliš malá. Tabuľka 5.2 ukazuje presné výsledky u tohoto systému.

train [h]	adapt [h]	WER [%]	D [%]	S [%]	I [%]
30.2	15.3	34.8	13.5	19.1	2.8

Tabuľka 5.2: Úspešnosť systému s plnou trénovanou sadou, s adaptáciou.

Keďže nastala takáto situácia, bola pre účely experimentovania s adaptáciou, trénovacia sada zredukovaná približne na 4 hodiny a zvyšné trénovacie dáta boli presunuté do adaptačnej sady.

5.4 Experiment s veľkosťou adaptačnej sady

U systému, kde nastalo obmedzenie trénovanej sady bolo očakávané, že poklesne i úspešnosť prepisu reči, ako u systému s väčšou sadou. Uvedený predpoklad sa potvrdil, „Slabý“ systém vykazoval presnosť 55.13% bez adaptácie, pri 4 hodinách trénovacích dát. Následnou adaptáciou sa jeho úspešnosť posunula až na 55.9%. To bolo dosiahnuté rovnakou adaptačnou sadou ako u systému popísaného v časti 5.3. Táto adaptačná sada bola postupne navyšovaná až na hodnotu 51 hodín, kde hodnoty presnosti vzrástli na hodnotu 58.1%. Ďalšie kroky viedli k navýšeniu trénovacej sady približne o polovicu v ďalších dvoch iteráciách experimentu. Postup bol rovnaký ako u prvej iterácie z 4 hodinami a to navyšovaním adaptačnej sady.

V tabuľke 5.3 je uvedená úspešnosť vytvorených systémov na testovacích dátach.

adapt [h]	WER [%]	D [%]	S [%]	I [%]
15.3	44.1	24.6	17.9	1.3
31.2	43.3	21.1	19.9	2.1
40.8	42.7	20.1	20.2	2.2
51.5	41.9	19.7	19.9	2.2

adapt [h]	WER [%]	D [%]	S [%]	I [%]
15.3	39.5	15.7	20.1	2.5
31.2	39.0	15.5	19.7	2.5
47.7	38.4	15.2	19.5	2.5

adapt [h]	WER [%]	D [%]	S [%]	I [%]
15.3	38.7	15.9	20.2	2.4
31.2	38.2	15.7	19.7	2.4
42.8	37.6	15.5	19.4	2.4

Tabuľka 5.3: Úspešnosť systémov s trénovacou sadou (3.9h, 8.2h, 12.6h) a adaptáciou

V grafe 5.1 vidíme závislosť a reakcie systému na adaptáciu a zmenu jeho presnosti. Všimnime si, že adaptácia prináša vylepšenie presnosti približne 0.5% na 10 hodín adapta-

čných dát. Tieto adaptačné dáta a dáta na ktorých sa trénoval základný systém, pochádzajú z jedného setu dát a sú si podobné, technicky aj tematicky. Z výsledkov preto nie je možné určiť, ako by systém reagoval na nepríbuzné dáta.

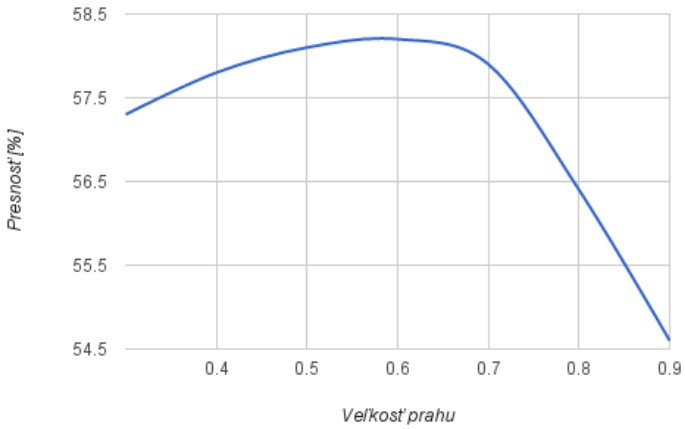
5.5 Experiment s voľbou prahu konfidencie

Adaptácia ako taká je závislá na počte adaptačných dát a ich kvalite. Dáta sú hodnotené pomocou konfidencie. Výber dát pre samotnú adaptáciu prebieha pomocou prahu aplikovaného na hodnoty dôveryhodnosti, ktoré sú v rozsahu 0 až 1 a je zvolený prah pre stanovenie minimálnej dostačujúcej dôveryhodnosti. Experimenty v tejto časti používajú systém s 4 hodinami trénovacích dát a 51 hodinami adaptácie.

Prah	Počet vybraných segmentov	WER [%]	D [%]	S [%]	I [%]
0.3	97062	42.7	21.0	19.7	1.9
0.4	91626	42.2	20.2	20.0	2.2
0.5	83534	41.9	19.7	19.9	2.2
0.6	72603	41.8	19.7	19.8	2.2
0.7	58897	42.1	19.9	20.0	2.0
0.8	43010	43.6	23.7	18.1	1.4
0.9	25930	45.4	21.9	21.2	2.1

Tabuľka 5.4: Úspešnosť systémov pri rôznych prahoch konfidencie

Postupne bol menený prah pre dôveryhodnosť a výsledky je možné vidieť v tabuľke 5.4. Je vidieť, že pre použitú adaptačnú sadu sa optimálny prah pohyboval okolo 0.5 – 0.6, kde systém dosahoval najlepších výsledkov. Dopad zvoleného prahu je taktiež znázornený v grafe 5.2. Pri hodnote 0.9 je vidieť rapídny pokles WER oproti iným hodnotám, čo je pravdepodobne spôsobené príliš veľkou redukciou adaptačných segmentov.



Obr. 5.2: Závislosť voľby prahu pre výber adaptačných dát na presnosť.

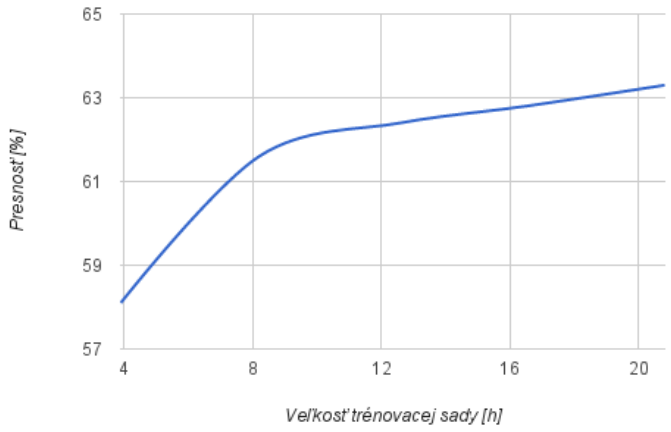
5.6 Experiment s veľkosťou trénovacej a adaptačnej sady

Experimenty v tejto sekcii boli navrhnuté a prevedené za účelom zistenia vplyvu kombinácie rôznych veľkostí trénovacej a adaptačnej sady. Boli použité všetky dostupné dáta, ktoré boli rozložené medzi trénovaciu a adaptačnú sadu. V tabuľke 5.5 je možné vidieť hodnoty skúmaných parametrov, dosiahnuté výsledky pred a po adaptácii. Pri 20 hodinách trénovacích dát a 34 hodinách adaptácie tento systém stráca na systém v časti 5.2 1.9% WER.

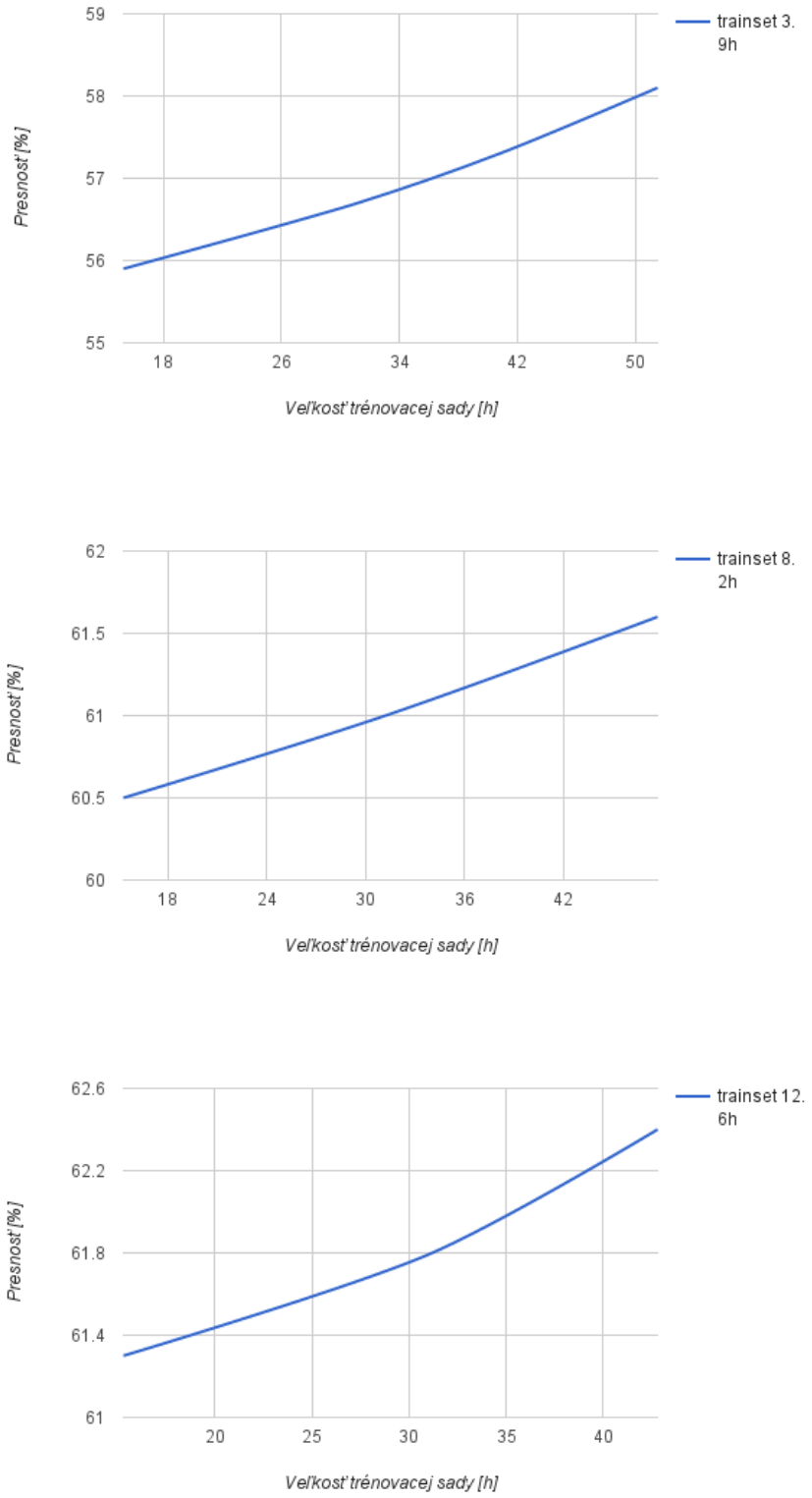
train [h]	adapt [h]	WER [%]	D [%]	S [%]	I [%]
3.9	51.5	45.4 / 41.9	21.9 / 19.9	21.2 / 19.8	2.1 / 2.0
8.2	47.2	40.3 / 38.4	17.3 / 15.7	20.2 / 20.1	2.6 / 2.5
12.6	42.8	38.6 / 37.6	15.8 / 15.6	20.1 / 20.2	2.6 / 2.5
16.5	38.9	37.9 / 37.2	15.4 / 15.2	19.7 / 19.5	2.7 / 2.5
20.8	34.6	37.0 / 36.7	14.4 / 14.6	19.2 / 19.1	2.8 / 2.7

Tabuľka 5.5: Úspešnosť systémov s rôznymi trénovacími a adaptačnými sadami

Grafickú reprezentáciu tabuľky 5.5 môžeme vidieť v grafe 5.3. Medzi 4 a 8 hodinami trénovacích dát je výrazný nárast presnosti oproti ďalšiemu priebehu trénovania. Zväčšovaním trénovacej sady doplnenej o adaptaciu sa presnosť systému sa lineárne zvýšila.



Obr. 5.3: Úspešnosť rozpoznávania pri trénovacích sadách doplnených o adaptáciu.



Obr. 5.1: Závislosť veľkosti adaptačnej sady na presnosť.

Kapitola 6

Záver

V tejto diplomovej práci som sa zaoberal popisom a vytvorením rozpoznávača reči. Začiatok práce predstavoval úvod do štruktúry rozpoznávačov reči a zvyšok práce pojednával o príprave dát pre akustický a jazykový model, popise tvorby samotného rozpoznávača a jeho adaptácie. Dáta pre akustický model poskytla firma Phonexia s.r.o. Dáta pre jazykový model bolo nutné získať z internetu. Pomocou týchto modelov bol zostavený celý rozpoznávač reči. Ďalšou kľúčovou časťou práce bola adaptácia rozpoznávača typu unsupervised. Kroky viedli k stanoveniu, prevedeniu a vyhodnoteniu experimentov s vhodnou metrikou merania dôvery k výstupu rozpoznávača. Experimenty mali charakter výberu trénovacej a adaptačnej sady dát. Vybrané prepisy boli následne použité ako anotácia k akustickým dátam a systém bol znovu natrénovaný s týmito novými dátami a bola vyhodnotená jeho úspešnosť.

Prvý experiment bol zameraný na trénovanie rozpoznávača klasickým spôsobom. Boli tu použité všetky dostupné trénovacie dáta a úspešnosť systému bola 35.5% WER na testovacích dátach. Systém trénovaný klasickým spôsobom s adaptáciou bol popísaný v nasledujúcom experimente, kde úspešnosť tohoto systému stúpla na hodnotu 34.8% WER. Ďalší experiment pojednával o voľbe veľkosti adaptačnej sady. Bolo zistené, že pri náraste adaptačnej sady až k hodnote 51 hodín jej presnosť bola 41.9 WER pre systém trénovaný na 4 hodinách reči. Experiment s voľbou prahu dôveryhodnosti ukázal, že jeho najvhodnejšia hodnota je medzi 0.5 až 0.6. Posledný experiment zameraný na rôzne kombinácie veľkostí trénovacej a adaptačnej sady ukázal, že systém trénovaný na 20 hodinách reči a na 34 hodinách pre adaptáciu, stráca na systém trénovaný klasickým spôsobom 1.9 WER presnosti. Táto konfigurácia preto bola najlepšia zo všetkých skúmaných.

Na záver je preto možné konštatovať, že adaptácia rozpoznávača reči ma prínos k navýšeniu presnosti rozpoznávania a to i pri malej trénovacej sade. Nezanedbateľný je aj ekonomický prínos, pretože nie je nutné vynakladať finančné prostriedky na získanie rozsiahlych dátových sád. Je ho možné použiť i v tých prípadoch, kedy potrebujeme systém rýchle adaptovať na novú doménu alebo nemáme dostatok trénovacích dát.

Literatúra

[1] Phonexia. 2006.
URL http://phonexia.com/

[2] Bishop, C. M.: *Pattern Recognition and Machine Learning.* Springer, 2006, ISBN 978-0387-31073-2.

[3] Burget, L.: *Complementarity of speech recognition systems and system combination.* Dizertační práce, Brno University of Technology Faculty, Brno, 2004.
URL http://www.fit.vutbr.cz/~burget/phd_activities/burget_thesis.pdf

[4] Davis, S.; Mermelstein, P.: Comparison of parametric representations formonosyllabicword recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing 28 (4)*, 1980: s. 357–366.

[5] Grézl, F.; Karafiát, M.; Kontár, S.; aj.: Probabilistic and bottle-neck features for LVCSR of meetings. In *ICASSP 2007*, Honolulu, Hawaii, USA, 2007, s. 757–760.

[6] Hermansky, H.: Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America 87 (4)*, 1990: s. 1738–1752.

[7] Huang, X.; Acero, A.; Hon, H.-W.: *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Prentice Hall, 2001.

[8] Karafiát, M.: *Study of Linear Transformations Applied to Training of Cross-Domain Adapted Large Vocabulary Continuous Speech Recognition Systems.* Dizertační práce, Brno University of Technology Faculty, 2008.

[9] Mangu, L.; Brill, E.; Stolcke, A.: Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language 14 (4)*, 2000: s. 373–400.

[10] Oldřich, P.; Valiantsina, H.; Lukáš, B.; aj.: Acquisition of Telephone Data from Radio Broadcasts with Applications to Language Recognition. 2008.

[11] Plahl, C.: *Neural Network Based Feature Extraction for Noisy Speech.* Dizertační práce, 2014.

[12] Psutka, J.; Müller, L.; Matoušek, J.; aj.: *Mluvíme s počítačem česky.* Academia, 2006, ISBN 80-200-1309-1.

[13] Seigel, M. S.: *Confidence Estimation for Automatic Speech Recognition Hypotheses.* Dizertační práce, University of Cambridge, 2013.

Dodatok A

Obsah CD

Priložené CD obsahuje:

- skript pre prevod TRS na PHX_STM (použitý pre prípravu akustických dát).
- skript pre sťahovanie textu z internetu. (súčasť jazykového modelu)
- PDF a LaTeXverzia tejto práce.