ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA KYBERNETIKY

BAKALÁŘSKÁ PRÁCE

EVALUACE TRÉNOVACÍCH DAT K TVORBĚ SPORTOVNÍCH AKUSTICKÝCH MODELŮ

Plzeň, 2016 Jan Hás

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne .....

.....

podpis

Abstrakt

Tato bakalářská práce se zabývá trénováním sportovních akustických modelů. První polovina práce je teoreticky zaměřena na statistické rozpoznávání řeči, tvorbu anotací a slovníků. Dále se práce zabývá analýzou chyb v anotacích, možnostmi jejich detekce a opravy. Druhá polovina práce popisuje praktický postup při trénování sportovních akustických modelů pomocí nástroje HTK. V závěru práce jsou porovnávány dva akustické modely, první vznikl natrénováním z neopravených dat a druhý akustickým model vznikl z dat opravených.

Klíčová slova: akustický model, automatické rozpoznávání řeči, Transcriber, HMM, HTK

Abstract

This bachelor's thesis is focused on training of sports accoustic models. The first half of this thesis is theoretically concerned with statistical speech recognition, creating annotations and dictionaries. Further, this thesis deals with annotation errors analysis, ways of detecting and correcting these errors. The second half of this thesis describes practical procedure of sports accoustic models training using HTK toolkit. In the end of the thesis there are compared two different trained accoustic models where the first accoustic model was trained with original, unprocessed data and the second one with data corrected using language dictionaries.

Keywords: accoustic model, automatic speech recognition, Transcriber, HMM, HTK

Poděkování

Rád bych tímto poděkoval svému vedoucímu práce, Ing. Mgr. Josefu Psutkovi, Ph. D., za trpělivost, podporu a ochotu při vypracovávání této práce.

Obsah

Kapitola 1

Úvod

1.1 Motivace a cíl práce

Mezi hlavní cíle této bakalářské práce patří analýza nejčastěji vyskytujících se chyb v anotačních textech sportovních přenosů, detekce a oprava těchto chyb za pomoci slovníků obsahujících gramaticky správné tvary slov a jejich správné výslovnosti. Následně budou analyzována řečová data a připraveny dvě sady dat pro trénování sportovních akustických modelů, což představuje další z cílů práce. První sada dat bude reprezentována originálními, neupravenými přepisy sportovních utkání. Druhá sada dat bude naopak vytvořena z anotací, které budou před procesem trénování opraveny za pomocí již připravených slovníků. S takto připravenými daty budou za pomocí nástroje HTK natrénovány akustické modely pro rozpoznávání řeči. Natrénované akustické modely budou následně upraveny s cílem zvýšit procentuální úspěšnost samotného rozpoznávání. Za pomoci testovacích dat pak budou experimentálně porovnány obě sady akustických modelů a ověřena hypotéza, že modely založené na opravených anotačních datech budou poskytovat procentuálně vyšší úspěšnost rozpoznávání, než modely natrénované z původních dat.

1.2 Struktura práce

Pro lepší orientaci v textu této práce je tento odstavec věnován základní struktuře dokumentu a stručnému obsahu jednotlivých kapitol.

Druhá kapitola práce je věnována základním poznatkům a teorii statistického rozpoznávání řeči. Jsou zde shrnuty a popsány základní úlohy statistického rozpoznávání řeči, teorie skrytých Markovových modelů (HMM), parametrizace, jazykového modelování a dekódování.

Třetí kapitola je zaměřena na popis a použití nástrojů pro anotaci řečových záznamů a tvorbu slovníků pro opravy anotačních textů.

Ve čtvrté kapitole jsou podrobněji popsány možné chyby, které se často vyskytují v anotačních textech. Je zde uvedeno několik typů anotačních chyb, možnosti jejich detekce a automatické opravy pomocí slovníků oprav.

V páté kapitole je popsáno předzpracování dat pro trénování akustických modelů v nástroji HTK.

Šestá kapitola je celá zaměřena na postup a proces trénování sportovních akustických modelů pomocí nástroje HTK. Jsou zde podrobněji popsány jednotlivé dílčí procesy trénování, od parametrizace řečových dat po samotný proces trénování akustických modelů a rozpoznávání řeči.

Předposlední sedmá kapitola shrnuje analýzu a přípravu dat určených pro trénování akustických modelů a výsledky rozpoznávání s natrénovanými sportovními akustickými modely. Je zde také porovnání úspěšnosti rozpoznávání s jednotlivými modely.

Naposled osmá kapitola představuje závěr celé této práce včetně stručného shrnutí dosažených výsledků.

Kapitola 2

Statistické rozpoznávání řeči

Předtím, než bude detailně popsán postup trénování sportovních akustických modelů, je v této kapitole přiblížena problematika rozpoznávání řeči. Teorie rozpoznávání řeči je poměrně komplexní, a proto budou ve stručnosti popsány jen ty nejdůležitější části relevantní k tématu práce. Většina zmíněné teorie byla čerpána z [2].

2.1 Statistický přístup k rozpoznávání řeči

Již přes padesát let je snahou výzkumných laboratoří navrhnout a zkonstruovat stroj schopný rozpoznávat v akustickém signálu jednotlivé fonémy a automaticky rekonstruovat co řečník vyslovil. V oblasti zpracování řečového signálu a samotné klasifikace byl již od samotných počátků do současnosti učiněn nesmírný pokrok. Přesto se nedaří sestrojit zařízení, které by umožňovalo rozpoznat souvislou promluvu libovolného řečníka užívajícího libovolná slova určitého jazyka. Ztěžujícími okolnostmi rozpoznávání řeči jsou variabilita řečníka, hluk prostředí, ve kterém je řeč pronášena, ale i samotná složitost konstrukce akustického rozpoznávání mluvené řeči, tj. například rozsáhlost slovníku s rozpoznanými slovy.

Rozdílnost promluv jednotlivých řečníků je způsobena odlišnými parametry hlasového ústrojí (tj. tvar dutiny hrdelní, ústní, nosní, různé frekvence kmitání hlasivek apod.) a různým způsobem artikulace. Z hlediska řečníka se systémy rozpoznávání řeči dělí následovně:

- **Systémy na řečníku závislé** – trénování akustického modelu je přímo závislé na hlasu právě jednoho konkrétního řečníka.
- **Systémy na řečníku nezávislé** – trénování akustického modelu je závislé na hlasech desítek, stovek až tisíců různých řečníků.

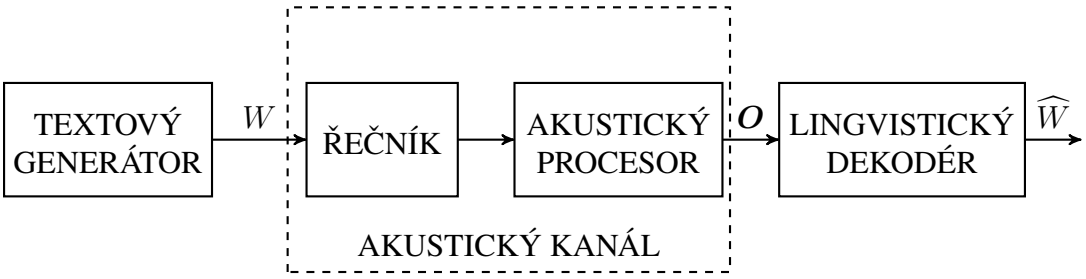
Hlas jednoho řečníka je odlišný v závislosti na hlasitosti promluvy, nachlazení či momentální náladě. Prakticky je nemožné, aby jedna osoba vyslovila ve dvou různých situacích stejné slovo naprosto stejným způsobem, což je povětšinou způsobeno i jen mírnou změnou nastavení hlasového ústrojí. Z hlediska aplikovaných metod rozpoznávání lze klasifi-

kátory řeči dělit na ty, které pracují na principu porovnávání se vzory *(template matching)* a na klasifikátory pracující s využitím statistických metod rozpoznávání.

První skupina metod byla na svém vrcholu v sedmdesátých a osmdesátých letech minulého století, kdy probíhala aplikace obzvlášť v klasifikátorech izolovaně vyslovených slov. Slovo se zpracovává jako celek, přičemž je klasifikováno do té třídy, k jejímuž vzorovému obrazu má nejmenší vzdálenost.

Ve druhé skupině je přístup ke klasifikaci založen na statistických metodách, ve kterých jsou slova a celé promluvy modelovány pomocí tzv. skrytých Markovových modelů, označovaných též HMM (z anglického *Hidden Markov Models*). Dílčí slova jsou modelována buď jako celek jedním skrytým Markovovým modelem slova nebo častěji subslovními jednotkami.[1] Promluva je modelována zřetězením těchto subslovních modelů.

Hlavní schéma statistického přístupu k rozpoznávání mluvené řeči se skládá z akustického procesoru a lingvistického dekodéru, jak je zobrazeno na obrázku 2.1. Akustický kanál spojuje řečníka s akustickým procesorem. Funkce akustického procesoru spočívá v transformaci řečových kmitů pronesených řečníkem na posloupnosti vektorů příznaků. Funkce lingvistického dekodéru spočívá v transformaci řetězců příznaků na řetězce slov. Rozpoznávání je zde formulován jako problém dekódování s maximální aposteriorní pravděpodobností (založena na smyslové zkušenosti).



Obr. 2.1: Schéma systému rozpoznávání řeči založené na statistickém přístupu.

Za předpokladu, že $W = \{w_1, w_2, \ldots, w_N\}$ značí posloupnost N slov a nechť $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ je akustická informace, tj. posloupnost vektorů příznaků, odvozená z řečového signálu, v níž se lingvistický dekodér snaží rozpoznat, jaká slova byla vyslovena. Záměrem je nalézt posloupnost slov $\widehat{W}$, která maximalizuje podmíněnou pravděpodobnost $P(W|\boldsymbol{O})$, tj. nejpravděpodobnější posloupnost slov pro danou akustickou informaci $\boldsymbol{O}$. Využitím Bayesova pravidla pak dostaneme následující výraz:

$$\widehat{W} = \arg\max_W P(W|\boldsymbol{O}) = \arg\max_W \frac{P(W)P(\boldsymbol{O}|W)}{P(\boldsymbol{O})} \tag{2.1}$$

Pravděpodobnost $P(\boldsymbol{O}|W)$ znamená, že při vyslovení posloupnosti slov W bude produkována posloupnost výstupních vektorů příznaků $\boldsymbol{O}$. $P(W)$ je apriorní pravděpodobnost

[1] Jako subslovní jednotky jsou považovány například slabika, foném, trifón apod.

posloupnosti slov W, tj. pravděpodobnost, že řečník bude číst posloupnost slov W. $P(\boldsymbol{O})$ je apriorní pravděpodobnost posloupnosti výstupních vektorů a jelikož není funkcí W , lze při hledání maxima tuto apriorní pravděpodobnost $P(\boldsymbol{O})$ ignorovat a výraz (2.1) lze upravit na tvar:

$$\widehat{W} = \arg\max_{W} P(W, \boldsymbol{O}) = \arg\max_{W} P(W)P(\boldsymbol{O}|W) \tag{2.2}$$

Z rovnice (2.2) je patrné, že problém stanovení nejlepší posloupnosti slov k danému akustickému signálu lze řešit pomocí dvou oddělených pravděpodobností $P(\boldsymbol{O}|W)$ a $P(W)$. Tyto pravděpodobnosti mohou být modelovány a trénovány nezávisle na sobě. Podmíněná pravděpodobnost $P(\boldsymbol{O}|W)$ nese informaci o akustickém modelu a apriorní pravděpodobnost $P(W)$ nese informaci o jazykovém modelu. Tyto dvě informace je nutno znát již před samotným rozpoznáváním, a to trénováním z řečových a jazykových dat.

Proces rozpoznávání spočívá v nalezení posloupnosti slov $\widehat{W}$, která pro danou posloupnost vektorů příznaků $\boldsymbol{O}$ maximalizuje součin pravděpodobností $P(W)$ a $P(\boldsymbol{O}|W)$ přes všechny možné posloupnosti slov W. Výpočetní náročnost je však přespříliš veliká i pro méně rozsáhlé slovníky, a proto se využívají různé prohledávací a rozhodovací strategie, které se snaží redukovat počet operací s minimálním zkreslením přesnosti rozpoznávání.

Úlohu statistického rozpoznávání mluvené řeči lze dekomponovat do několika základních úloh. Ty jsou následující:

1. Akustickou analýzu řečového signálu s určením vektorů příznaků $\boldsymbol{O}$.
2. Tvorbu akustického modelu pro určení pravděpodobnosti $P(\boldsymbol{O}|W)$.
3. Tvorbu jazykového modelu pro určení pravděpodobnosti $P(W)$.
4. Nalezení nejpravděpodobnějších posloupností slov použitím účinných prohledávacích strategií.

2.2 Analýza akustického signálu

První úlohou při procesu tvorby systému rozpoznávání řeči je analýza akustického signálu pro jeho další zpracování. Cílem akustické analýzy je metodami zpracování signálu získat z akustického signálu posloupnost vektorů příznaků $\boldsymbol{O}$.

Metodou kepstrální analýzy lze oddělit signály vzniklé konvolucí či součinem jednotlivých složek daného signálu. Řečové kmity představují konvoluci impulsní odezvy a budící funkce hlasového ústrojí. Touto analýzou získáme tzv. kepstrum, pomocí něhož je možné například detekovat znělé a neznělé zvuky.

Další metodou zpracování signálu je lineární prediktivní analýza (LPC). Tato analýza provádí estimaci parametrů modelu řečové produkce na krátkodobém základu. Výsledkem jsou kepstrální koeficienty LPC.

Pro simulaci subjektivního vnímání výšky zvuků člověkem se zavádí tzv. mely. Na respektování zmíněného subjektivního vnímání výšky tónu a kritických pásem slyšení jsou zavedeny tzv. Melovské kepstrální koeficienty (MFCC). Tyto koeficienty MFCC využívají banky nelineárně rozmístěných pásmových filtrů, přičemž rozmístění jednotlivých filtrů ovlivňuje subjektivní vnímání výšky tónu.

Poslední metodou je perceptivní lineární prediktivní analýza (PLP). Ta je založena na respektování kritických pásem slyšení, nelineárního vnímání hlasitosti a nelineárním vztahem mezi intenzitou a hlasitostí. Zpracování akustického signálu je podrobně rozebráno v [2].

2.3 Akustické modelování

Jelikož jedním z cílů této práce je tvorba sportovních akustických modelů, je této sekci o akustickém modelování věnována větší pozornost. Hlavním cílem akustického modelování je určit co možná nejpřesnější odhad podmíněné pravděpodobnosti $P(\boldsymbol{O}|W)$ pro libovolnou posloupnost vektorů příznaků $\boldsymbol{O}$ a libovolnou posloupnost slov W. Akustické modely by měly být přesné, flexibilní a účinné.

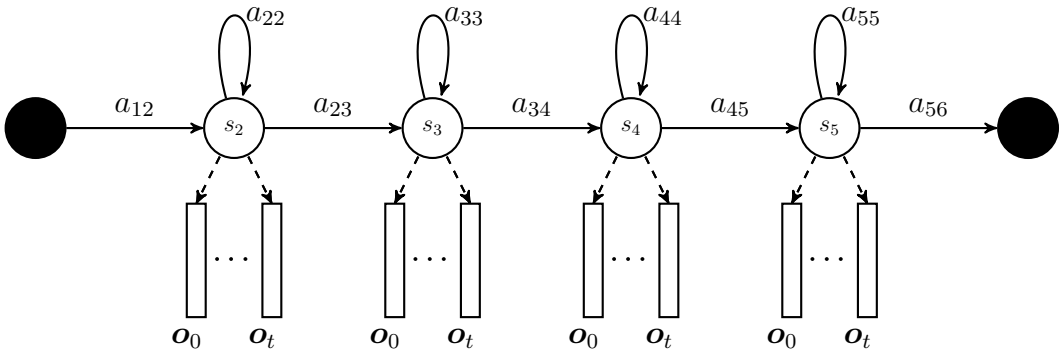
Flexibilita je nutná z důvodu, že podmínky, za kterých je rozpoznávač řeči provozován, jsou často zcela rozdílné od podmínek trénování. Tím jsou myšleny rozdílné hlasy řečníků, jiný způsob artikulace, odlišné tempo řeči, rozdílné vlastnosti akustického kanálu či jiné akustické pozadí atp. Přesnost je důležitá kvůli požadavku odlišit foneticky podobná slova s lingvisticky odlišnými významy. Účinnost je potřebná při nasazení systému rozpoznávání v reálných aplikacích, kdy odezva klasifikátoru musí být dostupná v reálném čase. Účinným způsobem řešení této náročné úlohy je využití skrytých Markovových modelů.

2.3.1 Struktura a parametry HMM

Metody modelování řeči Markovovými modely vychází z představy o vytváření řeči. Při generování promluvy řečníkem si lze představit, že hlasové ústrojí je během krátkého okamžiku v jednom z konečných počtů stavů konfigurace hlasového ústrojí. Ústrojí generuje signál, který je popsán spektrálními charakteristikami, které jsou reprezentovány vhodnými vektory příznaků $\boldsymbol{O}$. Vytváření řeči vychází i ze samotné konstrukce klasifikátoru, která je založená na modelování řečového signálu pomocí Markovova procesu. V tomto

procesu dochází ke generování dvou vzájemně svázaných časových posloupností náhodných proměnných, a to posloupností konečného počtu stavů zvaný podpůrný Markovův řetězec a spektrální charakter mikrosegmentů řečového signálu reprezentovaný řetězcem vektorů příznaků $\boldsymbol{O}$. Pro tyto spektrální charakteristiky existují funkce, které mají v kompetenci pravděpodobnostně ohodnocovat vztah charakteristik ke všem stavům.

Pro nalezení co nejlepšího odhadu rozdělení pravděpodobnosti $P(\boldsymbol{O}|W)$ je nutné určit topologii HMM, typu a hodnot jeho parametrů. Tyto neznámé údaje lze zjistit dvěma způsoby. Využitím expertního odhadu na základě apriorních znalostí nebo statistickou indukcí z množiny trénovacích dat. Bude-li využito obou znalostí, statistická indukce poslouží pro odhad parametrů HMM a apriorní znalost bude aplikována k vybrání vhodné struktury HMM a zvolení vhodného typu parametrů.



Obr. 2.2: Ukázka Markovova modelu slova se čtyřmi stavy, které odpovídají počtu hlásek daného slova.

2.3.2 Pravděpodobnosti přechodu a výstupní pravděpodobnosti

Podmíněné pravděpodobnosti přechodu a_{ij} určují s jakou pravděpodobností přechází model ze stavu s_i, v jakémkoliv čase t do stavu s_j v čase $t+1$. Pro podmíněnou pravděpodobnost platí následující vztah:

$$a_{ij} = P\big(s(t+1) = s_j | s(t) = s_i\big) \tag{2.3}$$

Stav modelu v čase t je značen $s(t)$. Zároveň se předpokládá, že pro všechny stavy s_i je splněna následující podmínka:

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.4}$$

Rozdělení funkce výstupní pravděpodobnosti $b_j(o_t)$ popisují rozdělení pravděpodobnosti pozorování o_t produkovaného stavem s_j v čase t. V případě nabývání konečného počtu diskrétních hodnot mají funkce $b_j(o_t)$ význam pravděpodobnosti a v případě pozorování

hodnoty spojité náhodné veličiny je daná funkce $b_j(o_t)$ hustotou pravděpodobnosti jevu, že stav s_j v čase t generuje pozorování o_t. Pro funkci $b_j(o_t)$ platí:

$$b_j(o_t) = P(o_t|s(t) = s_j) \tag{2.5}$$

Pro řádnou variabilitu a robustnost řečového signálu je požadováno specifické výstupní rozdělení pravděpodobnosti, aby se od sebe separovaly různé zvuky. V současné době dochází k využívání spojitého rozdělení se směsí Gaussovských hustotních funkcí. Spojité rozdělení se směsí normálních hustotních funkcí $\mathcal{N}(\cdot)$ je rozdělení, kde tvar výstupní hustoty pravděpodobnosti je tvořen váženým součtem jednotlivých normálních hustot pravděpodobností, kde každá z nich je určena svým vektorem středních hodnot $\boldsymbol{\mu}_{jrm}$ a kovarianční maticí $\boldsymbol{C}_{jrm}$. Z důvodů zvýšení výpočetní rychlosti a robustnosti se často počet parametrů výstupní hustoty pravděpodobnosti redukuje tak, že kovarianční matice $\boldsymbol{C}_{jrm}$ je uvažována pouze diagonální. Parametry rozdělení výstupní hustoty pravděpodobnosti neobsahují pouze střední hodnoty a diagonální kovarianční matice, ale také váhy jednotlivých složek c_{jrm}. Tvar hustotní funkce lze popsat potom následujícím vztahem:

$$b_j(\boldsymbol{o}_t) = \prod_{r=1}^{R} \left[\sum_{m=1}^{M_r} c_{jmr} \mathcal{N}\left(\boldsymbol{o}_{rt}; \boldsymbol{\mu}_{jrm}, \boldsymbol{C}_{jrm}\right) \right]^{g_r}, \tag{2.6}$$

kde R značí počet datových proudů, M_r počet složek hustotní směsi v r-tém datovém proudu daného vektoru pozorování $\boldsymbol{o}_{rt}$, c_{jrm} vyjadřuje váhu m-té složky r-tého datového proudu j-tého stavu a $\mathcal{N}\left(\boldsymbol{o}_{rt}; \boldsymbol{\mu}_{jrm}, \boldsymbol{C}_{jrm}\right)$ je multidimenzionální normální rozdělení se střední hodnotou $\boldsymbol{\mu}_{jrm}$ a kovarianční maticí $\boldsymbol{C}_{jrm}$.

$$\mathcal{N}(\boldsymbol{o}_{rt}; \boldsymbol{\mu}_{jrm}, \boldsymbol{C}_{jrm}) = \frac{1}{\sqrt{(2\pi)^{n_r}|\boldsymbol{C}_{jmr}|}} e^{\left(-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jrm})^{\mathrm{T}} \boldsymbol{C}_{jrm}^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_{jrm})\right)} \tag{2.7}$$

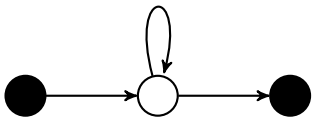
Kde n_r je dimenze vektoru pozorování $\boldsymbol{o}_{rt}$ v datovém proudu r.

2.3.3 Struktura skrytých Markovových modelů

Pro modelování mluvené řeči je využíváno levo-pravých Markovových modelů (*letf-to-right models*), které vyhovují pro modelování procesů, u nichž je vývoj spojen s postupujícím časem. Elementární vlastností uvedených modelů je, že proces začíná příchodem prvního spektrálního vzoru z počátečního stavu modelu. S rostoucím časem dochází k přechodům ze stavů s nižšími indexy do stavů s vyššími indexy či setrvání ve stejném stavu. Z toho je patrné pojmenování modelu jako levo-pravý. Konec procesu nastává s příchodem posledního spektrálního vzoru, kdy se model nachází v koncovém stavu.

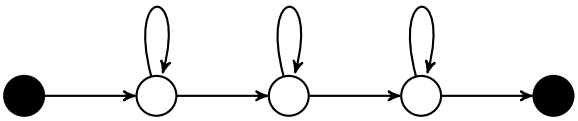
2.3.4 Skryté Markovovy modely fonémů

Pro zjednodušení a urychlení procesu trénování parametrů HMM slov je vhodné odvodit proces do menších řečových jednotek řeči, než-li jsou slova a to například do fonémů. Níže je uvedeno několik příkladů různých druhů struktur fonémů. Na obrázku 2.3 je zobrazena struktura třístavového modelu. První a třetí stav daného modelu jsou neemitující stavy, které negenerují žádná pozorování a nemají žádné příslušné rozdělení výstupní pravděpodobnosti. Druhý stav je příznačný emitující stav.



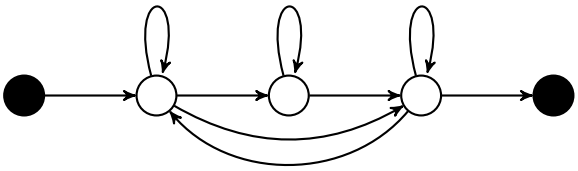
Obr. 2.3: Model fonému s jedním emitujícím stavem.

Struktura se třemi emitujícími stavy je využívána při realizaci této práce v nástroji HTK. První a pátý stav je tzv. fiktivní a slouží k přechodu mezi modely. Druhý, třetí a čtvrtý stav jsou jádrem daného fonému. Model s pěti stavy a třemi emitujícími stavy je vyobrazen na obrázku 2.4.

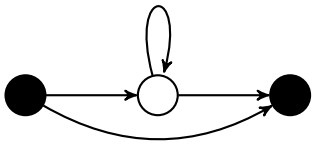


Obr. 2.4: Model fonému se třemi emitujícími stavy.

Model pauzy má naprosto odlišnou strukturu. Rozlišujeme dva druhy pauz. Dlouhou pauzu *_sil_* (z anglického *silence*) je modelována strukturou 2.5, která je využívána pro modelování pauzy na začátku a konci promluvy. Krátké pauzy mezi slovy mají model struktury, který je na obrázku 2.6.



Obr. 2.5: Model pauzy na začátku a konci slova (`_sil_`).



Obr. 2.6: Model krátké pauzy mezi slovy (_sp_).

2.3.5 Trénování parametrů skrytého Markovova modelu

Volba topologie skrytého Markovova modelu je úlohou především expertního návrhu, kdežto určení hodnot parametrů modelu je realizováno statistickou indukcí neboli estimací parametrů na základě přesně anotovaných trénovacích akustických dat. Plán spočívá ve stanovení hodnoty parametrů modelů. Nejčastěji je užíváno pro určení odhadu metoda maximální věrohodnosti (*Maximum Likelihood*).

Pro modelování řeči pomocí HMM existuje velmi efektivní metoda pro trénování parametrů modelu, která má jádro v kritériu maximální věrohodnosti. Používá se pro maximalizaci věrohodnostní funkce iterativní procedura nazývaná *Baumův-Welchův algoritmus*, který je zvláštním případem algoritmu *Expectation-Maximalization*. Existují dva způsoby trénování, a to trénování izolovaných jednotek promluvy a trénování vložených jednotek promluvy [2].

2.4 Jazykové modelování

Cílem jazykového modelování je určit pravděpodobnost $P(W)$ pro libovolnou posloupnost slov W, přičemž tato informace bude následně využita při dekódování. Ovšem určit pravděpodobnost $P(\boldsymbol{O}|W)$ všech možných posloupností slov W libovolné délky K je nemožné, výpočet podmíněné pravděpodobnosti se aproximuje použitím tzv. n-gramového jazykového modelu. Pro takový model je pak pravděpodobnost daného slova aproximována v závislosti na $n-1$ předcházejících slovech. Takto můžeme vytvořit takřka jakýkoliv n-gramový model v závislosti na tom, s jak rozsáhlou historií slov chceme pracovat. Mezi základní n-gramové jazykové modely se však považují:

1) **zerogramový jazykový model** – všechna slova mají stejnou pravděpodobnost výskytu $P(w_k) = \frac{1}{N}$, kde N je počet slov ve slovníku

2) **unigramový jazykový model** – pravděpodobnosti jsou závislé na výskytu daného slova, avšak nikoliv na historii předchozích slov, $P(w_k) = \frac{c(w_i)}{N}$, kde $c(w_k)$ je počet výskytů slova w_k, N počet slov ve slovníku

3) **bigramový jazykový model** – pravděpodobnosti jsou aproximovány na základě jednoho bezprostředně předcházejícího slova.

Obecně pro n-gramový jazykový model platí závislost na $n-1$ předchozích stavech, podmíněná pravděpodobnost má pak tvar:

$$P(w_k|w_1^{k-1}) \approx P\left(w_k|w_{k-n+1}^{k-1}\right). \tag{2.8}$$

Z rovnice (2.8) vyplývá, že pravděpodobnost posloupnosti slov na základě n-gramového modelu je

$$P(w_1^K) \approx \prod_{i=1}^{K} P\left(w_i|w_{i-n+1}^{i-1}\right) \tag{2.9}$$

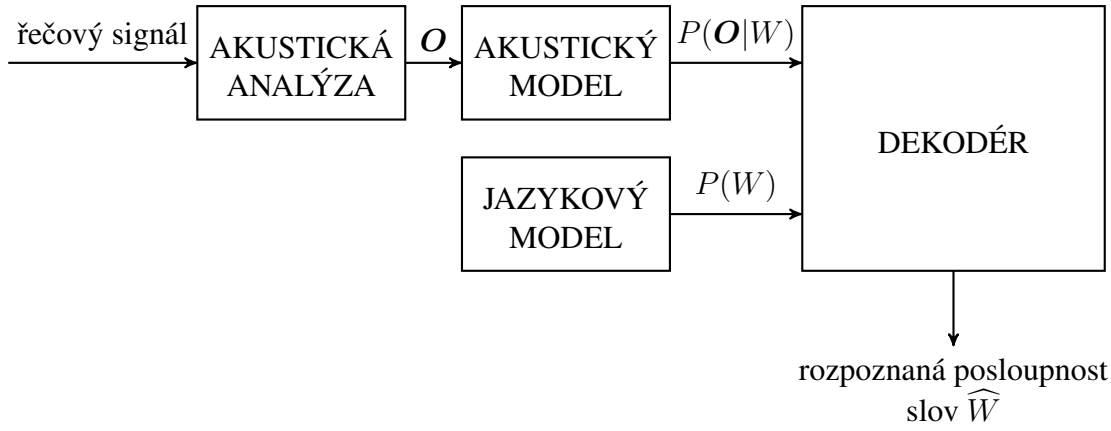
Pravděpodobnosti jednotlivých n-gramů se odhadují na základě relativní četnosti výskytu slov v trénovacích datech. Ovšem pro jazykový model s milionem slov vzniká 10^{18} potenciálních trigramů. Robustní odhad pravděpodobností je tedy téměř nemožný, navíc by bylo třeba prakticky nekonečné množství trénovacích dat. Proto se odhad těchto pravděpodobností počítá metodou maximální věrohodnosti s vyhlazováním.

Před samotným jazykovým modelováním se zpravidla provádí předzpracování trénovacích dat. To spočívá v čištění textu, tokenizaci, normalizaci a unifikaci textu. Velmi důležité také je z jakého zdroje jsou trénovací data získávána.

2.5 Dekódování

Z předchozích bodů je patrné, že posloupnost slov W procesem produkce řeči a následnou parametrizací řečového signálu je zakódována do posloupnosti pozorování $\boldsymbol{O}$. Proces dekódování se snaží najít posloupnost slov W na základě známé posloupnosti pozorování $\boldsymbol{O}$. Protože prohledávání celého stavového prostoru není výpočetně zvládnutelné, procesem dekódování je snaha nalézt nejpravděpodobnější posloupnost slov W v reálném čase. Snahou dekodéru tedy není nalézt skutečně vyslovená slova, ale slova, která byla s největší pravděpodobností vyslovena.

Jak ukazuje obrázek 2.7, pro dekódovací algoritmus musíme mít k dispozici několik důležitých dat. V první řadě se jedná o posloupnost vektorů pozorování $\boldsymbol{O}$, akustický model, který poskytuje HMM všech fonémů a neřečových událostí a podmíněnou pravděpodobnost $P(\boldsymbol{O}|W)$. Dále jazykový model poskytující pravděpodobnosti slov $P(W)$ a slovník včetně fonetických transkripcí slov. V neposlední řadě potom informace o rozpoznávané úloze.



Obr. 2.7: Blokové schéma systému rozpoznávání řeči dle řešených úloh.

2.5.1 Rozpoznávací síť

Rozpoznávací síť představuje stavový prostor všech posloupností slov W, které lze dekódovat. Zpravidla je rozpoznávací síť konstruována na bázi jazykového modelu, akustického modelu a slovníku. Nejprimitivnější strukturu rozpoznávací sítě lze realizovat lineárním zřetězením HMM fonémů jejich fonetických transkripcí, tedy tzv. lineární strukturou. Zřetězením na úrovni slov je realizováno zavedením zpětné smyčky a modelováním neřečových událostí.

2.5.2 Dekódování rozpoznávací sítě

Akustický model slova W lze reprezentovat s pomocí stavové posloupnosti S v rozpoznávací síti, tedy

$$P(\boldsymbol{O}|W) = \sum_{S} P(\boldsymbol{O}|S)P(S|W). \tag{2.10}$$

Jako hojně využívaný algoritmus pro dekódování rozpoznávací sítě je používán tzv. Viterbiho algoritmus [8], kde se dekódování řídí Viterbiho kritériem. Další možností je dekódování podle kritéria MAP [9]. Viterbiho kritérium má tvar:

$$\widehat{W} = \arg\max_{W} \left\{ P(W) \max_{S} P(\boldsymbol{O}|S)P(S|W) \right\} \tag{2.11}$$

Při výpočtu se kvůli možnému numerickému podtečení při násobení pravděpodobností toto kritérium převádí do logaritmické oblasti, viz. rovnice:

$$-\log\{\max_{s} P(\boldsymbol{O}|S)P(S|W)\} = \min_{s}\{-\log P(\boldsymbol{O}|S) - \log P(S|W)\} \tag{2.12}$$

Viterbiův algoritmus hledá cestu s nejmenší cenou v rozpoznávací síti s konečným počtem

stavů. Výsledkem dekódování je tedy nalezení nejpravděpodobnější posloupnosti slov $\widehat{W}$, která odpovídají příslušné posloupnosti vektorů pozorování $\boldsymbol{O}$.

Kapitola 3

Tvorba anotací a slovníků

3.1 Tvorba anotací v programu Transcriber

Aplikace Transcriber představuje nástroj, který umožňuje tvorbu manuálních anotací řečových signálů v uživatelsky přívětivém prostředí pro segmentaci rozsáhlých řečových nahrávek. Transcriber je pro nekomerční použití zcela zdarma [1].

3.1.1 Nastavení programu Transcriber

Při prvním spuštění aplikace Transcriber je doporučeno nastavit jméno anotátora, kódování a jazyk přepisu. Transcriber je při prvním spuštění nastaven defaultně v anglickém jazyce. Jazyk je možné změnit v základním nastavení aplikace, stejně tak i jméno anotátora a kódování. Pro přepisy zpracovávané v této práci bylo kódování nastaveno na Windows 1250. V nastavení lze také zadat interval automatického ukládání, což je mnohdy velmi užitečné v případě nečekaného pádu aplikace či počítače. Vytvořené anotace v Transcriberu je možné exportovat jako soubory `*.trs` či `*.xml` strukturované ve formátu XML. Podle standardu jsou ukládané přepisy pojmenovávány identicky ke zdrojovému zvukovému souboru.

3.1.2 Pravidla anotací

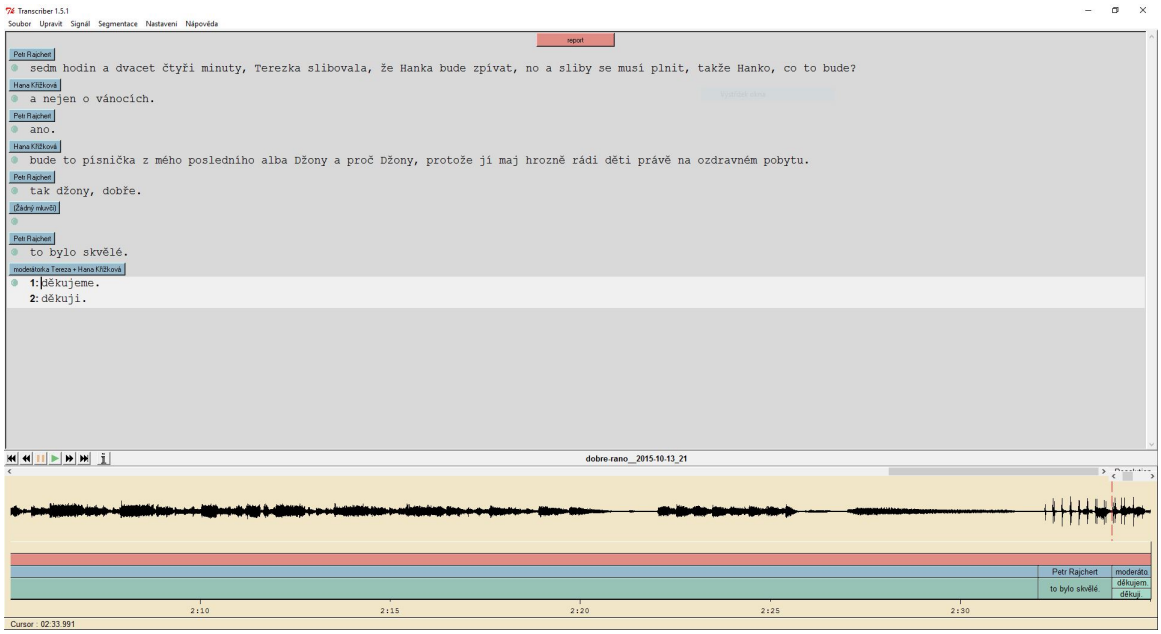
Anotace nahrávek je spojena s mnoha pravidly, která je nutné při jejich tvorbě zohledňovat a striktně dodržovat. V případě, že anotátor nějaká pravidla poruší, mohou při dalším zpracování anotovaných dat vzniknout nemalé problémy. Základní pravidla, která byla v rámci této práce dodržována, jsou uvedena v následujícím výčtu:

- Akustický signál, povětšinou soubor `*.wav`, je v programu Transcriber rozdělován na krátké úseky, tzv. segmenty. Jeden takový segment představuje zpravidla jednu větu, jejíž délka nepřesahuje tři řádky anotovaného textu. V případě vět delších než tři řádky je segment uměle rozdělen tak, aby byl smysl věty pokud možno zachován v obou segmentech.

- Konec každé věty je značen tečkou, vykřičníkem či otazníkem. Uvnitř vět je povoleno používat pouze čárky.
- Při umělém dělení věty na menší segmenty jsou tyto segmenty zakončeny čárkou, která značí, že v následujícím segmentu věta pokračuje.
- Věty začínají malými písmeny, velká písmena je povoleno používat pouze na začátku vlastních jmen *(Štěpánek, Nadal, Londýn, ...)* a při přepisu zkratek či názvů organizací *(BBC, ITF, ...)* atd.
- Pokud mluví více řečníků přes sebe, je nutné promluvu jednotlivých řečníků zapsat nejlépe do stejného segmentu jako novou větu oddělenou tečkou. V případě, že je promluva nesrozumitelná, je nutné označit tuto část jako samostatný segment a ponechat jej bez přepisu.
- Číslice jsou přepisovány slovní formou, nikoli číslicemi *(jednadvacet, dva tisíce šestnáct, ...)*.
- Je důležité přepisovat promluvu gramaticky správně. V případě nespisovné promluvy řečníka je tato promluva přepisována také v nespisovné formě.
- Při zadrhávání řečníka během promluvy se přepisuje pouze slovo, které bylo vyslovené celé, nikoli jen jeho nedořečenou část.
- Při přeřeknutí řečníka, kdy dojde k prohození dvou písmen či slabik a vznikne tím nesmyslné slovo, je třeba slovo přepsat pravopisně správně.
- Není nutné označovat neřečové události.
- Při vyslovení cizího slova řečníkem, které se však běžně používá, je nutné tato slova přepsat v počeštěném tvaru (*tak se dívejte na letošní šestý tačdaun*).
- Během promluvy může řečník některá slova vyslovit v cizím jazyce. Tato slova je nutno přepsat ortograficky správně.
- Všechna jména hráčů, rozhodčích, trenérů apod. zapisujeme do kulatých závorek `(...)` spolu s pádem v jakém dané slovo je. Závorkují se pouze aktuální sportovci, nikoliv jména nevztahující se ke sportu či sportovní legendy *((Erraniová 1) protestuje přes jestřábí oko)*
- Sportoviště, stadiony, kluby, státy a města se zapisují do hranatých závorek `[...]` společně s pádem ve kterém byly použity *(ve [Wimbledonu 6] nečekaně vypadla)*.

- Státní a klubová příslušnost je zapisována ve složených závorkách { . . . } společně s pádem ve kterém byly použity *(v řadě získaly {Američanky 1} a ujímají se)*.

Poslední tři pravidla popsaná ve výčtu výše jsou v přepisech označována pro potřeby obecného jazykového modelování pro daný sport, resp. událost. Právě pomocí označení slov zmíněnými závorkami (třídy) je možné dodat do jazykového modelu pro konkrétní událost (např. zápas), konkrétní údaje (např. jména sportovců).

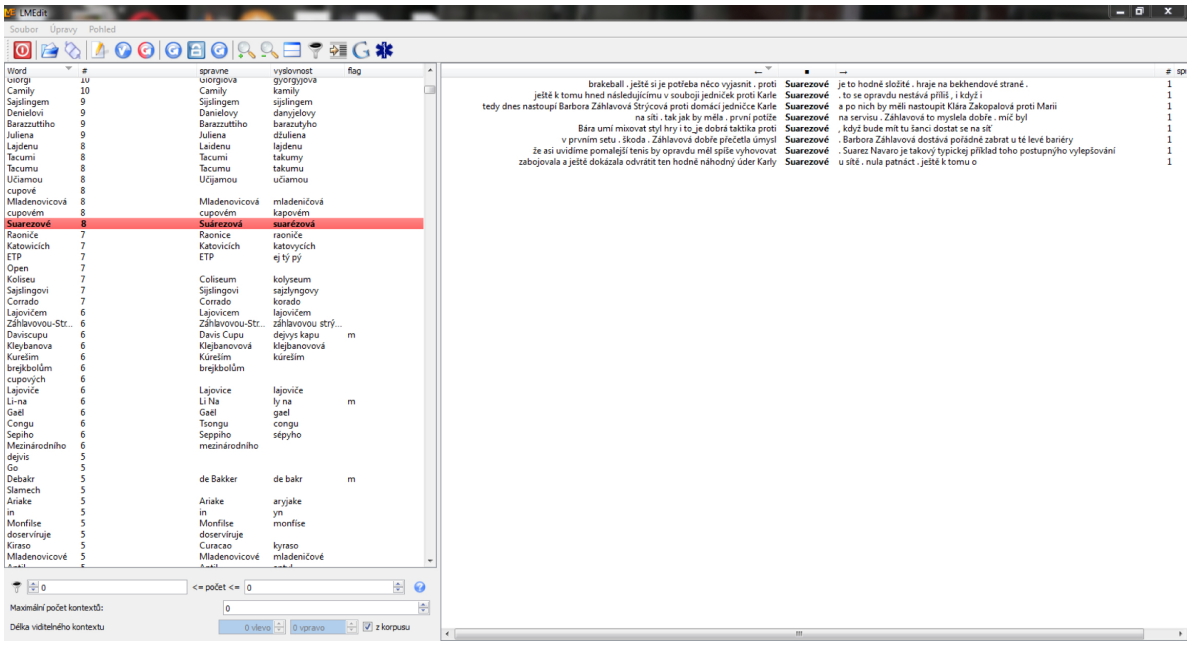


Obr. 3.1: Ukázka uživatelského rozhraní programu Transcriber.

3.2 Tvorba slovníků oprav v programu LMEdit

Po přepsání množiny rozsáhlých zvukových záznamů pomocí nástroje Transcriber je nutné provést kontrolu přepsaných slov využitím skriptu pro kontrolu slov. Skript funguje na principu slovníku, který obsahuje více než tři miliony českých slov. Jestliže se vyskytne slovo, které není obsaženo ve slovníku, je vytvořen soubor neznámých slov, který se nadále zpracovává v programu LMEdit. Jedná se především o slova obsahující překlepy, jména sportovců, sportovišť, atd.

Pomocí programu LMEdit dojde k označení správnosti slov, opravě slov a, je-li nutné, k nadefinování výslovnosti daného slova. Následně se nově zpracovaný slovník přidá k již existujícím rozsáhlým slovníkům.



Obr. 3.2: Ukázka uživatelského rozhraní aplikace LMEdit.

Na obrázku 3.2 je zobrazena podoba uživatelského rozhraní programu LMEdit. V levém sloupci se nacházejí neznámá slova. Druhý sloupec obsahuje četnost výskytu daných slov v kontrolovaných zdrojových textech. Ve třetím sloupci jsou uvedeny správné tvary daného slova. Poslední sloupec pak obsahuje správnou výslovnost daného slova. V pravé části program LMEdit je zobrazen kontext, ve kterém se určité slovo v textu vyskytuje.

Pro český jazyk existuje řada slovníků, od malých obsahujících konkrétní skupiny speciálních znaků až po velmi rozsáhlé slovníky. Následující tabulka 3.1 poskytuje přehled dostupných základních slovníků pro český jazyk.

slovník	**počet položek**
běžná česká slova	∼ 3 miliony
příjmení v ČR	∼ 952 tisíc
názvy firem registrovaných v ČR	∼ 341 tisíc
názvy obcí a ulic v ČR	∼ 145 tisíc
ostatní slova, názvy apod.	∼ 16 000
křestní jména v ČR	∼ 10 000
názvy států, národností, jazyků a velkých měst	∼ 10 000
sportovní výrazy	∼ 5 000
čísla	590
interpunkční znaménka a příkazy	20

Tab. 3.1: Tabulka rozsáhlosti slovníků jednotlivých okruhů. Data byla získána z [7].

3.2.1 Postup provádění oprav ve slovníku

V procesu zpracování slovníku je nutné správně zapisovat velikost písmen. Při opravě daného slovníku se může vyskytnout několik situací, které budou uvedeny níže.

1. **Slovo existuje výhradně samostatně a má svůj vlastní význam**
   Hovorová slova je potřeba zapsat v gramaticky správném tvaru, přičemž do sloupce výslovnosti se uvádí výslovnost nespisovné varianty spolu s výslovností spisovného tvaru, v případě že nejsou stejné stačí uvést jednu výslovnost. Pokud nelze hovorový výraz nahradit stejně dlouhým výrazem spisovným, ponechá se hovorový výraz, popř. jeho spisovnější varianta.

   ```
   vzájemě    vzájemně    vzájemě
   pomalej    pomalý      pomalej; pomalý
   Roterdamu  Rotterdamu  rotrdamu
   point      point       pojnt
   ```

2. **Slovo existuje pouze ve spojení s jiným výrazem a tvoří tak běžně používaný významový celek**
   Nutno vyhledat v kontextu časté výrazy se kterými zpracovávané slovo tvoří významové celky. A to omezením viditelného kontextu slova takovým způsobem, aby zůstal pouze významový celek, zároveň je nutné zapsat do sloupce `flag` písmeno "m". Vyskytne-li se ve slovníku špatně rozdělené slovo je vhodné jej spojit, v tomto případě nedochází k zápisu do sloupce `flag`.

   ```
   dejvicsup  Davis Cup  dejvys kap
   kapový     fedcupový  fedkapový   m
   ```

3. **Slovo existuje jak samostatně s vlastním významem, tak často i ve spojení s jiným výrazem a významem**
   Postup je stále stejný jako u dvou předchozích příkladů. Tento příklad se týká zejména příjmení sportovců, které je nutné zpracovat samostatně podle prvního bodu, ale i v kombinaci s křestním jménem, popřípadě podle bodu dva s různými křestními jmény sportovců se stejným příjmením.

   ```
   Tacuma  Tacuma      takuma
   Li-Na   Li Na       ly na       m
   Tacuma  Tacuma Itó  takuma yto  m
   ```

4. **Slovo nebylo možné zpracovat podle předchozích bodů**
   V tomto případě se zapíše do sloupce `flag` k danému nezpracovanému slovu `"x"`.

Kapitola 4

Analýza a detekce chyb v anotacích

Pro kvalitní akustické modely je více než žádoucí vycházet při trénování z dostatečně robustních, ale zejména bezchybným anotací. Tento požadavek je ovšem velmi těžko realizovatelný, neboť anotace tvoří lidé a ti občas chybují. Tato kapitola je zaměřena na analýzu nejčastějších chyb v transkripcích, kterých se anotátoři dopouštějí, v případě této práce při anotacích sportovních přenosů. Dále je v této kapitole popsán obecný postup, jak lze v anotačních textech tyto chyby detekovat a eliminovat.

4.1 Analýza nejčastějších chyb v anotacích

Při tvorbě anotací může dojít k mnoha chybám, od snadno detekovatelných překlepů až po případy, kdy o chybě dokáže rozhodnout pouze lidský korektor. Chyby v anotacích lze rozdělit na dva základní typy:

1) Slovní přepis nesouhlasí s obsahem relevantního akustického signálu - v anotaci se objevují překlepy, špatné pády, nepárové závorky či chybějící interpunkce.
2) Segmentace akustického signálu je provedena nesprávně ze strany anotátora - hranice segmentu narušuje zvukový signál, který by měl či neměl být součástí daného segmentu.

4.1.1 Chyby v anotacích

V anotacích se na základě pozorování objevuje poměrně velké množství chyb. Mezi nejčastější chyby, kterých se anotátoři při své práci dopouštějí, patří:

a) **Překlepy** - bývají nejčetnější chybou ve sportovních anotacích. Velmi často se jedná o prohozená písmena, nedokončená slova, chyby při psaní znaků s diakritikou, malá či velká písmena anebo více velkých písmen na začátku slova. Častou a snadno detekovatelnou chybou jsou také chyby, kdy jsou místo znaků s diakritikou napsány číslice.

jak **sjem** řekl
PEtra Kvitová

b) **Neřečená slova navíc** - Často se stává, že anotátor přepíše slova, která ve skutečnosti řečena nebyla. Tyto chyby většinou vznikají nevědomě domýšlením obsahu přepisované věty, když anotátor přepisuje přehrávanou zvukovou stopu. U většiny případů se jedná o krátká slova, tj. spojky, zvratná slovesa či předložky. Tento typ chyby se velmi těžko detekuje, neboť bez kontroly zvukového segmentu nelze s jistotou prohlásit, zda dané slovo do anotace patří či nikoliv.

Anotace:
takže v tom je **to** nebezpečnější, že obě holky
Nahrávka:
takže v tom je nebezpečnější, že obě holky

c) **Chybějící slova** - Chyba tohoto typu vzniká podobně jako chyba s neřečenými slovy navíc, resp. nepozorností anotátora. Opět se povětšinou jedná o krátká slova. Detekce těchto chyb bez poslechu zvukového segmentu je velmi obtížná.

Anotace:
po dvaadvaceti letech si zahrají finále této týmové ženské soutěže
Nahrávka:
po dvaadvaceti letech si zahrají **znovu** finále této týmové ženské soutěže

d) **Nespisovný či nesprávný tvar slova** - Přepsané slovo je uvedeno v nesprávné formě nebo v nespisovné formě, ačkoliv bylo vysloveno spisovně.

e) **Slovní vyjádření číslovek** - Veškeré číslovky a data musejí být přepsány ve slovní formě, jedinou výjimkou je označování pádů u tříd (viz 3.1.2).

4.1.2 Chyby v segmentaci

Při přepisování zvukových nahrávek v programu Transcriber (popsán v kapitole 3) může anotátor chybně rozdělit jednotlivé segmenty. Nejčastější chybou je ukončení segmentu uprostřed slova, popřípadě vytvoření příliš dlouhých segmentů s krátkou anotací následovanou dlouhým šumem.

4.2 Automatická detekce a oprava chyb v anotacích

Automatická detekce chyb, tedy proces, při kterém jsou v anotacích strojově vyhledávány chyby, je povětšinou realizována s použitím slovníků. Před procesem detekce a opravování chyb je tedy nutné mít k dispozici kvalitní a robustní slovníky, které definují správné tvary slov, jejich špatné formy a správné výslovnosti daných slov (viz 3.2).

Na základě připravených slovníků jsou pak v anotačních textech vyhledávány výrazy ze slovníků a nahrazovány jejich gramaticky správnou formou. Tento proces tvoří základní metodu detekce a opravy anotačních chyb. Z hlediska nahrazování je velmi důležité, aby tento proces detekce a nahrazování probíhal od nejdelších výrazů ve slovníku. To je prováděno z důvodu, aby byly výrazy nahrazovány správně například v případech, kdy je výraz složen ze dvou slov na sobě závislých. Na příkladu níže je tento problém znázorněn. V případě A nebyly výrazy nahrazovány od nejdelších výrazů ze slovníku a tím vznikla v opravených anotacích další chyba, kterou je těžké detekovat a opravit, neboť oba dva výrazy mají správný tvar ale neodpovídají původnímu významu. V případě B byly výrazy správně nahrazeny od nejdelších výrazů ve slovníku.

A) Špatná oprava:

`džejms blejk` → `džejms brejk` → `James brejk`

B) Správná oprava:

`džejms blejk` → `James Blake`

Kapitola 5

Předzpracování anotačních dat pro nástroj HTK

Před použitím nástroje HTK, který je detailně popsaný v kapitole 6, je nutné předzpracovat anotační texty a audionahrávky do formátu požadovaného nástrojem HTK. Konkrétněji, je nutné segmentovat zvuková data na úseky odpovídající jednotlivým větám do samostatných souborů a následně vytvořit ucelený přepis všech dat na úrovni slov.

5.1 Segmentace zvukových dat

Pro tvorbu akustických modelů je nutné mít k dispozici dostatek souborů obsahujících přepisy zvukových nahrávek. Tyto přepisy vytvořené v aplikaci Transcriber jsou ukládány ve formátu `*.trs`. Soubory tohoto typu jsou strukturovány jako XML, což usnadňuje vyhledávání a práci s informacemi v nich obsažených. Ukázka `trs` souboru je níže.

```
<?xml version="1.0" encoding="CP1250"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="Jméno anotátora"
       audio_filename="CeskoItalie-190414-00" version="16"
       version_date="140806">
        <Episode>
                <Section type="report" startTime="0" endTime="1544.128">
                        <Turn startTime="0" endTime="1544.128">
                                <Sync time="0"/>

                                <Sync time="32.146"/>

                                <Sync time="58.965"/>
                                po šesté v řadě hrají {české 1}...
                                ...
                        </Turn>
                </Section>
        </Episode>
</Trans>
```

5.1.1 Nalezení audio souborů k transkripcím

Před samotnou segmentací bylo nutné ošetřit následující situace:

1. Pro daný přepis neexistuje zvuková nahrávka - jméno zvukového souboru spojeného s přepisem je definováno přímo v souboru s transkripcí. Pokud takovýto soubor neexistuje, přepis je vyřazen z procesu segmentace. Informace o zvukovém souboru spojeném s transkripcí je uvedena v atributu `audio_filename`.
2. Název souboru s transkripcí se neshoduje s názvem spojeného zvukového souboru - soubory transkripcí jsou ukládány tak, aby jejich název odpovídal názvu zvukového souboru, který byl anotován. Pokud tomu tak není, je tento přepis vyřazen z procesu segmentace.

5.1.2 Nalezení mezních časů jednotlivých segmentů

Z každého souboru transkripce je nutné získat časové údaje začátku a konce jednotlivých segmentů. Tyto informace jsou obsaženy v atributu `time` u prvků `Sync` v dané XML struktuře souborů Transcriberu. V případě, že daný segment nemá přepis (viz první segment v ukázce níže), je tento úsek ignorován a není zpracováván.

```
<Sync time="0"/>

<Sync time="5.423"/>
a vlastně ani (Karin Knappová 1) není o tolik zkušenější,...
<Sync time="14.02"/>
zahrála si už za rozhodnutého stavu loňské finále na [Sardínii 6] ...
<Sync time="23.422"/>
...
```

Pro poslední segment je čas konce tohoto segmentu uložen v atributu `endTime` u nadřazeného prvku `Turn`.

```
<Turn startTime="0" endTime="365.995">
```

Pro takto analyzovaná data musí být vytvořen textový soubor, jehož struktura je následující:

```
Vstup = CeskoItalie-200414-10.wav
5.423       14.02       CeskoItalie-200414-10_01.wav
14.02       23.422      CeskoItalie-200414-10_02.wav
```

První sloupec obsahuje časy začátku segmentu, prostřední čas konce segmentu a třetí sloupec název souboru, do kterého bude tato část původního zvukového souboru vyříznuta. Tento soubor je poté možné použít jako vstupní soubor pro program `RizniWave.exe`, který na základě tohoto souboru vstupní data rozčlení do jednotlivých souborů.

5.2 Tvorba souboru s transkripcemi na úrovni slov

Pro potřeby nástroje HTK je velmi důležité vytvořit soubor `words.mlf` obsahující transkripce jednotlivých audio souborů na úrovni slov. Tento MLF[1] soubor má na první řádce definovanou hlavičku `#!MLF!#`. MLF soubor je koncipován jako obrovská databáze transkripcí jednotlivých segmentovaných nahrávek přidružených k daným souborům (viz ukázka níže). Jednotlivé transkripce daných segmentů jsou vždy zakončeny tečkou (`.`).

```
"*/CeskoItalie-200414-10_0.lab"
a
vlastně
ani
Karin
Knappová
není
o
tolik
zkušenější
...
body
.
"*/CeskoItalie-200414-10_1.lab"
zahrála
si
už
za
rozhodnutého
stavu
loňské
finále
na
Sardínii
...
vyhrály
.
"*/CeskoItalie-200414-10_2.lab"
...
```

[1] MLF = Master Label File.

Kapitola 6

Trénování akustických modelů pomocí nástroje HTK

Akustické modely pro rozpoznávání řeči je možné natrénovat pomocí nástroje HTK (Hidden Markov Model Toolkit). Tento nástroj slouží ke tvorbě a manipulaci s HMM a je primárně využíván k rozpoznávání řeči [6]. Aby bylo možné s trénováním akustických modelů v nástroji HTK začít, je nutné v první řadě analyzovat a připravit vstupní data pro trénování, jak je popsané v předchozí kapitole 5. S připravenými daty je pak možné natrénovat akustické modely podle postupu popsaného v následujících odstavcích.

6.1 Postup trénování akustického modelu

6.1.1 Vytváření souborů s přepisem na úrovni fonémů

Navrhovaný systém je založen na modelování konkrétních fónů, nikoliv na větších jednotkách, jako jsou slova. Jednotlivé fóny jsou reprezentovány příslušnými HMM, proto kromě anotací promluv na úrovni slov bylo zapotřebí vytvořit anotace na úrovni fónů. K tomu byl využit program `HLEd` nástroje HTK, který pomocí připravené transkripce na úrovni slov a slovníku výslovností vytvoří právě anotace na úrovni fónů (soubor **`phones0.mlf`**).

```
HLEd -l * -d dict_sp.txt -i phones0.mlf mkphones0.led words.mlf
```

Pro tento příkaz je zapotřebí vytvořit soubor `mkphones0.led`, který obsahuje následující příkazy:

```
EX
IS _sil_ _sil_
DE _sp_
```

Příkaz `EX` způsobuje, že je každé slovo ze souboru `words.mlf` nahrazeno jeho příslušnou fonetickou transkripcí vyhledanou ve slovníku `dict_sp.txt`. Příkaz `IS _sil_`

sil vloží fón _sil_ (značící dlouhou pauzu) na začátek a konec každé věty. Nakonec příkaz DE _sp_ odstraní všechny výskyty fónu _sp_ (značící krátkou pauzu) z důvodu prvotního trénování modelu bez tohoto fónu.

Analogickým způsobem se vytváří fónová transkripce promluv obsahující fón _sp_ (soubor **phones1.mlf**).

```
HLEd -l * -d dict_sp.txt -i phones1.mlf mkphones1.led words.mlf
```

Pro tento příkaz je zapotřebí vytvořit soubor mkphones1.led, který na rozdíl od mkphones0.led neodstraňuje fón _sp_.

```
EX
IS _sil_ _sil_
```

Níže jsou uvedeny příklady souborů phones0.mlf (vlevo) obsahující přepis vět na úrovni fónů bez krátké pauzy _sp_ a souboru phones1.mlf (vpravo), který již tyto pauzy obsahuje.

```
#!MLF!#
"*/CeskoItalie-200414-10_0.lab"
_sil_
a
v
l
a
s
T
J
e
a
J
i
...
d
v
a
b
o
d
i
_sil_
.
```

```
#!MLF!#
"*/CeskoItalie-200414-10_0.lab"
_sil_
a
_sp_
v
l
a
s
T
J
e
_sp_
a
J
i
_sp_
...
_sp_
d
v
a
_sp_
b
o
d
i
_sp_
_sil_
.
```

6.1.2 Parametrizace řečových dat

Dalším krokem je parametrizace řečových dat, čímž jsou převáděny zvukové nahrávky z formátu `wav` (resp. jiného zvukového formátu) na posloupnost vektorů parametrů (LPC, MFCC, PLP). Parametrizace "off-line"je výhodná z hlediska úspory času při trénování, kdy provádíme parametrizace pouze jednou, nikoliv při každém trénovacím cyklu. Pro parametrizaci se využívá program `HCopy` nástroje HTK s následujícími parametry:

```
HCopy -T 1 -C cf_param.mfc -S param.scp
```

6.1.3 Tvorba monofónních modelů

Aby bylo možné trénovat akustické modely, je nutné vytvořit definici HMM model pro nástroj HTK ve formátu, který je v tomto nástroji podporován. Modely fónů jsou reprezentovány pětistavovým HMM, viz obrázek 2.4.

Definice zmíněného modelu se z konvence ukládá do souboru s názvem `proto`.[1] V tomto souboru jsou definovány velikosti vektorů parametrů, jejich typ, počet stavů, jméno modelu, střední hodnoty jednotlivých stavů a kovarianční matice HMM.

Pravděpodobnost generování jednotlivých vektorů parametrů v emitujících stavech se řídí Gaussovským rozdělením pravděpodobnosti, u kterého potřebujeme znát již zmíněnou střední hodnotu a kovarianci, resp. vektor středních hodnot a kovarianční matici. Často je používána diagonální kovarianční matice, kdy je uchováván vektor diagonálních prvků.

Po vytvoření definičního souboru je možné přepočítat střední hodnoty a kovariance prototypu HMM pomocí programu `HCompV` s parametry:

```
HCompV -C cf.mfc -f 0.01 -m -S train.scp -M hmm0 proto
```

Tento příkaz spočítá celkovou střední hodnotu a kovarianci ze všech trénovacích dat uložených v souboru `train.scp` a nastaví všechna Gaussovská rozdělení modelu `proto` na takto spočtené hodnoty. Do adresáře **`hmm0`** je pak následně uložen výstup modelu `proto`.

Následně je nutné ve vytvořeném adresáři vytvořit tzv. MMF[2] soubor, který obsahuje definice pro jednotlivé fóny, které jsou kopiemi modelu `proto`. Tento proces kopírování modelů zajišťuje program `MakeMMF` s následujícími parametry:

```
MakeMMF proto monophones0 vFloors models
```

Skript vytváří MMF soubor **`models`**, přičemž soubor `vFloors` je meziproduktem výpočtu.

V tomto bodě jsou modely inicializované a připravené k trénování. Trénování, nebo také reestimace, je prováděna pomocí programu `HERest` nástroje HTK, který funguje na Baum-Welchově algoritmu známým jako Forward-Backward algoritmus. Reestimace musí být spouštěna v adresáři, v němž jsou inicializované HMM, tedy

```
HERest -T 1 -C cf.mfc -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp
                    -H hmm0/models -M hmm1 monophones0
```

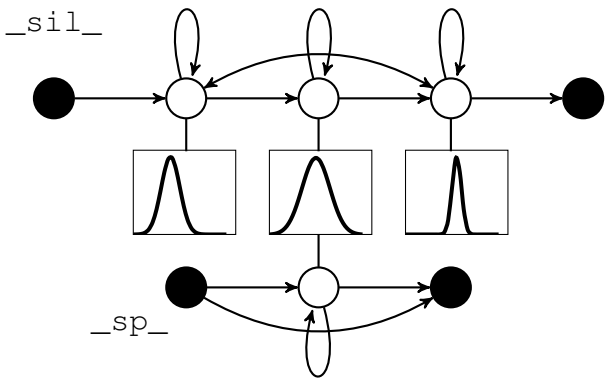
[1] Název `proto` je odvozen z anglického *prototype*, což vystihuje význam souboru.

[2] MMF (z anglického *Master Macro File*) obsahují definice HMM pro jednotlivé monofóny.

Tímto příkazem jsou HMM modely reestimovány s použitím trénovacích dat `train.scp` a výsledky uloženy do adresáře **`hmm1`**. Pro dostatečné natrénování modelů je třeba provést více iterací reestimace s tím, že pokaždé jsou výsledky reestimace ukládány do nového adresáře `hmm%`.

6.1.4 Úprava modelů pauz

Doposud bylo při trénování využíváno pouze jednoho modelu dlouhé pauzy `_sil_`. Model dlouhé pauzy měl stejnou strukturu HMM jako ostatní fonémy. Je však nutné modelovat různé pauzy od velice krátkých po velmi dlouhé, a proto je nutné změnit strukturu modelu pauzy tak, aby byl model schopný absorbovat všechny vyskytující se šumy v pauzách. Do původního modelu je nutné přidat přechod z druhého do čtvrtého stavu a naopak. Dále je zapotřebí vytvořit zcela nový model, který modeluje krátké pauzy `_sp_` mezi slovy. Tento model krátké pauzy `_sp_` má pouze jeden emitující stav, který sdílí parametry s prostředním stavem modelu dlouhé pauzy `_sil_`. Pro představu je na obrázku 6.1 vyobrazeno svázání těchto dvou modelů pauz.



Obr. 6.1: Schéma modelu krátké a dlouhé pauzy.

Pro úpravu modelu `_sil_` a vytvoření nového modelu `_sp_` slouží program `HHed` s parametry:

```
HHed -T 1 -H hmm4/models sil.hed monophones1
```

Tento příkaz otevře adresář s poslední reestimací `hmm4` a v něm nacházející se soubor **`models`**. Sadou příkazů zapsaných v `sil.hed` a `monophones1` vykoná sekvenci úprav (popsaných ve zmíněném souboru `sil.hed`) ve výstupním souboru `models`. Nejprve bylo nutné vytvořit soubor `sil.hed`, který obsahuje následující příkazy:

```
AT 2 4 0.2 {_sil_.transP}
AT 4 2 0.2 {_sil_.transP}
```

```
AT 1 3 0.3 {_sp_.transP}
TI slist {_sil_.state[3],_sp_.state[2]}
```

Příkaz `AT 2 4 0.2 {_sil_.transP}` přidá do modelu pauzy `_sil_` přechod z druhého stavu do čtvrtého stavu, kterému odpovídá pravděpodobnost přechodu 0.2. Druhý v pořadí je příkaz `AT 4 2 0.2 {_sil_.transP}` a ten přidá opět do modelu přechod ze čtvrtého stavu do stavu druhého s pravděpodobností 0.2. Příkaz `AT 1 3 0.3 {_sp_.transP}` doplní do modelu `_sp_` přechod z prvního stavu do třetího s pravděpodobností 0.3. Je nutné zmínit, že oba stavy jsou neemitující. Poslední ze sady příkazů je `TI slist {_sil_.state[3],_sp_.state[2]}` , který sváže třetí stav modelu `_sil_` se druhým stavem modelu `_sp_`.

6.1.5 Přerovnávání trénovacích dat

Trénovací data je pak nutné přerovnat za pomoci Viterbiho algoritmu. Program `HVite` dokáže rozpoznávat a přerovnávat trénovací data pro akustické modely. Program z transkripce na úrovni slov vytváří novou transkripci na úrovni fonémů, a pokud existuje ve slovníku více variant fonetické transkripce daného slova, vybírá nejvhodnější fonetickou transkripci odpovídající trénovacím datům a dosud natrénovaným datům.

Před samotným přerovnáváním slovníku je vhodné ve slovníku provést úpravu zdvojení jednotlivých fonetických transkripcí všech slov. Jednou bude transkripce končit krátkou pauzou `sp` a jednou dlouhou pauzou `sil`. Program `HVite` vybere pauzu za slovem, která nejlépe vyhovuje akustickým trénovacím datům.

```
HVite -T 1 -1 * -y lab -o SWT -b _SIL_ -C cf.mfc -m -a -H hmm7/models
-i aligned.mlf -t 250.0 -I words.mlf -S train.scp dict.txt monophones1
```

Výstupem přerovnávání programu `HVite` je soubor **`aligned.mlf`**, který obsahuje nové monofónní transkripce trénovacích promluv. Pokud se nepodařilo zarovnat transkripci promluvy z trénovací promluvy, v souboru `aligned.mlf` se tato transkripce nevyskytne a je nutné ji odstranit z trénovacích dat `train.scp`. K tomu napomůže program `CreateAligned`:

```
CreateAligned.exe aligned.mlf train.scp aligned.scp ne.scp
```

Výstupem tohoto programu jsou soubory **`aligned.scp`** a **`ne.scp`**. V prvním souboru je uveden seznam s dobře přerovnanými promluvami určený k dalšímu trénování a ve druhém nepřerovnané promluvy. Následně je nutno provést několik reestimací (pomocí programu `HERest`) s novým monofónním přepisem `aligned.mlf` a novým trénovacím seznamem `aligned.scp`.

```
HERest -T 1 -C cf.mfc -I aligned.mlf -t 250.0 150.0 1000.0 -S
aligned.scp -H hmm7/models -M hmm8 monophones1
```

6.1.6 Přidávání složek

Pro zvýšení robustnosti akustického modelu se zavádí vícesložkové rozložení jednotlivých fónů. Přidáním složek jsou dané fóny reprezentovány směsí středních hodnot a variancí, což ve výsledku přidává ke schopnosti správného zařazení fónu při rozpoznávání. V nástroji HTK je k tomu využíván program `HHed`, který přidává složky rozložení daných fonémů akustického modelu.

`HHEd -T 1 -A -C cf.mfc -H hmm_(`$N-1$`)_4/models -M hmm_`N`_1 add_next.hed monophones1 > hmm_`N`_1/log.txt`

Po přidání složky rozložení bylo nutné vzniklé akustické modely opět reestimovat pomocí programu `HERest`.

`HERest -T 1 -C CF.mfc -I aligned.mlf -t 200.0 100.0 500.0 -S aligned.scp -H hmm_`N`_`k`\models -M hmm_`N`_(`$k+1$`) monophones1 > hmm_`N`_(`$k+1$`)\log.txt`

6.1.7 Rozpoznávání

Po natrénování monofonového akustického modelu může být zahájen proces rozpoznávání zvukových nahrávek. Pro rozpoznávání je třeba připravit rozpoznávací síť ze slov testovacích dat. Pro vytvoření rozpoznávací sítě je možné využít program `VytvorSit`, jehož vstupem je seznam slov z testovacích dat a výstupem soubor `wdnet` obsahující popis a strukturu rozpoznávací sítě. Potřebný seznam slov je získán z anotací relevantních k daným testovacím souborům.

`VytvorSit wlist.txt` **`wdnet`**

S vygenerovanou rozpoznávací sítí a natrénovanými akustickými modely je s využitím již zmíněného programu `HVite` provedeno rozpoznávání testovacích dat. Program `HVite` se spouští s následujícími parametry:

`HVite -C cf.mfc -H hmm11/models -S test.scp -i` **`vysledek_all.txt`** `-l * -p -60 -w wdnet dict.txt monophones1`

Do souboru **`vysledek_all.txt`** jsou zapsány výsledky rozpoznávání jednotlivých částí testovacích dat. Shrnující informace o rozpoznávání jsou pak získány pomocí programu `HResults` a zapsány do souboru **`vysledek.txt`**. Tento program poskytuje informace o úspěšnosti rozpoznávání "neznámé"promluvy. Je logické, že musí být k dispozici informace o skutečném obsahu těchto "neznámých"promluv. Tato informace je k dispozici v souboru s transkripcemi `words.mlf`, přičemž program `HResults` porovná rozpoznané promluvy s jejich skutečnou anotací a do souboru **`vysledek.txt`** vypíše podrobný přehled výsledků rozpoznávání.

```
HResults -f -I words.mlf monophones1 vysledek_all.txt > vysledek.txt
```

O struktuře souboru `vysledek.txt` s výsledky rozpoznávání a evaluaci natrénovaných akustických modelů se dočtete v následující sekci.

6.1.8 Úspěšnost rozpoznávání

Po otestování akustických modelů trénovacími daty je získán z programu `HResults` textový soubor `vysledek.txt` ve kterém se nachází výsledky rozpoznávání v následující struktuře:

```
------------------------ Sentence Scores --------------------------
====================== HTK Results Analysis =======================
  Date: Mon Aug 15 20:05:19 2016
  Ref : HTK\scripts\words.mlf
  Rec : HTK\scripts\vysledek_all.txt
------------------------- File Results --------------------------

CeskoItalie-190414-08_0.rec:    0.00(  0.00)  [H=   0, D=  0, S=  2, I=  0, N=  2]
CeskoItalie-190414-08_1.rec:    0.00(  0.00)  [H=   0, D= 15, S= 28, I=  0, N= 43]
CeskoItalie-190414-08_10.rec:    0.00(  0.00)  [H=   0, D= 23, S= 16, I=  0, N= 39]
CeskoItalie-190414-08_100.rec:   10.53( 10.53)  [H=   2, D=  3, S= 14, I=  0, N= 19]
CeskoItalie-190414-08_101.rec:    0.00(  0.00)  [H=   0, D=  5, S=  9, I=  0, N= 14]
CeskoItalie-190414-08_102.rec:    0.00(  0.00)  [H=   0, D= 11, S= 13, I=  0, N= 24]
CeskoItalie-190414-08_103.rec:   14.63( 14.63)  [H=   6, D= 13, S= 22, I=  0, N= 41]
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=101, N=101]
WORD: %Corr=3.53, Acc=3.15 [H=65, D=682, S=1093, I=7, N=1840]
===================================================================
```

Přesnost rozpoznávání mluvené řeči je vyhodnocovaná pomocí dvou vzorců uvedených níže.

$$Acc = \frac{N - D - I - S}{N} \cdot 100(\%) \tag{6.1}$$

Vzorec 6.1 pro výpočet procentuální úspěšnosti rozpoznávání je v případě této práce vhodnější z důvodu zahrnutí chyb vkládání.

$$Corr = \frac{N - D - S}{N} \cdot 100(\%) \tag{6.2}$$

Kde N znamená počet slov v referenčních přepisech, D má význam počtu slov, která chybí v rozpoznaném textu. Proměnná I říká kolik slov se objevuje navíc v rozpoznaném textu a hodnota S je počet slov, která se v textech neshodují. V sekci `Overall Results` jsou celkové výsledky rozpoznávání akustického modelu. V první jsou zmíněny výsledky rozpoznávání vět(`SENT`) a v této práci podstatnější výsledky rozpoznávání daných slov (`WORD`).

Kapitola 7

Experimenty nad reálnými daty

Pro přípravu a zpracování dat pro trénování v HTK byl zvolen programovací jazyk Python, který umožňuje snadnou manipulaci s textovými daty a soubory. Zároveň s tím byl v tomto jazyce napsán skript, který celý proces trénování řídí. Tento skript je stručně popsaný v sekci 7.2

7.1 Data pro trénování sportovních akustických modelů

Z celkového počtu 160 párů zvukových souborů a přepisů vyhovovalo podmínkám viz. podkapitola 5.1.1 156 souborů. Zbylé 4 soubory byly z trénování vyřazeny a nebylo s nimi počítáno, aby nedošlo k možnému zkreslení výsledků trénování a rozpoznávání. Celkem tato zvuková data představují přes 91 hodin zvukového záznamu ze 44 přenosů tenisových utkání, v nichž figurovalo minimálně 17 různých řečníků. S těmito soubory byla provedena segmentace popsaná v kapitole 5.1. Paralelně se segmentací byl vytvářen i soubor `words.mlf` obsahující přepis segmentovaných úseků na úrovni slov, viz kapitola 5.2. Tento přepis obsahuje celkem 29 789 různých slov.

Výsledkem segmentace bylo získáno celkem 15 603 zvukových segmentů představující zvukovou stopu jednotlivých vět z anotačních textů. Z těchto souborů byly vybrány dvě množiny. První množina představující trénovací data ke tvorbě akustických modelů, která je tvořena 90% celkových dat, tj. 15 503 souborů. Druhá množina představuje testovací data, a tedy 10% původních dat, tj. 100 souborů.

Pro trénování akustických modelů byly připraveny dvě sady dat. První trénovací sada byla tvořena z původních dat. Druhá trénovací sada dat pak byla tvořena opravenými daty. Tyto opravy byly provedeny na základě znalostí ze sekce 4.2 pomocí dvou slovníků speciálně připravených pro tuto práci. Byl připraven obecný slovník obsahující výrazy relevantní k přenosům tenisových utkání (celkem 37 906 záznamů) a menší slovník konkrétních oprav obsahující další výrazy relevantní k přenosům tenisových utkání (celkem 2 911 záznamů).

Z takto připravených trénovacích dat byly natrénovány akustické modely dle popisu v kapitole 6.1. Do obou akustických modelů byly přidávány složky dokud docházelo k

razantním zlepšením při rozpoznávání, proto byly natrénovány akustické modely se 30. složkami.

7.2 Skript pro řízení trénování v HTK

Pro snazší manipulaci s nástrojem HTK a usnadnění procesu trénování byl v jazyce Python napsán skript, který výše zmíněný postup realizuje prostřednictvím volání definovaných metod.

Celý proces popsaný v sekci 6.1 tak lze snadno převést do následující sekvence příkazů s využitím modulů `htk.py`. Ukázka použití třídy `HTKToolkit` je uvedena níže, přičemž výčet jejích metod s popisem je uveden v tabulce 7.1.

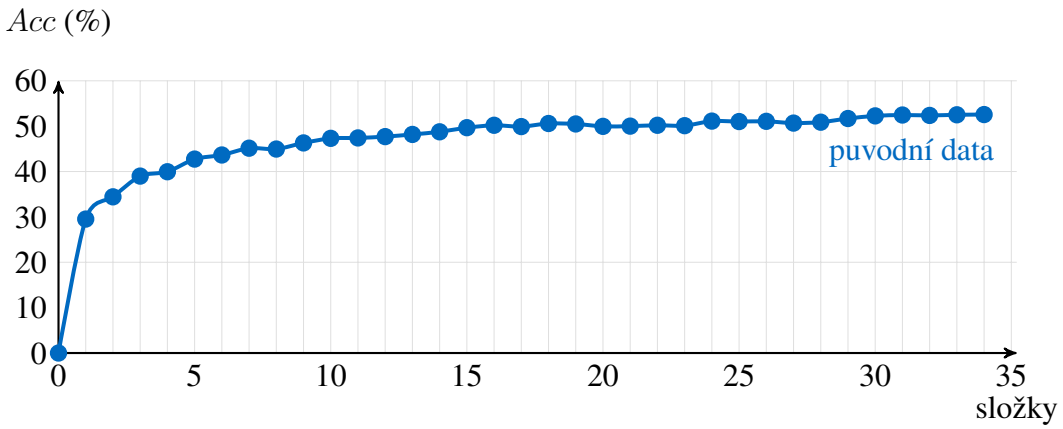
```
from htk import HTKToolkit

htk = HTKToolkit(r"HTK\exes", r"wave_out", r"HTK\scripts")
htk.create_param()
htk.create_mkphones()
htk.create_phones()
htk.parametrize()
htk.create_train_data(overwrite=True)
htk.create_proto()
htk.compv()
htk.reestimate(1, 2)
htk.reestimate(3, 4)
htk.create_sil()
htk.add_sp()
htk.reestimate(5, 7, step=1)
htk.align()
htk.reestimate(8, 11, step=1, aligned=True)
htk.result()
```

Název metody	Popis
`create_param()`	vytvoří soubor `param.scp`.
`create_mkphones()`	vytvoří soubory `mkphones` s příkazy.
`create_phones()`	vytvoří soubory `phones` s přepisem na úrovni fónů.
`parametrize()`	provede parametrizaci dat uvedených v `param.scp`.
`create_train_data()`	vytvoří sadu trénovacích a testovacích dat.
`create_proto()`	vytvoří soubor `proto`.
`compv()`	provede počáteční inicializaci středních hodnot a kovariancí všech HMM jednotlivých fónů.
`reestimate(i, j, N)`	provede reestimaci modelů od složky hmm_i do složky hmm_j se souborem `phones`N`.mlf`.
`create_sil()`	vytvoří soubor `sil.hed`.
`add_sp()`	přidá model pauzy `_sp_` do posledního reestimovaného souboru modelů.
`align()`	provede přerovnání dat.
`result()`	provede rozpoznávání testovacích dat s natrénovanými akustickými modely a vypíše výsledek do souboru `vysledek.txt`

Tab. 7.1: Popis metod třídy `HTKToolkit`.

7.3 Vyhodnocení úspěšnosti rozpoznávání s připravenými akustickými modely

Z obrázku 7.1 zobrazujícího úspěšnost rozpoznávání akustických modelů natrénovaných z původních dat je patrné, že s přibývajícím počtem složek (cca 15)se procentuální úspěšnost Acc zvyšuje velmi nepatrně (viz podkapitola 6.1.8, resp. rovnice (6.1)). Výčet přesných hodnot Acc pro vybraný počet složek je uveden v tabulce 7.2 ve sloupci "neopravená data".

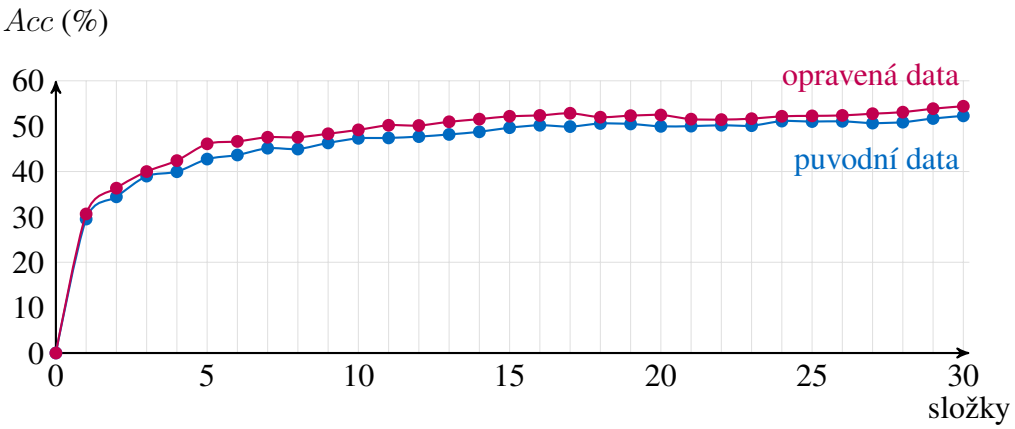


Obr. 7.1: Graf úspěšnosti rozpoznávání v závislosti na počtu složek jednotlivých HMM. Data byla získána z rozpoznávání s akustickými modely natrénovanými na neopravených datech.

	úspěšnost rozpoznávání [%]	
počet složek	**neopravená data**	**opravená data**
1	29.52	30.65
2	34.42	36.33
3	38.99	39.99
4	39.95	42.4
5	42.74	46.1
10	47.31	49.18
15	49.66	52.17
20	49.95	52.45
25	51.01	52.26
30	52.26	54.38

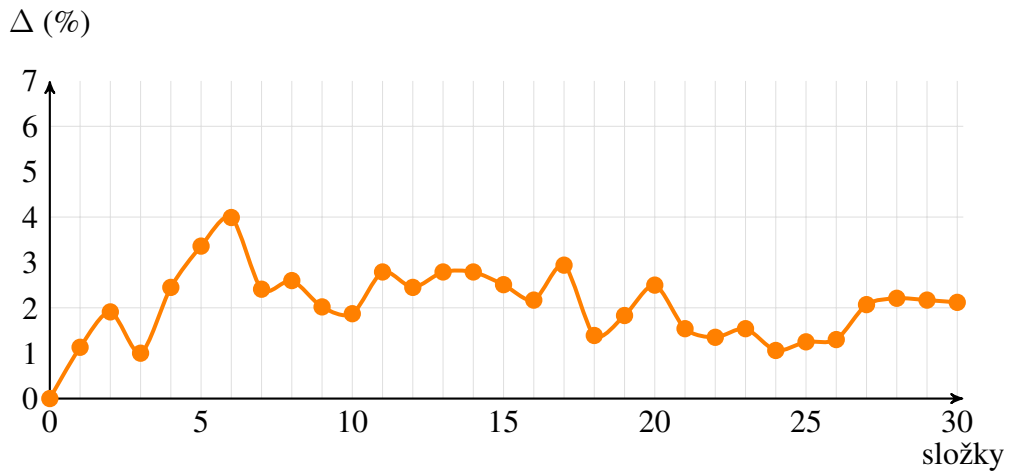
Tab. 7.2: Procentuální úspěšnost rozpoznávání akustických modelů v závislosti na počtu složek rozložení HMM jednotlivých fónů.

Na obrázku 7.2 je zobrazen vývoj procentuální úspěšnosti Acc rozpoznávání s akustickými modely natrénovanými z původních dat (modrá čára) a s akustickými modely natrénovanými z opravených dat. Přesné výsledky procentuální úspěšnosti rozpoznávání pro vybrané počty složek jsou uvedeny v tabulce 7.2. Úspěšnost rozpoznávání se u modelů se 30 složkami natrénovaných z původních dat zvýšila o 22.74% oproti modelům s jednou složkou. Jak je z grafu 7.2 patrné, úspěšnost rozpoznávání s modely se 30 složkami natrénovanými z opravených dat je vždy zhruba o 2% vyšší, než v případě modelů založených na neopravených datech. Úspěšnost rozpoznávání se u modelů se 30 složkami natrénovaných z opravených dat zvýšila o 23.73% oproti modelům s jednou složkou.



Obr. 7.2: Graf úspěšnosti rozpoznávání testovací sady s akustickými modely natrénovanými z původních a opravených anotačních dat.

Na obrázku 7.3 je vyobrazen rozdíl $\Delta Acc = m_c - m_o$ procentuální úspěšnosti rozpoznávání s akustickými modely založenými na opravených datech m_c a s modely založenými na původních datech m_o. Tento rozdíl se s přibývajícím počtem složek začíná ustalovat na hodnotě kolem 2%.



Obr. 7.3: Graf rozdílu procentuální úspěšnosti rozpoznávání mezi oběma sadami akustických modelů.

7.4 Test s jazykovým modelem

S natrénovanými akustickými modely se 30 složkami byl proveden test s jazykovým modelem reprezentovaný trigramy s více než 500 tisíci slov. Pro účely testu byla vybrána zvuková a anotační data záznamů tenisových utkání uvedených v tabulce 7.4. Zvuková data byla rozpoznávána souvisle, čímž bylo simulováno nasazení rozpoznávače v reálných podmínkách. N-gramové jazykové modely s počtem složek jsou uvedeny v tabulce 7.3. Test rozpoznávání

řád	n-gram	počet n-gramů
1	unigramy	524719
2	bigramy	22203818
3	trigramy	15010147

Tab. 7.3: Přehled n-gramových modelů s počtem jednotlivých n-gramů.

název zvukové nahrávky	délka [min]	počet slov
`FrenchOpen2014-140522.wav`	26	2977
`FrenchOpen2014-140527.wav`	51	4916
`tenis_zeny_CZE_ITA.wav`	61	3419
`tenis_zeny_GBR_RUS.wav`	30	1672

Tab. 7.4: Seznam testovacích nahrávek pro test s jazykovým modelem.

S daty uvedenými v tabulce 7.4 byl proveden experiment. Výsledky testu pro poslední dvě nahrávky uvedené v tabulce 7.4 jsou uvedeny níže:

Rozpoznávání s akustickým modelem natrénovaným z **původních** dat:

```
WORD: %Corr=81.45, Acc=75.05 [H=4146, D=227, S=717, I=326, N=5090]
```

Rozpoznávání s akustickým modelem natrénovaným z **opravených** dat:

```
WORD: %Corr=82.79, Acc=77.13 [H=4214, D=222, S=654, I=288, N=5090]
```

Z výsledků uvedených výše je patrné, že rozpoznávání s akustickým modelem natrénovaným na opravených datech poskytuje lepší procentuální úspěšnost rozpoznávání Acc než rozpoznávání s akustickým modelem natrénovaným z původních dat. S použitím trigramového jazykového modelu a akustického modelu se 30 složkami rozložení HMM bylo dosaženo celkového zlepšení 45.53 % o proti akustickému modelu s jednou složkou natrénovaného z původních dat. Stejným způsobem provedený test akustického modelu natrénovaného z opravených dat poskytuje zlepšení o 46.48%.

Kapitola 8

Závěr

Tato práce shrnuje základní principy a teorii statistického rozpoznávání řeči. Je zde kladen velký důraz na akustické modelování, neboť právě to tvořilo velkou část praktické části této práce. Dále je v této práci popsán postup tvorby anotací pomocí programu Transcriber, pravidla anotování a tvorba jazykových slovníků v programu LMEdit.

Vedle tvorby anotací jsou také analyzovány nejčastější chyby vyskytující se v anotacích, způsoby, jakými je možné tyto chyby detekovat a metody, kterými lze detekované chyby opravit.

Značná část práce je zaměřena na přípravu dat pro trénování akustických modelů pomocí nástroje HTK, což zahrnuje segmentaci zvukových nahrávek a tvorbu souboru s transkripcemi na úrovni slov. Tato problematika je následována detailně popsaným postupem trénování akustických modelů ve zmíněném nástroji HTK. V tomto postupu jsou uvedeny a popsány příkazy, jenž chronologicky po sobě představují celý postup trénování akustických modelů a rozpoznávání řeči pomocí nástroje HTK.

Na základě teoretických základů rozpoznávání řeči a obecného postupu trénování akustických modelů pomocí nástroje HTK byly vytvořeny a natrénovány sportovní akustické modely tenisových utkání. Celkem byly vytvořeny dva různé akustické modely. První model byl vytvořen z původních dat a reestimován se 30 složkami rozložení jednotlivých HMM. Procentuální úspěšnost rozpoznávání tohoto akustického modelu bez jazykového modelu pak v testu vyšla 52.26%. Druhý akustický model byl vytvořen z dat, která byla opravena pomocí slovníků. Tento model byl taktéž reestimován na počet 30 složek rozložení jednotlivých HMM. Procentuální úspěšnost rozpoznávání s druhým akustickým modelem bez jazykového modelu vycházela 54.38%, a tedy celkové zlepšení oproti rozpoznávání s akustickým modelem založeným na původních datech je 2.12%. 30 složek rozložení HMM modelů bylo zvoleno proto, neboť s dalším přidáváním složek se úspěšnost rozpoznávání měnila už jen v řádech desetin či setin procent.

S natrénovanými akustickými modely byl proveden test s jazykovým modelem. Tento test je podrobněji popsán v sekci 7.4. Procentuální úspěšnost rozpoznávání pro akustický model natrénovaný z původních dat vyšla ve výsledku testu 75.05%, u akustického modelu natrénovaného z opravených dat vyšla úspěšnost rozpoznávání 77.13%.

Literatura

[1] Trasncriber. *Transcriber, a tool for segmenting labeling and transcribing speech* [online]. [cit. 14. 8. 2016]. Dostupné z: `http://trans.sourceforge.net`

[2] Psutka, Josef et al. *Mluvíme s počítačem česky*. Vid. 1. Praha: academia, 2006, 746 s. Česká matice technická, roč. 111, č.spisu 502. ISBN 8020013091.

[3] S. Young et al.: *The HTK Book (for HTK Version 3.4)*. Cambridge, 2006, 359 s.

[4] Řezáček, Petr. *Automatická detekce anotačních chyb v TTS korpusech*. Plzeň, 2014. 42 s. Diplomová práce. Fakulta aplikovaných věd, ZČU.

[5] Matoušek, Jindřich; Tihelka, Daniel. *Annotation Errors Detection in TTS Corpora*. Interspeech, 2013. Francie: Lyon. s. 1511 - 1515.

[6] HTK Toolkit. *The Hidden Markov Model Toolkit (HTK)* [online]. [cit. 9. 8. 2016]. Dostupné z: `http://htk.eng.cam.ac.uk`

[7] Příprava jazykových modelů. *Úvod do praxe stínového řečníka* [online]. [cit 20.8.2016]. Skripta. Dostupné z: `http://www.kky.zcu.cz/uploads/courses/sru/SRU7.pdf`

[8] Wikipedia. *Viterbiho algoritmus* [online]. [cit. 20.8.2016]. Dostupné z: `https://cs.wikipedia.org/wiki/Viterbiho_algoritmus`

[9] Wikipedia. *Maximum a posteriori estimation* [online]. [cit. 21.8.2016]. Dostupné z: `https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation`

Seznam obrázků

Seznam tabulek