Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Plzeň, 2018 **Bc. Martin Jahn**

Prohlášení

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

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

V Plzni dne

……

Poděkování

Rád bych poděkoval vedoucímu této diplomové práce Ing. Mgr. Josefu Psutkovi, Ph.D. za jeho podporu, cenné rady a čas, který mi při řešení této práce věnoval.

Abstrakt

Tato diplomová práce se zabývá automatickou tvorbou akustických modelů z dat webu České televize. Začátek práce se věnuje teoretické stránce problematiky rozpoznávání řeči a představuje různé metody přístupu k tomuto problému. Dále se v práci nachází analýza dat webu ČT a je zde popsán navržený automatický modul pro extrakci dat a trénování akustických modelů. Ke konci této práce jsou zmíněny a vysvětleny metody úprav titulků za účelem zlepšení úspěšnosti rozpoznávání. Veškeré dosažené výsledky jsou přehledně zobrazeny v tabulkách a grafech.

Abstract

This thesis deals with an automatic creation of acoustic models from the Czech Television web. At the beginning, the author introduces the problematics of speech recognition and presents various methods of approaching this problem. There is also CT web data analysis and proposal of automatic module for data extraction and training of acoustic models in this work. At the end of this work, methods for improving the recognition success rate are commented and all the obtained results are clearly shown in the tables and graphs.

Klíčková slova

rozpoznávání řeči, akustický model, skryté Markovovy modely, skryté titulky, iVysílání, Česká televize

Keywords

speech recognition, acoustic model, Hidden Markov Models, closed captions, iVysilani, Czech Television

Obsah

Sezam zkratek

HTK	Hidden Markov Model Toolkit
MLF	Master Label File
HMM	Hidden Markov Model
MFCCs	Mel-Frequency Cepstral Coefficients
DNN	Deep Neural Network
TDNN	Time Delay Neural Network
EM	Expectation-Maximization
MAP	Maximum A Posteriori Probability
FFT	Fast Fourier Transform
IDFT	Inverse Discrete Fourier Transform
ML	Maximum Likelihood
OOV	Out Of Vocabulary
CMN	Cepstral Mean Normalization
ASR	Automatic Speech Recognition
AM	Akustický Model / Akustické Modely
ČT	Česká Televize

Úvod

Rozpoznávání mluvené řeči je velice komplexní a netriviální úloha. Rozpoznat, čili porozumět lidské promluvě, je občas obtížné i pro samotného člověka, a proto není divu, že se ještě nepodařilo vyvinout stroj, který by byl schopen rozpoznat lidskou promluvu bez chyby.

S problematikou rozpoznávání řeči, aniž bych si to uvědomoval, jsem se seznámil již během mého studia na gymnáziu. V té době jsem začal pro Katedru kybernetiky v Plzni přepisovat titulky ke sportovním pořadům České televize. V průběhu mého vysokoškolského studia, na již zmíněné katedře, jsem absolvoval řadu předmětů, které se touto problematikou zabývaly a velice mne zaujaly. Dále jsem si tuto problematiku vyzkoušel i v praxi, a to v podobě několika projektů a hlavně v bakalářské práci. To vše jsou důvody, proč jsem si téma diplomové práce vybral právě z této oblasti.

Tato diplomová práce se zabývá automatickou extrakcí dat z webových stránek ČT [1] za účelem tvorby akustických modelů pro zkvalitnění stávajícího systému ASR. Nejprve bude nutné zanalyzovat web ČT, tedy iVysílání a zjistit množství vhodných, čili použitelných dat pro trénování akustického modelu. Jako vhodná data budou použity takové pořady, které budou splňovat podmínky jako například: pořad bude obsahovat zvukovou stopu včetně jeho textového přepisu (titulků), pořad bude v českém jazyce, pořad bude v rozumné kvalitě, apod. Dále bude nutné navrhnout automatický modul, který ze vstupních pořadů natrénuje akustický model, kterým bude možno rozpoznávat mluvenou řeč. Jako vstupem bude tedy pouze textový soubor s internetovými adresami odkazujícími na dané pořady a výstupem budou zpracovaná data pro tvorbu AM a také samotný natrénovaný AM. Akustické modely se budou trénovat jak pomocí GMM v nástroji HTK, tak také prostřednictvím neuronové sítě TDNN v Kaldi. Dále bude pravděpodobně potřeba se vypořádat se skutečností, že ne ve všech případech budou titulky přesně odpovídat zvukové stopě - to znamená, že hranice jednotlivých titulků budou buď posunuty dopředu, nebo zpět, anebo v titulkách nebude uveden doslovný přepis zvukové stopy. Oba tyto aspekty budou určitě negativně ovlivňovat úspěšnost rozpoznávání a bude potřeba se s tímto problémem vypořádat. Závěr práce bude obsahovat jednotlivé návrhy a realizace řešení těchto problémů. Dosažené výsledky budou zaznamenány do tabulek a grafů, aby se ukázala účinnost jednotlivých navržených řešení.

1. Teoretický úvod

1.1 Řeč a její vznik

Mluvená řeč je základním a zároveň nedůležitějším prostředek komunikace, který slouží k přenosu informace mezi lidmi. Bohužel jsou mezi námi takoví lidé, kteří kvůli svému handicapu mohou komunikovat pouze omezeně anebo vůbec. Právě z těchto důvodů je v současné době kladen velký důraz na to, aby se stal počítač pro člověka rovnocenným partnerem v mluveném dialogu, anebo aby jej mohl zastoupit. Tato komplexní úloha sestává z dílčích úloh neboli modulů, jako je zpracování signálu, počítačová syntéza anebo právě automatické rozpoznávání řeči.

Lidská řeč vzniká ve spolupráci několika orgánů, které se dohromady označují jako řečové orgány. Základem je dechové ústrojí, které tvoří zdroj energie pro řeč. Tento signál putuje při výdechu z plic do hlasového ústrojí skrze hlasivky, které jsou uloženo v hrtanu. Pokud je člověk zticha, pak je hlasivková štěrbina odkryta a vzduch tím pádem prochází bez odporu. Pokud se ovšem hlasivky nacházejí v tzv. hlasovém postavení, kladou procházejícímu proudu vzduchu odpor a začínají kmitat. Při kmitání hlasivek je proud vzduchu periodicky rozdělován na množství hustšího a řidšího vzduchu. Tím vznikají vzduchové vlny, které jsou člověkem vnímány jako zvuk. Tento periodický proud vzduchu je označován jako základní tón a tvoří základ lidského hlasu. Velikost tohoto základního tónu se v závislosti na jedinci pohybuje v rozmezí asi 60 – 400 Hz[1]. Takto modifikovaný signál dále putuje do artikulačního ústrojí, které umožňuje tvorbu velkého množství zvuků, kterými je řeč charakterizována. Dochází zde tedy k finální úpravě zvukového signálu a tím pádem ke vzniku řeči.

[1] U mužů se hodnota základního tónu pohybuje mezi 80 – 160 Hz, u žen mezi 150 – 300 Hz a u dětí dokonce až mezi 200 – 600 Hz [2]

1.2 Problémy při automatickém rozpoznávání řeči

Rozpoznávání řeči představuje automatický převod mluvené řeči do textové podoby. Problematika rozpoznávání řeči má již za sebou sice dlouhou historii, ale ani v dnešní době neexistuje žádný stroj, který by byl schopen rozpoznávat bez chyby promluvu libovolného jedince užívajícího libovolná slova. Důvodů, proč tomu tak je, je hned několik.

- Odlišnost hlasů více řečníků – každý člověk má trochu jinak uzpůsobené hlasové ústrojí, tzn., že může mít jiný tvar nosní dutiny, ústní dutiny anebo jinou frekvenci kmitání hlasivek. Dále má také každý člověk jiné tempo mluvy a jinak artikuluje. Kvůli těmto odlišnostem nenajdeme na světě dvě osoby, které by mluvili naprosto identicky.
- Odlišnost hlasu jednoho řečníka – nejenom, že stejná promluva více řečníků není nikdy identická, ale také dokonce se liší i promluva jednoho řečníka v různých situacích. Řečový signál promluvy je pokaždé odlišný, pokud člověk stejnou promluvu řekne nahlas, potichu, rozčíleně nebo šeptem. Tyto všechny aspekty se mohou negativně promítnout na úspěšnosti rozpoznávání.
- Měnící se akustické pozadí – během promluvy je na pozadí každého řečového signálu přítomen nějaký šum. Tento šum se může v průběhu promluvy měnit (např. vstřelení gólu na hokejovém utkání) a může výrazně ztížit rozpoznávání užitečného, tj. řečového signálu anebo jej může dokonce úplně znemožnit.
- Složitost řešeného problému – rozpoznávání izolovaných slov (např. povely pro hlasové zadávání do navigace) je daleko jednodušší úloha než rozpoznávání souvislé promluvy, kdy rozpoznávací slovník čítá desetitisíce slov, mezi kterými musí vybrat to správné. Další ztížení této úlohy přichází ve formě spontánní řeči, kdy řečník používá tzv. vycpávková slova (hm, no, čili, jaksi, atd.), jednu promluvu začíná vícekrát, zakoktává se a velmi často používá nespisovné tvary nebo koncovky slov (kterej, dobrej, apod.).

2. Metody rozpoznávání řeči

V dnešní době se využívá dvou hlavních přístupů k rozpoznávání řeči. Je to pomocí statistického přístupu, který využívá k pravděpodobnostnímu ohodnocení skrytých Markovových modelů směsi Gaussovských funkcí (GMM) a dále pomocí neuronových sítí, kde jsou jednotlivé pravděpodobnosti generovány v rámci sítě. Trénování neuronových sítí je časově i výpočetně velice náročné a to je důvod, proč se neuronové sítě začaly hojně používat až v posledních letech s rozvojem výkonné výpočetní techniky.

2.1 Statistické metody rozpoznávání řeči

Základními bloky metody statistického rozpoznávání řeči jsou akustický procesor a lingvistický dekodér. Akustický procesor převádí řečový signál *W*, kde $W = \{w_1, w_2, w_3, \ldots w_N\}$ je posloupnost jednotlivých slov, na posloupnost vektorů příznaků $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \ldots \boldsymbol{o}_N\}$. To znamená, že každé slovo má svůj vektor příznaků, kterým je reprezentováno. Lingvistický dekodér potom převádí tyto vektory příznaků na řetězce slov *W'*. Dekódování je tedy vlastně maximalizace aposteriorní pravděpodobnosti, kde se využívá Bayessova pravidla

$$W' = \underset{\boldsymbol{w}}{argmax}\, P(W|\boldsymbol{O}) = \underset{\boldsymbol{w}}{argmax}\, \frac{P(\boldsymbol{O}|W)\, P(W)}{P(\boldsymbol{O})},$$

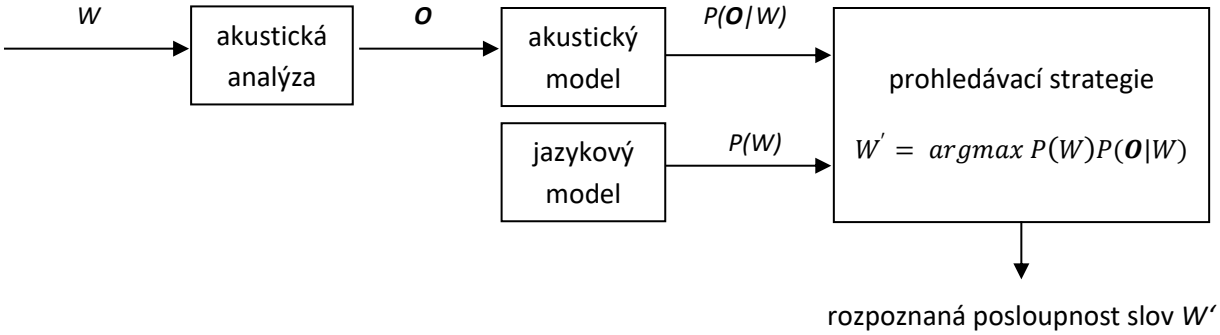
kde $P(\boldsymbol{O}|W)$ je aposteriorní pravděpodobnost, že při vyslovení slov *W* bude generována série vektorů příznaků $\boldsymbol{O}$, $P(W)$ je apriorní pravděpodobnost jednotlivých slov *W* a $P(\boldsymbol{O})$ je apriorní pravděpodobnost vektorů příznaků $\boldsymbol{O}$. Z důvodu toho, že pravděpodobnost $P(\boldsymbol{O})$ není funkcí *W*, tak ji můžeme z předchozího vzorce vypustit. Výsledný vzorec pro získání posloupnost slov *W'* lze tedy určit jako maximalizaci sdružené pravděpodobnosti $P(W, \boldsymbol{O})$

$$W' = \underset{\boldsymbol{w}}{argmax}\, P(W, \boldsymbol{O}) = \underset{\boldsymbol{w}}{argmax}\, P(W) P(\boldsymbol{O}|W)$$

Z uvedené rovnice vyplývá, že k získání rozpoznaného řetězce slov *W'* je zapotřebí dvou pravděpodobností, tj. $P(W)$ a $P(\boldsymbol{O}|W)$. Tyto pravděpodobnosti se dají získat odlišným způsobem nezávisle na sobě. Pravděpodobnost $P(W)$ udává informaci o jazykovém modelu a pravděpodobnost $P(\boldsymbol{O}|W)$ informuje o akustickém modelu. Kombinací těchto modelů lze potom dekódovat výslednou posloupnost slov *W'* [2].

Výsledná úloha se dá tedy rozdělit do čtyř základních kroků:

I. Akustická analýza – ze vstupního řečového signálu *W* vytvořit posloupnost vektorů příznaků ***O.***
II. Akustický model – na základě vstupních vektorů pozorování vytvořit akustický model pro získání aposteriorní pravděpodobnosti $P(\boldsymbol{O}|W)$.
III. Jazykový model – vytvořit jazykový model pro získání apriorní pravděpodobnosti $P(W)$.
IV. Prohledávací strategie – nalézt nejpravděpodobnější posloupnost slov na základě vstupních informací z akustického a jazykového modelu.



Obr. 2.1 Systému rozpoznávání řeči podle jednotlivých funkčních bloků [2]

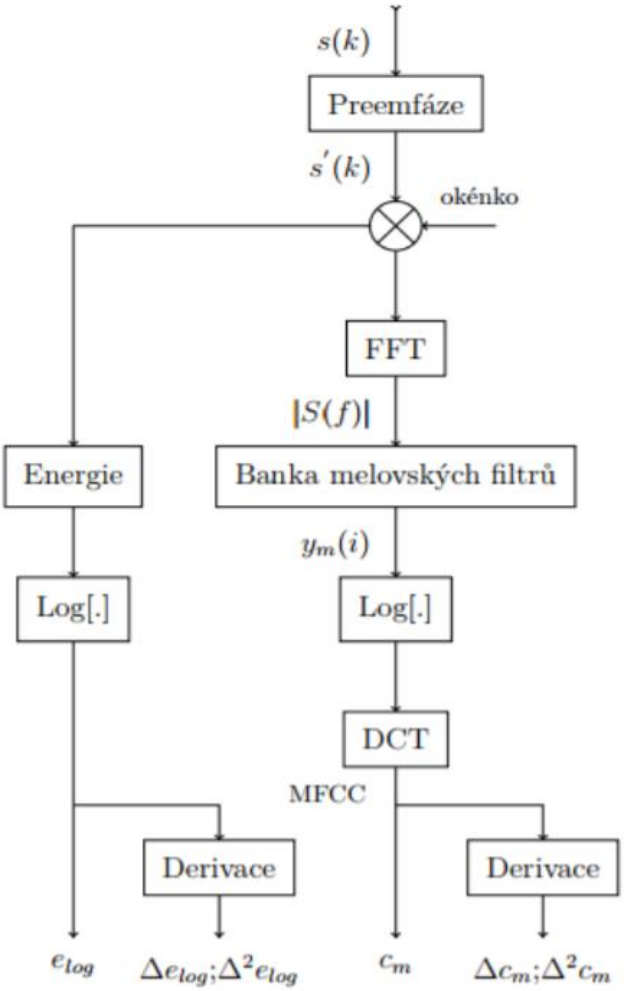
2.1.1 Akustická analýza

Jak již bylo výše zmíněno, akustická analýza převádí vstupní řečový signál na posloupnost vektorů příznaků ***O***. V úlohách rozpoznávání řeči se nejčastěji využívá metody tzv. Melovských kepstrálních koeficientů (MFCCs), která analyzuje signál jak ve frekvenční oblasti, tak i v oblasti časové. Tato metoda je navržena tak, aby respektovala nelineární vnímání zvuku lidským uchem, a to pomocí lineárního rozložení banky trojúhelníkových filtrů v tzv. melovské frekvenční škále, která je definována vztahem

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

kde *f* [Hz] je frekvence v lineární škále a f_m [mel] je frekvence v melovské škále [2].

Na vstup bloku akustické analýzy je přiveden řečový signál, který je rozdělen na mikrosegmenty o obvyklé délce 30 ms (viz kap. 2.1.2, druhý odstavec). Na tyto mikrosegmenty je dále aplikováno Hammingovo okénko, přitom je doporučeno toto okénko posouvat v časových úsecích 10 ms. Dále je signál převeden pomocí rychlé Fourierovy transformace (FFT) do spektra a aplikují se banky trojúhelníkových filtrů. Počet pásem banky filtrů se volí podle počtu kritických pásem s ohledem na vzorkovací frekvenci. Posledními kroky jsou výpočet logaritmu výstupu $y_m(i)$ jednotlivých filtrů a následná zpětná Fourierova transformace (IDFT). Počet výsledných koeficientů je možné volit menší než je počet kritických pásem. Obvykle se uvažuje 13 MFCCs koeficientů [2].



Obr. 2.2 Algoritmus výpočtu koeficientů MFCC [2]

Výše zmíněné koeficienty jsou statické a popisují řečový signál v daném okénku. K těmto koeficientům se dále přidávají dynamické koeficienty delta Δc_m a delta-delta $\Delta^2 c_m$. Tyto koeficienty vyjadřují dynamiku časové změny vektorů příznaků. Celkově je tedy každý analyzovaný mikrosegment popsán typicky 39 koeficienty MFCCs (3 * 13 MFCCs).

2.1.2 Akustický model

Akustický model je základním kamenem systému rozpoznávání řeči. Jak již bylo zmíněno, akustický model poskytuje co nejpřesnější a zároveň co nejrychlejší odhad aposteriorní pravděpodobnosti $P(\boldsymbol{O}|W)$. Jinými slovy říká, jaká posloupnost vektorů příznaků $\boldsymbol{O}$ je nejpravděpodobnější pro příchozí řetězce slov *W*. Akustické modely by měly splňovat tři základní podmínky, tj. přesnost, flexibilita a účinnost. Přesnosti se využívá k odlišení foneticky podobných slov s lingvisticky rozdílnými významy. Flexibilita je důležitá z toho důvodu, že při nasazení do provozu, bude rozpoznávač pracovat za jiných podmínek, než byl natrénován, tzn., bude muset rozpoznávat jiné hlasy na jiných akustických pozadích. A účinnosti je potřeba, aby mohl klasifikátor pracovat v reálném čase.

Při modelování řeči se u statistických metod využívá skrytého Markovova modelu. Tento přístup má za vzor princip generování řeči člověkem, kdy během promluvy je hlasové ústrojí v krátkých časových intervalech (mikrosegmentech) v jednom z ustálených stavů. Tyto krátké časové okamžiky závisí na rychlosti promluvy, ale obecně se uvažují o délce 30 ms. Během tohoto krátkého času je hlasovým ústrojím produkován signál, jehož parametry závisí na nastavení artikulačního ústrojí a tím vznikají různé zvuky (fonémy). Každá takováto slovní subjednotka má potom svůj vlastní skrytý Markovův model, kterým je modelována. Celá slova a celé promluvy nakonec vzniknou zřetězením těchto jednotlivých menších modelů.

2.1.2.1 Skyrtý Markovův v model

Skrytý Markovův model je konečný stochastický automat, který v jednotlivých diskrétních krocích generuje posloupnost vektorů příznaků $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \ldots \boldsymbol{o}_L\}$ a zároveň mění svůj stav s_i podle předem daných pravděpodobností přechodů a_{ij}. Změně stavu modelu se také jinak říká emitace. Pravděpodobnosti přechodů a_{ij} jsou podmíněny aktuálním stavem a vymezují veškeré pravděpodobnosti přechodů ze stavu s_i v čase *t* do stavu s_j v čase *t+1*.

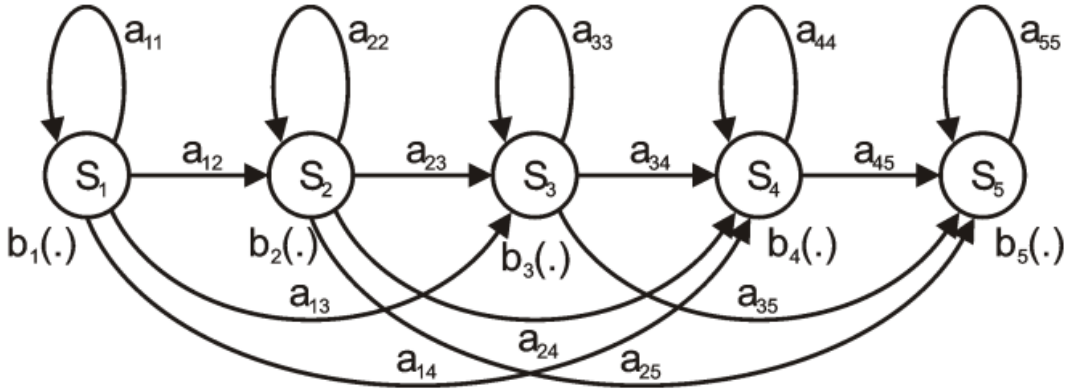
$$a_{ij} = P(s_j(t+1)|s_i(t))$$

Dále musí být také splněna podmínka, že součet všech pravděpodobnostních přechodů a_{ij} je roven jedné a to pro všechny stavy s_i, kde $i = 1, 2, \ldots, N$.

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

Při procesu modelování řeči se využívá skrytého Markovova modelu, respektive levo-pravého Markovova modelu, který výborně modeluje procesy vyvíjející se v čase. Model začíná v počátečním stavu příchodem prvního spektrálního příznaku a dále s rostoucím časem

přechází do nového stavu anebo v aktuálním stavu setrvává. Přechody do jednotlivých stavů jsou určeny pravděpodobností přechodů a_{ij}. Celý tento postup končí příchodem posledního spektrálního vzoru, kdy se model nachází v koncovém stavu.



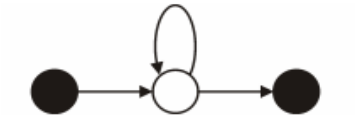
Obr. 2.3 Pětistavový skrytý Markovův model pro reprezentaci slova [2]

Dříve se pomocí levo-pravého Markovova modelu modelovala celá slova. Počet stavů modelu byl odvozen od průměrného počtu segmentů ve slově, tj. 40-60 stavů. Tento model prokázal velice dobré výsledky v rozpoznávání, ale byl výpočetně náročnější. Z tohoto důvodu došlo ke dvěma zásadním změnám, aniž by se dramaticky snížila přesnost rozpoznávání. Zaprvé, došlo ke snížení počtu stavů na pěti stavové modely. Zadruhé, pomocí levo-pravého Markovova modelu nejsou modelována celá slova najednou[2], ale jejich menší subslovní jednotky jako fonémy a hlavně trifóny, tj. fóny, které zohledňují levého a pravého souseda, čili kontext. Během trénování prokázaly trifóny daleko lepší výsledky než fonémy. Důvodem toho je, že v trifónových modelech je také zároveň zohledněna informace o koartikulaci, tj. informace o tom, jak předchozí foném ovlivňuje foném následující.

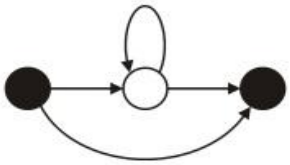
Monofonový kontext slovního spojení: „Dobrý den“
sil *d o b r I* ***sp*** *d e n* ***sil***

Trifonovy kontext slovního spojení: „Dobrý den“
sil *sil-d+o d-o+b o-b+r b-r+I r-I+d* ***sp*** *I-d+e d-e+n e-n+sil* ***sil***

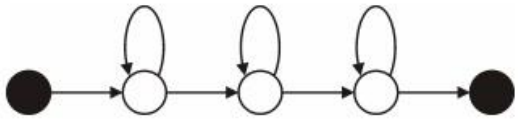
[2] Tímto došlo k dramatickému snížení paměťových nároků. Nebylo totiž potřeba trénovat všechna možná slova, ale stačilo trénovat pouze jednotlivé fonémy abecedy.



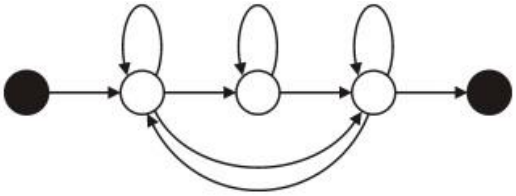
a) Model fonému s jedním emitujícím stavem



b) Model krátké mezislovní pauzy (sp)



c) Model fonému s třemi emitujícími stavy



d) Model pauzy na začátku a konci slova (sil)

Obr. 2.4 Různé struktury skrytých Markovových modelů fonémů [2]

2.1.2.2 Trénování skrytého Markovova modelu

Parametry skrytého Markovova modelu, se stanovují pomocí emitace, neboli trénování parametrů na základě známých, anotovaných datech, tj. trénovacích datech. Tyto parametry se nastavují pro každou subslovní jednotku modelu zvlášť. Jako nejčastěji používané subslovní jednotky jsou fóny nebo lépe trifóny. Parametry, které se při tomto procesu trénují, jsou pravděpodobnosti přechodů a_{ij} a parametry hustotních funkcí b_j, což v případě použití směsí normálního rozložení znamená střední hodnoty u_{jm}, kovariančí matice C_{jm} a váhy jednotlivých složek směsí c_{jm}. Souhrnně lze tedy hledané parametry zapsat ve tvaru $\lambda = \{a_{ij}, u_{jm}\ C_{jm}, c_{jm}\}$ [2].

Trénování parametrů skrytého Markovova modelu vychází z metody maximální věrohodnosti (ML). Tato metoda vychází z předpokladu tvaru pravděpodobnostního modelu $P(x|\lambda)$ s neznámými parametry λ, které se snaží najít pomocí trénovací sady (trénovacích vět) $x_1, x_2, \ldots, x_N$. Princip této metody využívá tzv. Fisherovu funkci věrohodnosti, která je dána vztahem

$$F(x_1, x_2, \ldots, x_N|\lambda) = \prod_{n=1}^{N} P(x_n|\lambda)$$

Hledá se tedy maximu funkce přes neznámé parametry λ

$$\hat{\lambda} = \underset{\lambda}{argmax}\ \prod_{n=1}^{N} P(x_n|\lambda)$$

V praxi se potom spíše využívá logaritmu této věrohodnostní funkce

$$\hat{\lambda} = \underset{\lambda}{argmax}\ log \prod_{n=1}^{N} P(x_n|\lambda)$$

Nastavení optimálních parametrů věrohodnostní funkce je velice složitá úloha, která v podstatě nemá explicitní řešení. Z tohoto důvodu se využívá velice efektivní numerické metody v podobě EM algoritmu. Tento algoritmus pracuje v pěti základních krocích:

1) Volba počátečních hodnot parametrů $\lambda = \lambda_0$
2) Spočtení podmíněné pravděpodobnosti všech dat, že na vstupu jsou parametry právě dané směsi.
3) Spočtení maximální věrohodnosti
4) Přepočtení pravděpodobností jednotlivých složek
5) Určení nových parametrů λ pro každou složku normálního rozložení

V úlohách rozpoznávání řeči se skrytými Markovovými modely se využívá tzv. Baum-Welchova algoritmu, což je speciální případ EM algoritmů. Velice detailní popis tohoto algoritmu lze nalézt v [2].

2.1.3 Jazykový model

Jazykový model tvoří další ze základních bloků systému rozpoznávání řeči. Úkolem akustického modelu je co nejrychleji a co možná s největší pravděpodobností poskytnout apriorní pravděpodobnost *P(W), a to* pro libovolnou posloupnost slov. Každý jazyk má svá pravidla a zákonitosti[3], jimiž se od ostatních jazyků liší. Jazykový model má za úkol tyto zákonitosti modelovat a tím určovat jistá omezení na možné posloupnosti slov *W*. Tyto omezení jsou v podstatě dvojího druhu, tj. deterministické a pravděpodobnostní omezení.

Deterministické omezení v praxi znamená, že nemůže být vysloveno takové slovo, které není obsaženo ve slovníku systému a nemůže být vysloveno jinak, než je uvedeno v odpovídajícím výslovnostním slovníku. Naproti tomu pravděpodobnostní omezení říká, jaká slova budou s největší pravděpodobností spojena do slovního spojení, čili jaké slovo bude za aktuálním následovat. Tato pravděpodobnost se velice liší situací, ve které se mluvčí nachází. To znamená, že při rozpoznávání fotbalového utkání by velice pravděpodobně za slovem *červená* následovalo slovo *karta*, zatímco při rozpoznávání pořadu o vaření by za tímto slovem nejspíše stálo slovo *řepa*.

[3] Mezi tyto zákonitosti patří slovník daného jazyka, pravidla pro tvorbu vět, apod.

Pravděpodobnost *P(W)* lze vyjádřit vztahem

$$P(W) = \prod_{k=1}^{K} P(w_k | w_{k-1} \dots w_n)$$

Tuto pravděpodobnost, tj. pravděpodobnost libovolné posloupnosti slov, však nelze dostatečně dobře odhadnout, protože při rozpoznávání řeči se pracuje s velice objemnými slovníky. V praxi se proto využívá aproximace této pravděpodobnosti. Tedy

$$P(W) \approx \prod_{k=1}^{K} P(w_k | w_{k-1} \dots w_{k-n+1}),$$

kdy všechny historie $w_{k-1},\dots,w_{k-n+1}$, které se shodují v posledních *n-1* slovech, jsou zařazeny do stejné třídy. Těmto modelům se říká n-gramové modely, kde n-gramem se rozumí posloupnost *n* za sebou jdoucích slov náhodného výběru. V reálných systémech jsou nejčastěji používány bigramy (*n* = 2) nebo trigramy (*n* = 3).

N-gramové modely mají svoje výhody, ale také i nevýhody. Tyto modely se velice hojně používají v jazycích, které mají relativně pevné pořadí slov ve větách. V takovýchto jazycích se tedy pořadí vybíraných slov dá poměrně snadno odhadnout. Naproti tomu největší nevýhodou n-gramových modelů je nedostatek trénovacích dat. Slovník, který by čítal 50 000 slov[4], tak by obsahoval 1,25 * 10^{14} trigramů, což je nepředstavitelné množství trénovacích dat, kterého nelze dosáhnout [2].

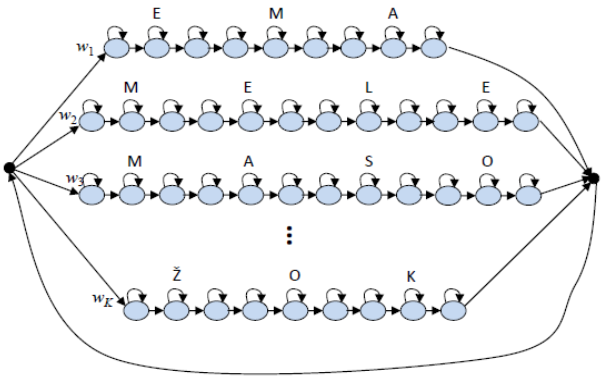
[4] Tento počet slov je běžně obsažen ve slovnících systémů rozpoznávání řeči

2.1.3 Prohledávací strategie – dekódování

V průběhu rozpoznávání je řeč zakódována do posloupnosti vektorů pozorování ***O***. Úkolem procesu dekódování je získat z posloupnosti pozorování ***O*** posloupnost slov *W'*. Při dekódování máme tedy k dispozici kromě posloupnosti pozorování ***O*** také ještě pravděpodobnosti *P(**O**|W)* a *P(W).* Dekodér se potom snaží na základě kritéria maximální aposteriorní pravděpodobnosti (MAP) najít posloupnost slov W', která nabývá maxima pro součin pravděpodobností *P(**O**|W)* a *P(W).*

$$W' = \underset{W}{argmax}\, P(W)P(\boldsymbol{O}|W)$$

Nejprve je potřeba vytvořit tzv. rozpoznávací síť, což je vlastně stavový prostor všech posloupností slov, které mohou být vydekódovány. Tato síť je konstruována na základě jazykového modelu, akustického modelu a slovníku. Nejjednodušší rozpoznávací sítí je síť lineární, kde jsou jednotlivá slova reprezentována lineárním zřetězením HMM fonémů.



Obr. 2.5 Lineární rozpoznávací síť [3]

Takováto rozpoznávací síť potom musí být dekódována, neboli musí se v ní najít posloupnost slov W'. Toho se dá v praxi dosáhnout dvojím přístupem, tj. dekódování podle kritéria MAP anebo dekódování podle Viterbiho kritéria. Oba dva přístupy jsou detailně popsány v [2].

Přesnost výsledného rozpoznávání se potom určuje podle Levensgteinovy vzdálenosti, která je definována vztahem

$$Acc = \frac{N - D - I - S}{N} * 100\%,$$

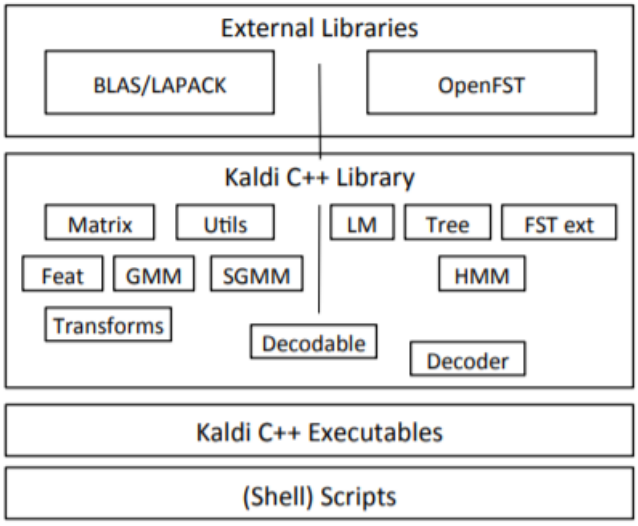
kde *N* je počet slov v textové promluvě, *D* je počet slov chybějících v rozpoznaném textu, *I* je počet slov, která jsou navíc v rozpoznaném textu a *S* je počet slov neshodujících se s daným textem [3].

2.2 Neuronové sítě

Neuronové sítě jsou inspirovány fungováním biologických neuronových sítí. Tyto sítě se skládají z neuronů, které jsou uskupeny do jednotlivých vrstev, které jsou navzájem propojeny a předávají si signál. Tento vstupní signál je v každém neuronu převeden na výstupní signál pomocí aktivační funkce.

Rozpoznávání řeči pomocí neuronových sítí je moderní metoda, která se začala využívat v několika posledních letech. Důvodem toho je, že tento přístup je velice výpočetně náročný a dřívější výpočetní technika prostě takto náročnou úlohu nebyla schopna zpracovat. Dnes již však tato technika existuje a hlavním důvodem, proč je tak populární a hojně využívaná je to, že rozpoznává řeč s vyšší úspěšností než klasický statistický přístup založený na HMM-GMM.

V úlohách rozpoznávání řeči se využívá k trénování neuronových sítí např. nástroje Kaldi. Kaldi je open-source nástroj, který je napsán v jazyce C++ a snaží se o to, aby se dal lehce pochopit, modifikovat a také rozšířit.



Obr. 2.6 Přehled různých částí nástroje Kaldi [4]

Základem tohoto nástroje jsou dvě externí knihovny, tj. OpenFST a BLAS/LAPACK. OpenFST je knihovna sloužící ke konstrukci vážených konečných automatů, které převádí vstupní řetězec *x* na výstupní řetězec *y* s nějakou váhou. BLAS/LAPACK je potom knihovna, která slouží k algebraickým operacím. V úloze akustického modelování je důležité, že Kaldi podporuje klasické modely gaussovských směsí (GMM), ale také je schopna vytvářet nové druhy modelů. Další velkou výhodou je to, že Kaldi obsahuje nástroje na snadné převedení klasických jazykových modelů ve formátu ARPA do formátu FST, který využívá [4].

3. Trénování akustických modelů pomocí HTK

Velice podrobný návod na trénování akustických modelů pomocí HTK byl již uveden v [5] anebo jej lze nalézt v [6]. Z tohoto důvodu zde sice budou uvedeny veškeré teoretické kroky vedoucí k natrénování akustického modelu, ale nebudou již detailněji vysvětleny.

3.1 Příprava k trénování

3.1.1 Vstupní soubory

Trénování akustických modelů vyžaduje několik druhů vstupních souborů a proto je nejprve nutné tyto soubory na základě stažených dat vytvořit. K trénování je tedy potřeba:

a) zvukové trénovací promluvy (ve formátu .wav)

b) *param.scp* - soubor promluv pro parametrizaci ve formátu:

/wav/veta001.wav /htk/veta001.htk
/wav/veta002.wav /htk/veta002.htk
…

kde vlevo se nachází cesta a název věty pro zprametrizování a vpravo potom cesta k uložení zparametrzovaného titulku spolu s jeho názvem. Zde se nachází jak trénovací promluvy tak také testovací.

c) *test.scp* - soubor testovacích, zparametrizovaných promluv ve formátu:

/htk/veta001.htk
/htk/veta002.htk
…

d) *train.scp* - soubor trénovacích, zparametrizovaných promluv ve formátu:

/htk/veta001.htk
/htk/veta002.htk
/htk/veta003.htk
/htk/veta004.htk
…

e) *words.mlf* - referenční soubor s přepisem všech promluv na úrovni slov ve formátu:

```
#!MLF!#
"*/veta001.lab"
měli
jste
jednoho
nepřítele
.
"*/veta002.lab"
zákazník
volající.
```

f) *dict_sp_* - slovník ve formátu:

```
a           a _sp_
absence     a p s e n c e _sp_
absolutní   a p s o l u t n I _sp_
…
žvýkání     Z v I k A N I _sp_
```

kde vlevo se nacházejí slova, která se objevila v promluvách a vpravo je potom jejich fonetická transkripce v podobě sekvence fonémů české fonetické abecedy.

g) *monophones0* a *monophones1* – soubory obsahující seznam fonémů české fonetické abecedy bez modelu krátké mezislovní pauzy *_sp_,* respektive s ní.

3.1.2 Tvorba souborů s přepisem na úrovni fonémů

Jak již bylo zmíněno výše, při trénování AM se nevyužívají celá slova, nýbrž jejich menší části jako slabiky, fón, trifóny, apod. Nejprve se tedy vytvoří referenční soubor, respektive dva referenční soubory všech promluv na úrovni fonémů, kde každý foném je reprezentován jedním skrytým Markovovým modelem (viz kapitola 2.1.2.1). Tyto dva referenční soubory se budou lišit tím, že jeden z nich nebude obsahovat model krátké mezislovní pauzy *_sp_* a druhý ano. Model krátké mezislovní pauzy se totiž využívání až dále v průběhu trénování a bude odvozen od modelu dlouhé pauzy *_sil_*.

Referenční soubory mají následující podobu:

phones0.mlf
#!MLF!#
"/veta001.lab"*
*_sil_*
m
N
e
l
i
s
t
e
j
e
d
n
o
h
o
n
e
p
R
I
t
e
l
e
*_sil_*
.
…

phones1.mlf
#!MLF!#
"/veta001.lab"*
*_sil_*
m
N
e
l
i
*_sp_*
s
t
e
*_sp_*
j
e
d
n
o
h
o
*_sp_*
n
e
p
R
I
t
e
l
e
*_sp_*
*_sil_*
.
…

Obr. 3.1 Struktura referenčních souborů

3.1.3 Parametrizace řečových dat

Při parametrizaci dochází k převodu zvukových nahrávek na posloupnost vektorů ***O***, které tuto nahrávku charakterizují (viz kapitola 2.1.1). V této práci se využívají MFCC koeficienty. Z důvodu menší časové náročnosti, je lepší provést parametrizaci před samotným trénováním než při jeho průběhu a k tomuto slouží program *HCopy*.

3.2 Tvorba monofonních modelů

3.2.1 Definice topologie HMM a jejich inicializace

Tvar skrytého Markovova modelu má v HTK následující tvar. Jedná se o pětistavový model se třemi emitujícími stavy.

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
0.0 0.0 0.0 ...
<Variance> 39
1.0 1.0 1.0 ...
<State> 3
<Mean> 39
0.0 0.0 ...
<Variance> 39
1.0 1.0 ...
<State> 4
<Mean> 39
0.0 0.0 ...
<Variance> 39
1.0 1.0 ...
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Obr. 3.2 Tvar skrytého Markovova modelu v HTK

kde parametr *<VecSize>* určuje délku vektoru příznaků ***O*** a jejich typ *(<MFCC_0_D_A>*). Dále za znakem *~h* následuje jméno modelu, sekvence <BeginHMM> uvozuje začátek, <NumStates> značí počet stavů, <Mean> a <Variance> určují střední hodnotu, respektive kovarianční matici daného stavu a <TransP> určuje matici přechodů. Takovýto skrytý model je samozřejmě nutné vytvořit neboli inicializovat pro všechny fonémy české abecedy.

Po provedení inicializace všech skrytých modelů je nutné tyto modely natrénovat na základě trénovacích dat. Trénování neboli reestimace se provádí pomocí programu *HERest* a pro dosažení lepších výsledků, je nutné reestimaci několikrát zopakovat.

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

Zde je uveden pouze příklad reestimace z adresáře hmm0 do adresáře hmm1. Tento příkaz je však nutné opakovat až do složky hmmB (viz opět [5]).

3.2.2 Úprava modelů pauz

V této chvíli je, za účelem lepšího modelování trénovacích dat, potřeba vytvořit model krátké mezislovní pauzy _*sp*_. Tento model bude mít velice podobnou strukturu jako již zmíněné modely a vznikne modifikací modelu dlouhé pauzy _sil_.

Model _*sil*_ :

~h "_sil_"
<BeginHMM>
<NumStates> 5
<State> 2
…
<State> 3
<Mean> 39
-4.74 2.88 –1.03 …
<Variance> 39
1.11 2.85 1.92 …
<GConst> 1.01
…
<TransP> 5
0.00 1.00 0.00 0.00 0.00
0.00 0.67 0.33 0.00 0.00
0.00 0.00 0.84 0.16 0.00
0.00 0.00 0.00 0.95 0.05
0.00 0.00 0.00 0.00 0.00
<EndHMM>

Model _sp_:

~h "_sp_"
<BeginHMM>
<NumStates> 3
<State> 2
<Mean> 39
-4.74 2.88 –1.03 …
<Variance> 39
1.11 2.85 1.92 …
<GConst> 1.01
<TransP> 3
0.00 1.00 0.00
0.00 0.84 0.16
0.00 0.00 0.00
<EndHMM>

Obr. 3.3 Struktura modelů dlouhé (*sil*) a krátké (*sp*) pauzy v HTK

3.2.3 Přerovnání trénovacích dat

Před samotným přerovnáním je potřeba upravit výslovnostní slovník do formátu, kdy každá výslovnostní varianta slova bude zvlášť pro model krátké pauzy a pro model dlouhé pauzy.

*a a _sp_*
*a a _sil_*
*absence a p s e n c e _sp_*
*absence a p s e n c e _sil_*
*absolutní a p s o l u t N I _sp_*
*absolutní a p s o l u t N I _sil_*
...

Při následném přerovnání dat dojde k tomu, že se vytvoří nový referenční soubor na úrovni fonémů, kde tyto foném znamenají nejpravděpodobnější výslovnost daného slova v daném kontextu.

*_sil_*
d
A
d
*_sp_*
i
*_sp_*
d
o
s
t
a
d
*_sp_*
d
U
v
j
e
r
u
*_sp_*

3.3 Tvorba trifónových modelů

Jak již bylo řečeno v kapitole 2.1.2.1, modelování promluvy na základě trifónů přináší daleko lepší přesnost rozpoznávání než při modelování řeči monofóny. Proto je velice vhodné převést monofónové modely v souboru *MODELS* (uložený v *hmmB*) na trifónové.

Nejprve je nutné vytvořit na základě referenčního souboru přepis promluvy na úrovni trifónů. Toho se dá docílit pomocí programu *Hled*.

*Hled -l * -n triphones0 -i crwtri.mlf mktri.led aligned.mlf > hled.txt*

Následně se na základě nejlepšího monofónového modelu vytvoří model na základě trifónů. To se provádí pomocí programu *HHed*.

HHed -T 1 -C CF_trif.mfc -H hmmB\models -M hmm_trif_0 mktri.hed monophones1 > hhed.txt

Takto získaný model je dále potřeba opět několikrát reestimovat.

Herest -T 1 -C CF_trif.mfc -I crwtri.mlf -t 250.0 150.0 1000.0 -S aligned.scp -H hmm_trif_0\models -M hmm_trif_1 triphones0

…

Herest -T 1 -C CF_trif.mfc -I crwtri.mlf -t 250.0 150.0 1000.0 -S aligned.scp -H hmm_trif_4\models -M hmm_trif_5 -s hmm_trif_5\stats triphones0

Jak již bylo zmíněno, největším problémem trifónů je nedostatek trénovacích dat. Tento problém se řeší slučováním stejných parametrů několika modelů. Proces slučování vychází z představit, že akusticky podobné trifóny lze brát v podstatě za ekvivalentní. Takovému trifónu se říká tzv. zobecněný trifón, který může reprezentovat celou třídu podobných trifónů.

V HTK se toto slučování dělá pomocí programu *HHed*.

HHed -C CF_trif.mfc -H hmm_trif_5\models -M hmm_trif_6 tree.hed triphones0 > hhed.txt

Na základě nastavení souboru *tree.hed* dojde ke sloučení trifónů do daného počtu tříd.

Takto získaný model se opět čtyřikrát reestimuje. Výsledný model je tedy uložen například v adresáři *hmm_trif_10*.

3.4 Přidávání složek

Pro dosažení lepšího modelování modelů trifónů a pro zlepšení úspěšnosti rozpoznávání je dobré do modelu přidat několik složek normálního rozložení. Složka se přidává pomocí následujícího příkazu:

HHEd -T 1 -A -C CF.mfc -H hmmB/models -M hmm_2_0 add_next.hed monophones1 > hmm_2_0\log

Po přidání složky je zapotřebí nově získaný model opět několikrát reestimovat. Nově získaný model je tedy připraven k rozpoznávání. Do modelu je možné přidat libovolný počet složek, avšak v určitém okamžiku se dosáhne maxima a úspěšnost rozpoznávání se přidáváním dalších složek nezlepšuje, respektive zlepšuje zanedbatelným způsobem.

4. Trénování akustických modelů pomocí KALDI

4.1 Vstupní data

Stejně jako při trénování akustických modelů pomocí HTK, tak také nástroj Kaldi vyžaduje vstupní data v určitém formátu.

a) *Param.txt* – soubor s cestami k jednotlivým zparametrizovaným souborům ve formátu:

H:/telefony/Siemens/A10000/A10000A0.htk
H:/telefony/Siemens/A10000/A10000A1.htk
H:/telefony/Siemens/A10000/A10000A2.htk
H:/telefony/Siemens/A10000/A10000A3.htk
...

b) *Dict.txt* – fonetický slovník ve formátu:

a a
absence a p s e n c e
absolutní a p s o l u t n I
…
žvýkání Z v I k A N I

c) *Train.txt* – referenční soubor jednotlivých promluv ve formátu:

A10000A0 ano
A10000A1 ne
A10000A2 dobrý den jak se máte
A10002C1 jedna nula osm pět šest
A10002C2 nula jedna šest čtyři devět jedna jedna šest pět pět,

kde vlevo je název věty (promluvy) a vpravo její obsah.

d) Train_utt2spk.txt – soubor s informací, která věta odpovídá jakému řečníkovi ve formátu:

A10000A0 A10000
A10000A1 A10000
A10000A2 A10000
A10000C1 A10000
A10000C2 A10000,

kde vlevo je název věty a vpravo je identifikační značka řečníka. V této práci se zpracovává velké množství nahrávek s velkým množství řečníků, kde označení všech řečníků by bylo velice časově náročné, a proto identifikační značka řečníka je v rámci jednoho pořadu stejná.

e) nonsilence_phones.txt – soubory obsahující seznam řečových fonémů české fonetické abecedy.

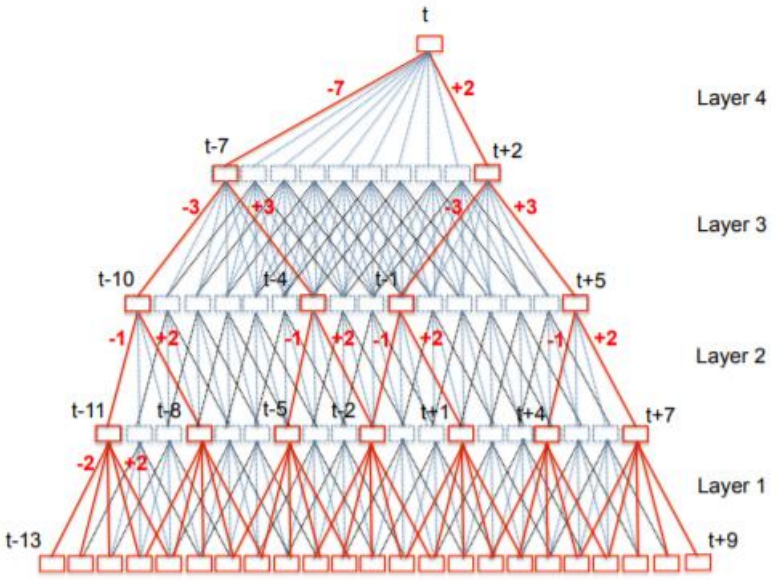
f) silence_phones.txt – soubor obsahující seznam neřečových fonémů (*sil* a *sp*).

4.2 Trénování

Při trénování neuronových sítí se vychází z již existujícího modelu, protože trénovat akustický model z tzv. flat startu je velice komplikované. Tento vstupní model se nejčastěji předtrénuje klasickým přístupem GMM-HMM pomocí HTK a tento model se přivede na vstup neuronové sítě. V úlohách rozpoznávání řeči se používá buď *DNN* a nebo častěji *TDNN*[5]. Podrobný popis těchto dvou sítí a rozdíl mezi nimi je popsán v [7]. Neuronová síť *TDNN* se trénuje pomocí učení s učitelem, kdy pro každý trénovací vzor je dopředu známo, do jaké třídy má být zařazen. Tato informace se následně po každém průchodu sítí porovná s informací o zařazení z neuronové sítě a na základě tohoto rozdílu dojde pomocí algoritmu back-propagation k přepočtu vah jednotlivých neuronů.

TDNN je neuronová síť, která se od klasické neuronové sítě liší ve dvou bodech. Zaprvé, na vstup neuronu (perceptronu) v aktuální vrstvě jsou přivedeny výstupy z více neuronů, tzv. kontextového okna (viz obrázek 4.1), předchozí vrstvy a ne klasicky pouze výstup z jednoho neuronu. Druhá odlišnost je v tom, že o finálním zařazení zpracovávaného vzorku rozhodují jak neurony z předchozích vrstev, tak také neurony z vrstev budoucích. To znamená, že neuronová síť si pro každý neuron, na základě kontextového okna, pamatuje jak jeho předchůdce, ale tak také jeho následovníky. Pro lepší pochopení tento postup zobrazuje následující obrázek.

[5] V této práci se také využívala neuronové sítě TDNN, která měla 5 vrstev s aktivační funkcí *relu* a výstupní vrstva byla *log-softmax*. V každé vrstvě se nacházelo 650 neuronů.



Obr. 4.1 Znázornění fungování TDNN [7]

Experimentálně bylo zjištěno, že nejlepších výsledků dosahuje neuronová síť s časovým krokem <-13,9> a s kontextovým oknem v první vrstvě <-2,2>, <-1,2> ve druhé vrstvě, <-3,3> ve třetí vrstvě, respektive <-7,2> ve vrstvě výstupní [7].

5. Praktická část

Katedra kybernetiky v Plzni již dlouhodobě spolupracuje s ČT na titulkování pořadů přemlouvaných stínovým řečníkem. K tomu, aby tento systém dobře fungoval, je zapotřebí kvalitních AM, které se musí natrénovat z velkého množství dat. ČT však disponuje rozsáhlým audiovizuálním archivem různých pořadů, tj. 94000[6] hodin přenosů a proto jsou zde tyto data použita za účelem zlepšení výše zmíněného systému.

Na začátku této kapitoly je nejprve provedena analýza dostupných dat na webu ČT. Další část řeší tvorbou systému pro automatický proces přípravy dat a trénování akustických modelů. Předposlední, ovšem neméně důležitá část, se zabývá navrženými metodami k synchronizaci titulků se zvukovou stopou a ke zlepšení úspěšnosti rozpoznávání. Závěrem jsou zhodnoceny veškeré dosažené výsledky.

5.1 Analýza dat

5.1.1 Analýza webu České televize (iVysílání)

Na webu iVysílání jsou jednotlivé pořady rozděleny podle žánru do dvanácti sekcí, tj. filmy, seriály, dokumenty, sport, hudba, atd. V této práci jsou analyzována a využita data ze všech sekcí kormě sekcí sport a hudba. Většina titulků ke sportovním pořadům byla přepsána Katedrou kybernetiky, a tudíž tato data jsou již „známá“ a nemá cenu je znovu používat. Hudební pořady naproti tomu obsahují velice malé množství mluveného textu, a proto jsou pro úlohu rozpoznávání řeči zcela nevodné.

Veškeré pořady v jednotlivých sekcích jsou seřazeny dle abecedy. Po rozkliknutí odkazu se otevře stránka s daným pořadem, kde ve spodní části se nachází tabulka s možnostmi jako: zapnout skryté titulky, přispět do diskuze, přejít na další díl, apod. Tyto možnosti se však liší pořad od pořadu, tzn., že ne všechny pořady mají skryté titulky, ne ke všem pořadů je vytvořena diskuze, apod. Pro účel této práce je však relevantní jediný z těchto aspektů, tj. jestli k danému pořadu existují skryté titulky. Pokud by k takovémuto pořadu totiž neexistoval jeho slovní přepis, tento pořad by se nedal využít k trénování AM.

Na základě analýzy byl zjištěn celkových počet pořadů dostupných na webu iVysílání. Toto číslo čítalo přibližně 6192 různých pořadů[7]. Tyto pořady byly následně rozděleny do 3 skupin.

1) Pořady zcela nevhodné pro trénování AM

[6] Toto číslo se váže k datu 15. 5. 2018

[7] Toto číslo se váže k datu 16. 2. 2018 a čítá pouze odlišné pořady, tj. nejsou v něm započítány různé díly stejného pořadu. Nicméně i tyto pořady (díly) byly zkoumány, zdali nejsou vhodné pro účel této úlohy. Celkový počet všech pořadů byl odhadem cca 35000.

2) Pořady, kterými by se dali trénovat AM
3) Pořady zcela vhodné pro trénování AM

V první skupině, jak již název napovídá, se nacházely všechny pořady, které se z nějakých důvodů nedaly použít jako trénovací data. Těchto klíčových důvodů byla hned celá řada: pořad nebyl v českém jazyce, obsah pořadu neodpovídal ve velké míře přepsaným titulkům, cizí jazyk na pozadí byl přemlouván češtinou, chyběly skryté titulky k pořadu, velice špatná kvalita pořadu (zejména staré pořady), otitulkovaná promluva v cizím jazyce nebo otitulkovaná hudba, atd.

Ve druhé skupině se nacházely pořady, které, sportovní terminologií řečeno, nebyly stoprocentní, a to ze dvou důvodů. Prvním problémem těchto pořadů bylo to, že řečový signál a tomu odpovídající titulky nebyly přesně časově sesynchronizovány. To znamená že, začátky promluv nezačínaly úplně přesně ve chvíli danou časovou značkou titulku, ale byly o pár sekund zpožděny, respektive v předstihu. Druhý důvod byl takový, že obsah titulek neodpovídal doslovně vyslovenému textu. Takovéto pořady by nebyly zcela vhodné pro trénování systému rozpoznávání řeči, avšak v kapitole 5.3 jsou diskutovány metody, které tyto problémy řeší a tím pádem umožňují použití těchto cenných dat v dané problematice.

Poslední skupinou byly ty pořady, ve kterých se nevyskytoval žádný z uvedených problémů, a tato data mohla být použita k akustickému modelování. Celkový počet použitých pořadů, tj. pořady z druhé a třetí skupiny, čítal 272 různých pořadů (celkem potom 1009 pořadů včetně jednotlivých dílů) v celkové časové délce 405 hodin. Vzhledem ke všem důvodům, které byly zmíněny výše, se tedy nelze divit tomu, že použitelných pořadů není více. Nadcházející tabulka detailněji popisuje daná data z webu ČT.

Žánr	***Počet pořadů***	***Z toho použitých***
Filmy	910	62
Seriály	257	36
Dokumenty	1890	55
Zábava	604	2
Děti a mládež	505	59
Vzdělávání	365	29
Zpravodajství	909	2
Publicistika	376	10
Magazíny	140	8
Náboženské	236	9

Obr. 5.1 Tabulka obsahující celkový počet pořadů na iVysílání a celkový počet použitých pořadů (v tabulce je každý pořad započítán pouze jdou, tzn., že různé díly jednoho pořadu nejsou započítány). Tato čísla se vážou k datu 16. 2. 2018.

5.1.2 Analýza formátu skrytých titulků

Formát skrytých titulků byl u všech pořadů identický. Každý titulek začínal uvozovací sekvencí tří čísel, tj. pořadí titulku se středníkem, začátek titulku a konec titulku (obojí v milisekundách). Na další řádce, respektive řádkách byl potom vlastní obsah titulku.

1; 11520 13200

Vítejte na Smetanově Litomyšli.

2; 13200 15560

Tak jako má mít ideální žena

svých pět P,

3; 15560 18960

má je i událost,

které věnujeme tento dokument.

4; 19600 20920

Navzdory ženskému rodu můžeme,

myslím,

5; 20920 25520

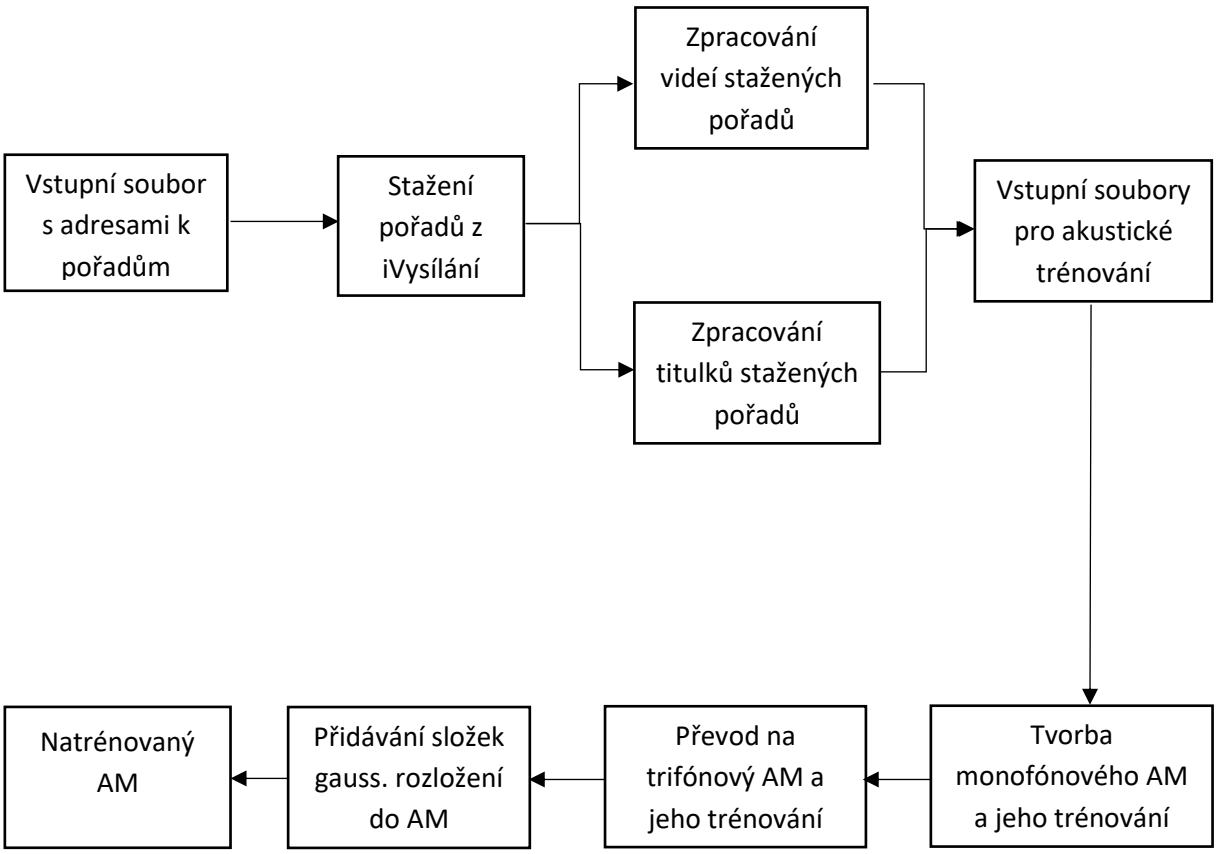
s klidným svědomím prozradit

její ročník 1949.

Obr. 5.2 Příklad struktury skrytých titulků na webu ČT

5.2 Tvorba automatického modulu

Základ automatického systému pro přípravu dat a trénování akustických modelů byl položen již v [5], kde jsou také detailně popsány jednotlivé kroky přípravy dat. Cílem tvorby automatického systému bylo ale to, aby na vstupu celého systému byl pouze soubor s odkazy na jednotlivé pořady a výstupem byl natrénovaný akustický model, připravený k rozpoznávání řeči. Idea navrženého a realizovaného systému je zobrazena na následujícím blokovém schématu.



Obr. 5.3 Schéma navrženého systému pro automatickou tvorbu akustických modelů

Data.txt

```
1  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/
2  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060041
3  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060040
4  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060039
5  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060038
6  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060037
7  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060036
8  http://www.ceskatelevize.cz/ivysilani/11032587292-novinky-z-prirody/215543112060035
```

Obr. 5.4 Příklad vstupního souboru

Tento systém byl vytvořen v prostředí Python[8], které se k tomuto úkolu dokonale hodilo, protože je velice přívětivé k práci s textovými soubory.

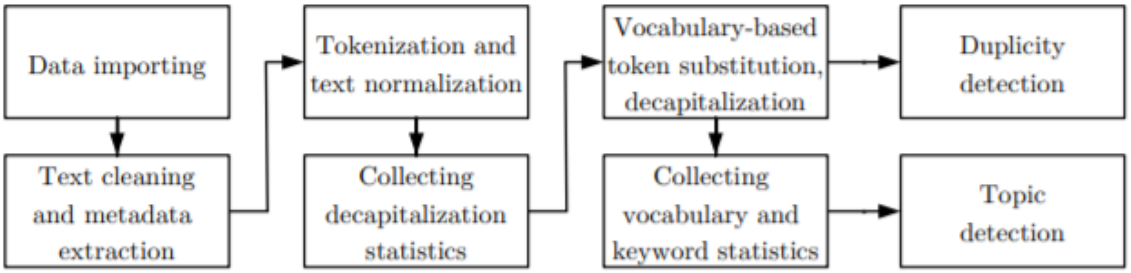
Celý systém se skládá z hlavního programu *data_preparing_run.py*, kterým se celý systém spouští a který využívá funkce z knihovny *data_preparing_library.py*. Tyto dva programy jsou uloženy v jednom adresáři dohromady s dalšími potřebnými nástroji.

- WaveCutter.exe - pro rozřezání celých pořadů na menší časti
- ffmpeg.exe - pro převod stažených *mpeg* souborů do *wav* formátu (16 kHz)
- get_video.py – program pro stažení pořadů z webu ČT
- remotejmzwFULL.py – program pro transport dat do databáze JMZW a zpět
- Data.txt – soubor obsahující adresy k pořadům
- Scripts a parsers – adresáře, které využívá *get_video.py*
- HTK – adresář obsahující HTK-programy pro tvorbu akustických modelů a dále skripty *htk.py*, *monototripho.py* a *add_components.py*.

Tento systém tedy stáhne, pomocí skriptu *get_video.py*, veškeré pořady uvedené ve vstupním souboru. Takto získaná data (skryté titulky a videa pořadů) jsou dále zpracována funkcemi v programu *data_preparing_library.py.* Tyto funkce vytvoří vstupní soubory pro akustické modelování (kap. 3.1) a jejich funkčnost je opět vysvětlena v [5].

Velice zajímavým nástrojem na úpravu textových dat je systém JMZW. Tento systém byl vyvinut za účelem úpravy dat pro Jazykové Modelování Z různých Webových stránek. Jeho základem je SQL databáze a řada algoritmů zpracovávající text. Vstupní data jsou zpracována v několika krocích. Nejprve jsou textové soubory pomocí skriptu *remotejmzwFULL.py* importovány do databáze, kde jsou následně „vyčištěny“, tzn., že jsou odstraněny všechny neřečové znaky (např. tagy v *html* dokumentu, apod.). Data jsou dále normalizována, tzn., všechny číslovky jsou převedeny do textové podoby, jsou rozvinuty některé zkratky, první písmeno všech slov je převedeno na malé písmeno (dekapitalizace), apod. Nakonec je ke všem slovům, buď na základě fonetického slovníku anebo pomocí pravidel, vytvořena jejich fonetická transkripce. Výstupem tohoto nástroje je tedy „vyčištěný“ text a slovník unikátních slov s jejich fonetickou transkripcí [8], [9], [10].

[8] Přesněji řečeno Python 2.7



Obr. 5.5 Schéma struktury nástroje JMZW [8]

V tomto okamžiku jsou vytvořeny veškeré vstupní soubory potřebné k trénování akustického modelu. Hlavním programem jsou potom dále spuštěny ještě další tři podprogramy, tj. *htk.py*, *monototripho.py* a *add_components.py*. Program *htk.py* vytvoří a natrénuje na základě vstupních souborů (*txt* soubory a *wav* soubory) a pomocí softwaru HTK monofónový akustický model (viz kap. 3.2). Tento model se potom pomocí programu *monototripho.py*, a opět využití softwaru HTK, převede na trifónový akustický model a natrénuje se (viz kap. 3.3). Do takto získaného modelu jsou následně programem *add_components.py* přidávány jednotlivé složky normálního rozložení a s každou přidanou složkou se sleduje zvýšení míry rozpoznávání. Po přidání každé druhé složky se provede „cvičné“ rozpoznání testovací promluvy. Testovací promluva je zde vybrána jako podmnožina trénovací sady, tj. čtyři titulky z každého trénovacího pořadu, kde tato sada samozřejmě není použita v procesu trénování. Pokud se další přidanou složkou úspěšnost rozpoznávání již dále výrazněji nezvyšuje, je celý proces ukončen a získaný akustický model je připraven pro používání v reálných úlohách rozpoznávání řeči.

5.3 Metody sloužící ke zlepšení úspěšnosti rozpoznávání

V kapitole 5.1.1 byla analyzována data na webu ČT. Tato data byla následně podle kvality rozdělena do tří skupin. Tato kapitola se zabývá daty z druhé skupiny, tj. daty, která budou moci být po určité úpravě použita k akustickému modelování.[9]

Jak již bylo výše zmíněno, v této skupině dat byly přítomny dva klíčové problémy. První z nich je fakt, že skryté titulky pořadů neodpovídaly časově jejich zvukové stopě. Po následujícím rozřezání pořadů na menší části tedy došlo k tomu, že slova patřící do aktuálního titulku (věty) byla přítomna až v titulku následujícím a naopak slova z předchozí věty se objevila ve větě aktuální. Takovéto titulky by tedy přispívaly minimálním způsobem

[9] Tyto metody byly navrženy pro úpravu nedokonalých titulků, avšak ve snaze získání kvalitního akustického modelu, byly těmito metodami upraveny pro jistotu také titulky z první skupiny.

k trénování akustických modelů nebo by dokonce mohly být z procesu trénování vyřazeny[10]. Druhý problémem bylo to, že titulky občas neodpovídaly doslova vyslovené promluvě. Následující navržené metody se snaží tyto problémy eliminovat.

5.3.1 Metoda prodlužování titulků

Tato metoda vychází z idey, že čím méně hranic titulků je v pořadu obsaženo, tím méně nesprávných slov se v určitém titulku objeví. Jinými slovy, snaží se udělat pořady co možná nejvíce spojité v rámci zachování obvyklého počtu slov ve větě. Tato metoda byla nejprve vyzkoušena spojením tří po sobě jdoucích titulků, následně šesti, respektive devíti titulků.

Původní titulky:

*1; **87720** 88960*
Ano, máte pravdu,

2; 88960 91880
toto datum do termínu konání
festivalu nezapadá,

*3; 91880 **94920***
ale právě proto jsme se
do Litomyšle vypravili.

Příklad spojení tří titulků:

*1; **87720 94920***
Ano, máte pravdu, toto datum do termínu konání festivalu nezapadá, ale právě proto jsme se do Litomyšle vypravili.

Takto upravené titulky vznikly tím způsobem, že se uložila časová značka určující začátek promluvy prvního titulku a časová značka určující konec titulku třetího. Na základě těchto časových značek se vytvořily hranice nového titulku, který obsahoval promluvu všech tří předešlých částí. Tímto způsobem se pomocí programu na prodloužení titulků *new_longer_titles.py* zpracovala veškerá použitá data a tato data se následně přivedla zpátky na vstup celého systému přípravy dat a trénování AM.

[10] Zarovnání dat (viz kapitola 4.1.3).

5.3.2 Metoda zarovnání nedokonalého textu

Při řešení úloh, jako je rozpoznávání řeči, hraje obrovskou roli množství trénovacích dat. Čím více dat je totiž k dispozici, tím se stává rozpoznávač robustní a pracuje lépe při rozpoznávání nových neznámých hlasů. Právě toto ale bývá největším kamenem úrazu, protože získat velké množství kvalitních dat (kvalitní nahrávka s přesným přepisem) je velice obtížné a také časově a finančně náročné. Z tohoto důvodu se velice často využívají nedokonalá data, která jsou dostupná ve velkém množství jako např. skryté titulky, různé abstrakty apod. Právě v těchto případech je tedy nutné použití nějakého nástroje, který by časově zarovnal referenční text se zvukovou stopou a zároveň by vypustil všechna slova, která se v přepise sice nachází, ale nikdy nebyla vyslovena.

V této práci se využívá tzv. fónového zarovnávače. Tento zarovnávač funguje, velice podobně jako v [11], na principu dynamického programování, kde se porovnává rozdíl mezi automatickým fonémovým přepisem vstupního řečového signálu a fonémovým přepisem vstupního textu, tj. skrytých titulků. Jako míra rozdílu mezi těmito dvěma přepisy se udává upravená Levensgteinova vzdálenost. Každý foném je potom penalizován každou změnou pozice a takto nasčítaná pokuta je potom sečtena pro celé výsledné slovo (viz níže formát zarovnaného textu).

Metod na zarovnání nedokonalého textu existuje celá řada, například v [12] se využívá kombinace „driven decoding“ algoritmu a „spotting text island“ algoritmu. Tento přístup vlastně zarovnává nedokonalý text při samotném dekódování. Na základě primárního dekódování se získají počáteční hypotézy. Tyto hypotézy se následně porovnají, respektive zarovnají pomocí „spotting text island“ algoritmu porovnáním hypotéz se segmenty v obrovské databázi. Takto nově vzniklý zarovnaný text se následně opět dekóduje.

Jako další se tedy vyzkoušelo fonémové zarovnání vstupních skrytých titulků. Vstupem této metody byly pouze dva soubory, tj. seznam slov v pořadí promluvy a nahrávka této promluvy.

- Formát seznamu slov promluvy

PAUSA pt=#
pětice pt=pjetice
PAUSA pt=#
p pt=pE
PAUSA pt=#
která pt=kterA
PAUSA pt=#
Hodláme pt=hodlAme

V prvním sloupci byla pod sebou slova v pořadí, v jakém byla v nahrávce vyslovena. Druhý sloupec vždy začínal uvozovacím znakem *„pt=“*, za kterým následovala samotná

výslovnost slova. Mezi každé slovo byla přidána pauza, to bylo z důvodů toho, aby si mohl zarovnávač vybrat, jestli se mezi slovy opravdu pauza vyskytovala nebo ne a aby došlo k odstranění nevyslovených slov. Tím se dospělo k přesnějšímu určení začátků a konců jednotlivých slov.

- Formát nahrávek pořadů

  Nahrávky pořadů byly ve formátu *.wav* o frekvenci 16 kHZ.

Samotné přerovnání se potom provedlo pomocí fonémového zarovnávače, který byl vyvinut na Katedře kybernetiky v Plzni. Přerovnání se provedlo pomocí příkazu

ToolsExample.exe pokus.wav vocab.txt output.txt ,

kde *pokus.wav* je vstupní nahrávka, *vocab.txt* je vstupní soubor a *output.txt* je soubor se zarovnanými slovy.

Výsledkem zarovnávače byl textový soubor ve formátu

13.00	*13.04*	*1.00*	*PAUSA*
13.04	*13.22*	*1.00*	*tak*
13.22	*13.22*	*1.00*	*PAUSA*
13.22	*13.39*	*1.00*	*jako*
13.39	*13.39*	*1.00*	*PAUSA*
13.39	*13.56*	*1.00*	*má*
13.56	*13.56*	*1.00*	*PAUSA*
13.56	*13.76*	*0.50*	*mít*
13.76	*13.76*	*1.00*	*PAUSA*
13.81	*14.32*	*0.75*	*ideální*
14.32	*14.32*	*1.00*	*PAUSA*
14.32	*14.61*	*1.00*	*žena*
14.61	*14.61*	*1.00*	*PAUSA*
14.61	*14.94*	*0.44*	*svých*
14.94	*14.94*	*1.00*	*PAUSA*

První a druhý sloupec značí začátek, respektive konec daného slova v sekundách. Ve třetím sloupci se nachází konstanta, která vypovídá o tom, s jakou mírou důvěry se dané slovo v promluvě opravdu vyskytlo[11]. Poslední sloupec obsahuje samotná slova.

[11] Tato konstanta se pohybovala v rozmezí 1.0 (slovo tam určitě patří) až do mínusových hodnot. (To záleželo na množství nasbíraných penalt).

Takto získaná data bylo opět potřeba převést do klasického formátu titulků a to na základě následujících pravidel.

i. Pokud má slovo nižší míru důvěry než 0.5, tak se slovo v promluvě pravděpodobně nevyskytlo a bude ignorováno[12].
ii. Pokud hranice začátku a konce „slova“ *PAUSA* je stejná, tak pauza mezi slovy nenastala a bude ignorována.
iii. Pokud se časová značka konce aktuálního slova nerovná s časovou značkou začátku slova následujícího anebo pokud se již v aktuálním titulku nachází 15 slov, tak se ukončí aktuální titulek a začne nový[13].

Na základě těchto pravidel vznikly pomocí skriptu *new_alig_titles.py* nové titulky v požadovaném formátu a opět se z nich natrénoval akustický model.

V průběhu trénování se však přišlo na to, že nějaká slova na konci zarovnaných titulků byla lehce uříznuta, tzn., že nahrávka daného titulku neobsahovala celé závěrečné slovo, ale pouze jeho část. Z tohoto důvodu byly pro porovnání vytvořena ještě jedna sada titulků, tentokrát však bez daného posledního slova každého titulku. Začátky všech trénovacích titulků byly bezproblémové, protože začínaly v mezislovní pauze, která se před každé slovo přidala aditivně (viz formát vstupních dat do zarovnávače), a proto se nezkoušely další pokusy, jako například vypouštět také první slovo apod.

Úspěšnost těchto dvou přístupů, stejně jako výsledky všech ostatních zmíněných metod jsou diskutovány v následující kapitole.

[12] Na základě experimentů s různými prahy důvěry se dospělo k závěru, že hranice 0.5 je pro řešení této úlohy nejlepší.
[13] Tento postup má opět za cíl udržet titulky co možná nejvíce spojité, avšak, v rámci udržení přirozenosti, ne delší než je průměrný počet slov v české větě (viz [13]).

5.4 Diskuze dosažených výsledků

Strategie pro experimentování s navrženými metodami byla následující. Po analýze a získání všech trénovacích dat, se náhodně vytvořila podmnožina trénovací sady, která čítala 150 pořadů o délce 82 hodin. Z této trénovací sady se na základě výše zmíněných přístupů vytvořily akustické modely, které se natrénovaly v prostředí HTK (GMM modely). V této fázi bylo zapotřebí zjistit, která metoda zarovnání bude nejlepší a ne jaké úspěšnosti rozpoznání se dosáhne, a proto se pro prvotní experimenty zvolila pouze část trénovací sady a akustické modely se natrénovaly v HTK. Pokud by se již od prvních pokusů využívalo celé trénovací sady, bylo by to zbytečně časově náročné a získané výsledky by byly de facto stejné. Pro každou metodu se dále natrénoval trifónový akustický model, do kterého se následně přidávaly složky normálního rozložení. Aby se pro každou metodu zjistil optimální počet stavů a složek, provedlo se po každém druhém přidání složky cvičné rozpoznání a pokud se dva po sobě získané výsledky lišily pouze v rámci půl procenta, byl celý trénovací proces ukončen a model s daným počtem složek se považoval za optimální (počet složek akustických modelů se u všech metod pohyboval v rozmezí 15 – 20 složek). Jako testovací sada se zvolila jedna věta z každého pořadu, tedy 75 promluv, které nebyly součástí trénovací sady.

Na základě takto získaných akustických modelů jednotlivých metod bylo zapotřebí zjistit, jaká navržená metoda je nejlepší, tzn. přidat tyto akustické modely do stávajícího systému ASR a rozpoznat testovací pořady. Testovací sada obsahovala 6 náhodně vybraných pořadů z trénovací sady o délce 2 hodiny a 36 minut a 14459 slov, kde tyto pořady opět samozřejmě nebyly použity v procesu trénování. Jediný požadavek pro zajištění objektivity výsledků byl takový, aby trénovací sada byla vyvážená, tj. aby půlka trénovací sady obsahovala pořady s jedním řečníkem a druhá půlky pořady pro více řečníků. V rámci systému ASR byl k rozpoznání dále použit trigramový obecný jazykový model, který čítal 1.3 milionů slov, kde míra OOV[14] byla 0.025 % a dále dekodér, využívající CMN[15] a silence detektor.[16]

Jako první se do systému ASR přidal akustický model natrénovaný z originálních, původních, nijak neupravených titulků. Podle očekávání byla míra úspěšnosti rozpoznávání velice nízká, pouhých 65,42 %. Dále se proto vyzkoušela metoda spojování, respektive prodlužování titulků. Nejprve se provedlo test s akustickým modelem natrénovaným spojením tří po sobě jdoucích promluv. Úspěšnost rozpoznávání stoupla výrazně na 74,36 %. Z výsledku je patrné, že opravdu došlo ke snížení počtu nesprávných slov ve větě a tím také ke zlepšení úspěšnosti rozpoznávání. Z důvodu funkčnosti této metody se ještě dále provedlo rozpoznávání pomocí akustických modelů natrénovaných spojením šesti, respektive devíti titulků. V těchto případech byla výsledná úspěšnost rozpoznání 74,75 %, respektive 73,63 %.

[14] OOV vyjadřuje míru přidaných slov oproti jazykovému modelu. Jinými slovy, jedná se o slova testovací sady, která nebyla přítomna v jazykovém modelu.

[15] CMN je normalizační metoda, která průměruje proměnné charakteristiky kanálu. Využívá se při parametrizaci. Podrobné informace lze nalézt v [2].

[16] Jedná se o nástroj, který informuje rozpoznávač o tom, jestli zvuková stopa v daném okamžiku obsahuje užitečný signál a má rozpoznávat anebo jestli se jedná o šum a rozpoznávání má být zastaveno.

Z výsledků je jasné, že v případě spojení šesti titulků došlo pouze k mírnému zlepšení úspěšnosti, a když navíc spojení devíti titulků přineslo snížení, nezkoušelo se již další prodlužování.

Dále byla vyzkoušena metoda zarovnání titulků fonémovým zarovnávačem. Jak již bylo zmíněno výše, v této metodě se využívalo dvou přístupů úpravy titulků, tj. titulky se všemi zarovnanými slovy ve větě a titulky bez posledního slova. Nejprve se vyzkoušel přístup ponechání všech zarovnaných slov ve větě, kde došlo k nárůstu úspěšnosti rozpoznání testovací sady na 76,30 %. Velkou výhodou tohoto přístupu je to, že tento zarovnávač, nejenže zarovná časově každé jednotlivé slovo, ale také udává informaci o tom, jaká slova pravděpodobně nebyla vyslovena a tím se tato „falešná“ slova nemohou dostat do procesu trénování. Tato skutečnost se samozřejmě musí promítnout v nárůstu úspěšnosti rozpoznávání. Po dosažení tohoto optimistického výsledku se ještě vyzkoušel druhý přístup, tj. otestování akustického modelu natrénovaného bez posledního slova ve větě. Tímto přístupem se dosáhlo dokonce ještě lepší úspěšnosti rozpoznávání, tj. 77,65 %. Při krátkém zamyšlení dává tento výsledek smysl, protože všechna „uříznutá“ koncová slova nemohla přispívat k dobrému trénování, ba dokonce je možné, že trénování negativně ovlivňovala, proto po odstranění těchto slov došlo opět ke zlepšení úspěšnosti rozpoznávání.

Použitá metoda	Počet složek	Počet stavů	Úspěšnost rozpoznání - Acc [%]
Originální titulky	15	4100	**65,42**
Spojení 3 titulků	20	4300	**74,36**
Spojení 6 titulků	20	4400	**74,75**
Spojení 9 titulků	20	4400	**73,63**
Fonémový zarovnávač – výběr všech slov	20	4100	**76,30**
Fonémový zarovnávač – odstranění posledního slova	20	4500	**77,65**

Obr. 5.6 Tabulka úspěšnosti rozpoznávání v závislosti na použité metodě přípravy titulků.

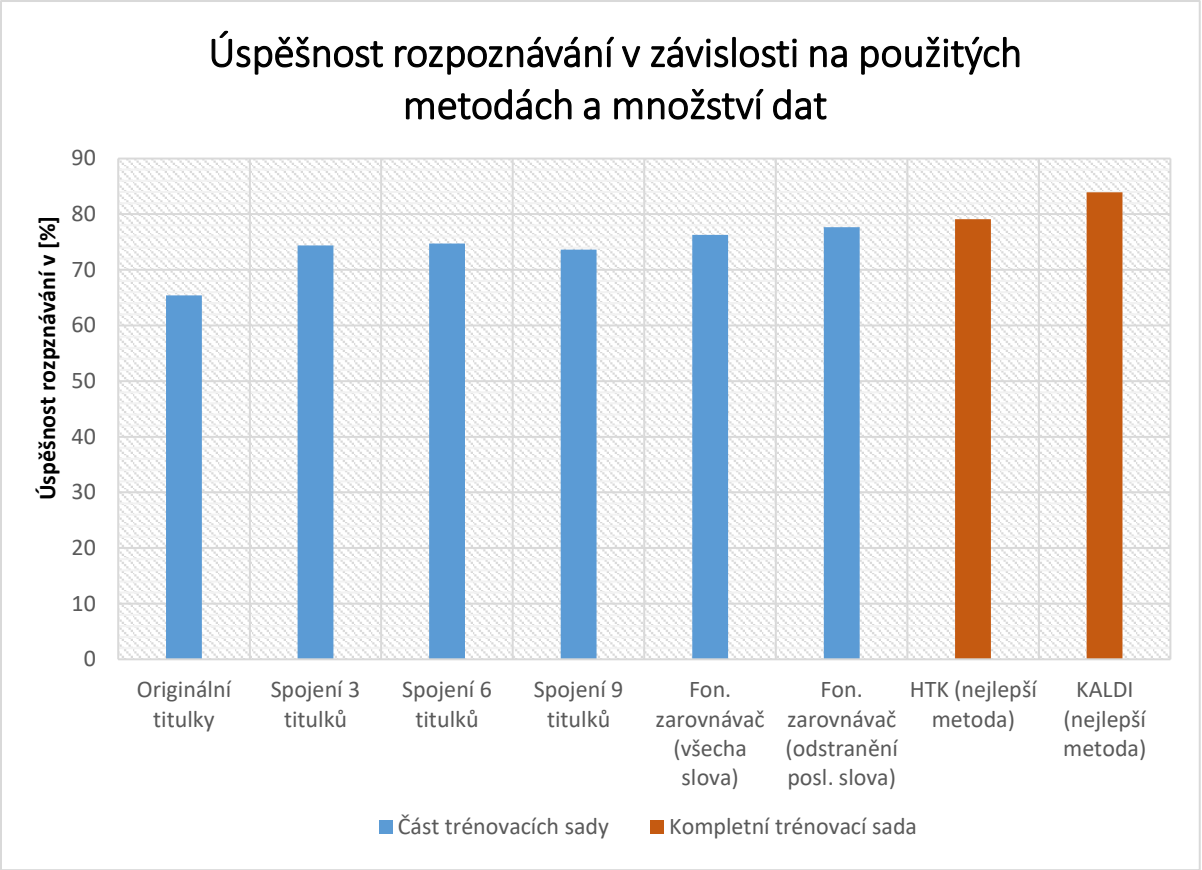
Tyto prvotní experimenty sloužily ke zjištění nejlepšího přístupu úpravy stažených televizních titulků, což je tedy kombinace použití fonémového zarovnávače a odstranění posledního slova

ve větě. Dosažený výsledek této metody 77,65 % je sice sympatický, avšak toto procento ještě zdaleka nestačilo k nasazení tohoto akustického modelu do praxe.

Po získání nelepší metody se trénovací podmnožina obohatila o zbytek trénovacích dat a celá procedura trénování se pro danou nejlepší metodu zopakovala ještě jednou. Celkový počet trénovacích pořadů byl 1009 o celkové délce 405 hodin. Dále se provedly dva další experimenty, tj. natrénování akustického modelu na celé trénovací množině pomocí GMM v nástroji HTK, respektive pomocí neuronové sítě TDNN v Kaldi. Jak již bylo zmíněno výše, neuronové sítě dosahují lepší procentuální úspěšnosti rozpoznávání než klasický přístup GMM-HMM a proto cílem těchto dvou experimentů byla jak snaha o dosažení co možná nejlepší úspěšnosti rozpoznávání, tak také demonstrace rozdílu míry rozpoznávání mezi těmito dvěma přístupy. Úspěšnost rozpoznávání se po přidání všech trénovacích dat a natrénování akustického modelu pomocí HTK zvedla na 79,11 % a v případě neuronové sítě tomu bylo dokonce 83,93 %.

	Použitá metoda / nástroj	Úsp. rozpoznávání – Acc [%]
Část trénovací sady	Originální titulky	65,42
	Spojení 3 titulků	74,36
	Spojení 6 titulků	74,75
	Spojení 9 titulků	73,63
	Fonémový zarovnávač – výběr všech slov	76,30
	Fonémový zarovnávač – odstranění posledního slova	77,65
Kompletní trénovací sada	HTK (GMM) - nejlepší metoda zarovnání	79,11
	KALDI (TDNN) - nejlepší metoda zarovnání	83,93

Obr. 5.7 Tabulka shrnující dosažené výsledky



Obr. 5.8 Graf dosažených výsledků

Z předchozí tabulky a grafu je vidět, že i z nedokonalých slovních přepisů, kterých je na internetu hojné množství, je možné, při využití vhodné metody, natrénovat kvalitní akustický model a dosáhnout tak dobrých výsledků rozpoznávání. Z výsledků je také patrné, že čím více trénovacích dat je k dispozici, tzn. čím více různých řečníků a variabilních akustických pozadí je obsaženo v trénovací sadě, tím se stává akustický model robustnější a tím pádem celý systém ASR funguje podstatně lépe. Dále je také jasné, že neuronové sítě skutečně dosahují v dnešní době podstatně lepších výsledků než klasický přístup založený na GMM-HMM.

Jako poslední experiment se vyzkoušelo přidat všechna trénovací data získaná nejlepší metodou úpravy titulků[17] k již stávajícímu akustickému modelu[18], který Katedra kybernetiky používá v praxi v rámci celého systému ASR pro rozpoznávání médií a pokusit se tak vylepšit a zrobustnit fungování tohoto systému. Úspěšnost rozpoznávání před přidáním zmíněných dat byla na testovacích pořadech 82,72 % a následně po přidání tato úspěšnost stoupla poněkud výrazně na 86,89 %.

[17] To je akustický model natrénovaný na všech trénovacích datech pomocí neuronové sítě.
[18] Akustický model natrénovaný pomocí TDNN neuronové sítě.

Varianta AM	Úspěšnost rozpoznávání – Acc [%]
Před přidáním trénovacích dat	82,72
Po přidání trénovacích dat	86,89

Obr. 5.9 Tabulka popisující úspěšnost rozpoznávání ASR bez přidání, respektive s přidáním dat z webu ČT.

Z výsledků je tedy jasné, že přidáním dat zpracovaných v této diplomové práci došlo skutečně ke zkvalitnění stávajícího systému pro rozpoznávání médií.

Závěr

Cílem této diplomové práce bylo nalézt způsob, jak z webu ČT automaticky extrahovat dostupná data pro tvorbu kvalitních akustických modelů. Na základě hlubšího rozboru tohoto problému došlo k rozdělení úlohy na několik dílčích kroků.

Nejprve byla provedena analýza webu ČT, tj. iVysílání a zjistil se počet a stav pořadů nacházejících se na tomto webu. Bylo zjištěno, že všechny pořady jsou rozděleny podle žánrů do dvanácti sekcí, avšak z výše zmíněných důvodů byly v této práci použity pořady pouze z deseti tříd, kde jejich celkový počet byl odhadem kolem 35000. Dále bylo zjištěno, že z tohoto velkého počtu pořadů je pro akustické modelování nejvhodnější 1009 pořadů o celkové délce 405 hodin.

Dále bylo nutné navrhnout automatický modul, který by právě takto vybraná data stáhl, zpracoval a natrénoval z nich akustický model. Tento systém vznikl v prostředí Python 2.7 a skládá se ze dvou hlavních částí, tj. hlavní program a knihovna. Takto navržený modul má na vstupu pouze soubor s internetovými odkazy na jednotlivé pořady a výstupem je natrénovaný akustický model. Vytvořený modul je tedy zcela automatizovaný a je vhodným nástrojem pro získávání nových dat, která přispívají ke kvalitnějším AM.

Rozborem výše zmíněných dat se dále přišlo na to, že drtivá část vybraných pořadů pro akustické modelování má dva zásadní problémy. Prvním problémem bylo to, že časové značky titulků neodpovídaly zvukové stopě a docházelo zde k časovým nesrovnalostem. Druhým problémem bylo zjištění, že skryté titulky neodpovídaly doslovně své promluvě. Z tohoto důvodu byly navrženy metody, které měly tyto dva problémy vyřešit. Jako nejúčinnější se ukázala metoda fonémového zarovnání promluvy. Díky této metodě došlo k velice výraznému nárůstu úspěšnosti rozpoznávání z původních 65,42 % na 77,65 %. Těchto čísel bylo dosaženo na části trénovacích dat, která se využívala pro zjištění optimální metody úpravy skrytých titulků. Po získání nejlepší metody se do trénovací množiny dodal zbytek upravených trénovacích dat a následně se provedly dva experimenty, tj. experiment využívající GMM a druhý na základě TDNN. V případě GMM došlo k nárůstu úspěšnosti rozpoznávání na 79,11 % a v případě TDNN dokonce na 83,93 %. Je tedy jasné, že velké množství trénovacích dat je velice důležité pro dosažení kvalitního AM a také je zřejmé, že neuronové sítě jsou v dnešní době v úlohách rozpoznávání řeči jasnou jedničkou.

Za účelem potvrzení kvality zpracovaných dat byl na konci proveden experiment, kdy se stejná testovací množina rozpoznala pomocí ASR systému s TDNN akustickým modelem, který využívá Katedra kybernetiky v praxi pro rozpoznávání médií. Následně se provedl druhý experiment pomocí stejného AM, do kterého však byla přidána zpracovaná data. V prvním případě bylo dosaženo úspěšnosti rozpoznávání 82,72 % a v případě přidání dat potom 86,89 %. Je tedy zřejmé, že navržený postupu v této práci opravdu funguje a lze jím produkovat kvalitní AM, které pomáhají zvyšovat úspěšnost rozpoznávání v praxi.

Literatura

[1] Web České televize – iVysílání [online]. Únor 2018. Dostupné z <http://www.ceskatelevize.cz/ivysilani>.

[2] Psutka, Josef, et al. Mluvíme s počítačem česky. Vyd. 1. Praha: Academia, 2006. ISBN 9788020013095.

[3] Pražák, Aleš. Analýza a rozpoznávání řeči - Dekódování [přednáška]. Plzeň, 2016.

[4] Daniel, Povey, et al. The Kaldi Speech Recognition Toolkit. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, 2011. ISBN 9781467303668.

[5] Jahn, Martin. Příprava dat pro tvorbu akustického modelu z webu České Televize [online]. 2016. Dostupné z <https://dspace5.zcu.cz/bitstream/11025/23782/1/BP_Martin_Jahn.pdf>.

[6] Young, Steve, et al. The HTK book (for HTK version 3.4). Cambridge University Press, 2006.

[7] Peddinti, Vijayaditya, Povey, Daniel, & Khudanpur, Sanjeev. A time delay neural network architecture for efficient modeling of long temporal contexts. Interspeech, 2015.

[8] Švec, Jan, et al. Web Text Data Mining for Building Large Scale Language Modelling Corpus. Lecture Notes in Computer Science, 2011, str. 356-363.

[9] Švec, Jan, et al. General framework for mining, processing and storing large amounts of electronic texts for language modeling purposes. Language Resources and Evaluation, 2014, str. 227-248.

[10] Švec, Jan, et al. Technologie pro multimediální a jazykové modelování [online]. 2011. Dostupné z <http://www.kky.zcu.cz/cs/sw/jmzw>.

[11] Haubold, A., Kender, R. J. Alignment of speech to highly imperfect text transcriptions. Beijing: IEEE Signal Processing Society, 2007.

[12] Lecouteux, B., Linarés, G. Using prompts to produce quality corpus for training automatic speech recognition systems. Ajaccio: IEEE Signal Processing Society, 2008.

[13] Prošek, Martin. Čísla vypovídající o češtině [online]. Plzeň, 2010. Dostupné z <https://plzen.rozhlas.cz/cisla-vypovidajici-o-cestine-6803995>.

Přílohy

I. Příklad originálních titulků

1; 11520 13200
Vítejte na Smetanově Litomyšli.

2; 13200 15560
Tak jako má mít ideální žena
svých pět P,

3; 15560 18960
má je i událost,
které věnujeme tento dokument.

4; 19600 20920
Navzdory ženskému rodu můžeme,
myslím,

5; 20920 25520
s klidným svědomím prozradit
její ročník 1949.

6; 26000 28760
Pětice P, která hodláme
akcentovat, představuje slova:

7; 28760 32280
pořadatelé, peníze, program,
počasí a publikum.

II. Formát zarovnaných titulků

13.00	*13.04*	*1.00*	*PAUSA*
13.04	*13.22*	*1.00*	*tak*
13.22	*13.22*	*1.00*	*PAUSA*
13.22	*13.39*	*1.00*	*jako*
13.39	*13.39*	*1.00*	*PAUSA*
13.39	*13.56*	*1.00*	*má*
13.56	*13.56*	*1.00*	*PAUSA*
13.56	*13.76*	*0.50*	*mít*
13.76	*13.76*	*1.00*	*PAUSA*
13.81	*14.32*	*0.75*	*ideální*
14.32	*14.32*	*1.00*	*PAUSA*
14.32	*14.61*	*1.00*	*žena*
14.61	*14.61*	*1.00*	*PAUSA*
14.61	*14.94*	*0.44*	*svých*
14.94	*14.94*	*1.00*	*PAUSA*

14.94	*15.18*	*1.00*	*pět*
15.18	*15.22*	*1.00*	*PAUSA*
15.22	*15.48*	*1.00*	*p*
15.48	*15.48*	*1.00*	*PAUSA*
15.48	*15.66*	*0.63*	*má*
15.66	*15.66*	*1.00*	*PAUSA*
15.66	*15.80*	*1.00*	*je*
15.80	*15.80*	*1.00*	*PAUSA*
15.80	*15.88*	*1.00*	*i*
15.88	*16.07*	*0.25*	*PAUSA*
16.07	*16.62*	*1.00*	*událost*
16.62	*16.62*	*1.00*	*PAUSA*
16.62	*16.92*	*1.00*	*které*
16.92	*16.92*	*1.00*	*PAUSA*
16.92	*17.48*	*1.00*	*věnujeme*
17.48	*17.48*	*1.00*	*PAUSA*
17.55	*17.96*	*0.63*	*tento*
17.96	*17.96*	*1.00*	*PAUSA*
17.96	*18.48*	*0.91*	*dokument*
18.48	*18.86*	*1.00*	*PAUSA*
18.86	*19.32*	*1.00*	*navzdory*
19.32	*19.32*	*1.00*	*PAUSA*
19.32	*19.98*	*1.00*	*ženskému*
19.98	*19.98*	*1.00*	*PAUSA*
19.98	*20.24*	*0.63*	*rodu*
20.24	*20.24*	*1.00*	*PAUSA*
20.24	*20.63*	*1.00*	*můžeme*
20.63	*20.63*	*1.00*	*PAUSA*
20.63	*20.96*	*0.80*	*myslím*
20.96	*20.96*	*1.00*	*PAUSA*
20.96	*21.10*	*1.00*	*s*
21.10	*21.10*	*1.00*	*PAUSA*
21.10	*21.53*	*0.89*	*klidným*
21.53	*21.53*	*1.00*	*PAUSA*
21.53	*22.10*	*1.00*	*svědomím*
22.10	*22.10*	*1.00*	*PAUSA*
22.10	*22.71*	*1.00*	*prozradit*
22.71	*22.71*	*1.00*	*PAUSA*
22.71	*22.99*	*1.00*	*její*
22.99	*22.99*	*1.00*	*PAUSA*
22.99	*23.45*	*0.75*	*ročník*
23.45	*23.45*	*1.00*	*PAUSA*
23.45	*23.45*	*0.40*	*tisíc*
23.45	*23.45*	*1.00*	*PAUSA*
23.45	*23.74*	*0.71*	*devět*
23.74	*24.03*	*-0.04*	*PAUSA*
24.03	*24.23*	*1.00*	*set*
24.23	*24.23*	*1.00*	*PAUSA*
24.23	*24.81*	*0.50*	*čtyřicet*
24.81	*24.81*	*1.00*	*PAUSA*

24.81	*25.24*	*0.75*	*devět*
25.24	*25.55*	*1.00*	*PAUSA*
25.55	*26.03*	*0.89*	*pětice*
26.03	*26.03*	*1.00*	*PAUSA*
26.03	*26.24*	*1.00*	*p*
26.24	*26.24*	*1.00*	*PAUSA*
26.24	*26.53*	*1.00*	*která*
26.53	*26.53*	*1.00*	*PAUSA*
26.53	*27.02*	*1.00*	*hodláme*
27.02	*27.02*	*1.00*	*PAUSA*
27.02	*27.72*	*0.93*	*akcentovat*
27.72	*27.72*	*1.00*	*PAUSA*
27.72	*28.37*	*0.93*	*představuje*
28.37	*28.37*	*1.00*	*PAUSA*
28.37	*28.77*	*1.00*	*slova*
28.77	*28.81*	*1.00*	*PAUSA*
28.81	*29.49*	*0.85*	*pořadatelé*
29.49	*29.49*	*1.00*	*PAUSA*
29.49	*29.97*	*1.00*	*peníze*
29.97	*29.97*	*1.00*	*PAUSA*
29.97	*30.53*	*1.00*	*program*
30.53	*30.62*	*1.00*	*PAUSA*
30.62	*31.21*	*1.00*	*počasí*
31.32	*31.39*	*1.00*	*PAUSA*
31.39	*31.54*	*1.00*	*a*
31.54	*31.54*	*1.00*	*PAUSA*
31.62	*32.19*	*0.72*	*publikum*

III. Příklad zarovnaných titulků bez posledního slova

1; 14940 15880
pět p má je

2; 16070 17480
událost které

3; 17550 23740
tento dokument navzdory ženskému rodu můžeme myslím s klidným svědomím prozradit její ročník

4; 24030 31210
set čtyřicet devět pětice p která hodláme akcentovat představuje slova pořadatelé peníze program

IV. Příklad výsledku rozpoznávání – pomocí HTK při nejlepší metodě

---------- Sentence Scores ----------
========== HTK Results Analysis ==========
Date: Mon May 14 20:39:53 2018
Ref : reference.TMP
Rec : TEMP.TMP
---------- File Results ----------
habsburkove_valka-ktera-zmenila-svet_0.rec: 77.27(72.75) [H=1849, D=137, S=407, I=108, N=2393]
narodni-klenoty_kladruby-kone-pro-cisare.rec: 91.49(88.01) [H=2076, D= 21, S=172, I= 79, N=2269]
nove-objevy-ve-starem-egypte_egyptske-pompeje_0.rec: 79.86(70.80) [H=2106, D= 89, S=442, I=239, N=2637]
turnov-srdce-ceskeho-raje-mesto-drahokamu-a-ceskeho-granatu_11.rec: 90.37(86.84) [H=2073, D= 40, S=181, I= 81, N=2294]
zidovske-pamatky-ceskeho-raje-a-okoli_23_10_2017_14_25_0.rec: 93.90(89.93) [H=2386, D= 23, S=132, I=101, N=2541]
zivot-a-doba-soudce-a-k_exekuce_0.rec: 72.26(66.97) [H=1680, D=198, S=447, I=123, N=2325]
---------- Overall Results ----------
SENT: %Correct=0.00 [H=0, S=6, N=6]
*WORD: %Corr=84.17, Acc=**79.11** [H=12170, D=508, S=1781, I=731, N=14459]*
==========