Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

PLZEŇ, 2018

BC. ONDŘEJ VÁCHAL

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 15.5.2018 ……..

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Mgr. Josefu Psutkovi, Ph.D. za jeho pomoc a úsilí, které mi věnoval při tvorbě této diplomové práce.

Anotace v češtině

Tato diplomová práce je zaměřena na rozpoznávání řeči. Teoretická část se zabývá statistickým rozpoznáváním a to převážně tvorbou akustických modelů pomocí skrytých Markovových modelů. Práce dále obsahuje instrukce k přepisu akustických nahrávek a také návod pro natrénování akustického modelu pomocí nástroje HTK. Praktická část je zaměřena na přípravu dat pro trénování doménových akustických modelů, analýzu vysílacího schématu České televize, výběr vhodných reprezentantů pro přímé titulkování sportovních přenosů a v neposlední řadě pak diskusí získaných výsledků.

Klíčová slova v češtině

rozpoznávání řeči, akustické modelování, skryté Markovovy modely, trénování modelů, referenční přepis, jazykové modelování

Anotace v angličtině

This diploma thesis is focused on speech recognition. The theoretical part deals with statistical recognition. It is mainly focused on acoustic modeling which uses hidden Markov models. Thesis also contains instructions how to annotate audio recordings and also how to train domain acoustic model with tool HTK. The practical part is focused on data preparation and modification, analysis of Czech television broadcasting, choosing the right sport candidates for direct recognition and also discusson of experimental results.

Klíčová slova v angličtině

speech recognition, acoustic models, hidden Markov models, models training, reference transcritpion, langauge modeling

Obsah

1. Úvod

Hlasová komunikace je jeden z nejdůležitějších způsobů dorozumívání se mezi lidmi. V dnešní počítačové době je velká snaha využít této komunikace pro řešení spousty úloh. K tomu je potřeba dokázat člověkem produkovanou řeč rozpoznat. Tímto problémem se zabývají světová výzkumná centra již několik desítek let. Ačkoliv se ze začátku tohoto výzkumu předpokládalo, že velice brzy bude vyvinut systém, který snadno rozpozná, co člověk řekl, tak ani v dnešní době i přes velké pokroky není stále vyvinut rozpoznávač, který by tuto úlohu dokázal bezchybně vyřešit.

Bezchybný systém pro rozpoznávání řeči není stále vyvinut z několika důvodů. Každý člověk má svůj unikátní hlas a způsob mluvení. Zaměříme-li se na Českou republiku a potažmo češtinu, tak i zde nám jinak mluví z hlediska slovní zásoby Čech, Morava či Slezan a drobnější rozdíl nadejme i v řeči člověka z Plzně anebo z Prahy. Další problém je pak samozřejmě barva hlasu, rychlost mluvení, způsob artikulace, neboť z tohoto hlediska se občas člověk neshoduje ani sám se sebou je-li například nachlazen. Dokonce ani za stálých podmínek nedokáže člověk vyslovit jedno slovo vícekrát úplně dokonale stejně.

Dalším problémem z hlediska rozpoznávání řeči, se kterým se také setkáme v diplomové práci, je akustické pozadí. Řeč se mnohem snáze rozpoznává, mluví-li člověk v nahrávacím studiu, v klidu bez emocí. Bohužel téměř vždy je v pozadí přítomen nějaký šum. Ve sportovních přenosech se pak nejčastěji jedná o skandování diváků nebo nějakou hudbu v pozadí akustického signálu. Velký rozdíl také je, jedná-li se o řeč připravenou anebo spontánní, kdy se člověk zadrhává, načne nějaké slovo a pak ho nedořekne a to je samozřejmě další věc, která velice ztěžuje úlohu rozpoznávání. V této práci je rozpoznávána nejhorší varianta řeči a to řeč spontánní, s akustickým ruchem a často také s více řečníky.

Tato práce vznikla z toho důvodu, že katedra kybernetiky již dlouhou dobu spolupracuje s Českou televizí na tvorbě skrytých titulků například pro sportovní pořady nebo politické diskuse. Snahou této diplomové práce je zjistit, které sporty jsou dostatečně kvalitně namluvené a daly by se tedy rozpoznávat přímo ze zvukové stopy. Všechny sporty jsou prozatím přemlouvány stínovým řečníkem a až pak rozpoznávány a vzhledem k tomu, že se jedná o finančně velice náročnou operaci, bylo by dobré, kdyby některé sporty, ideálně samozřejmě všechny, šly rozpoznávat přímo ze zvukové stopy.

Z tohoto důvodu bude v diplomové práci nejprve provedena analýza vysílacího schématu ČT Sport ke zjištění četnosti vysílání jednotlivých sportů. Na základě této analýzy budou vybráni vhodní reprezentanti pro přímé titulkování. Nakonec bude na reálném vysílání ověřena přesnost rozpoznávání získaných titulků a zhodnoceny výsledky.

.

2. Teoretická část

2.1. Řeč a její vznik

Pojem řeč můžeme chápat jako zvukový projev člověka sloužící převážně ke komunikaci. Jedná se také o nejstarší způsob komunikace mezi lidmi. Dávno před tím, než se lidé naučili zaznamenávat své myšlenky písmem, a tak je uchovávat na delší dobu, dorozumívali se mezi sebou ústně. V dnešní době bere člověk řeč jako samozřejmost a nepředstavuje pro něj žádný složitější problém. Zvládne při ní tedy vykonávat mnoho dalších činností a to bez větších obtíží.

Řeč je přenášena komunikačním kanálem ve formě akustického signálu. Podstatou akustického signálu je vlnění elastického prostředí v oboru slyšitelných frekvencí [1]. Pod pojmem komunikační kanál si můžeme představit přenos řeči od úst řečníka k uším posluchače. V akustickém signálu je pak zakódováno spoustu informací a to především informace lingvistické, které nám vyjadřují významy sdělovaných myšlenek. Důležitou součástí je ale také informace o daném mluvčím, která nám určuje například intonaci anebo rytmus řeči. Pomocí této kombinace informací pak poznáme, co přesně se nám daný řečník snaží sdělit. Toto je nesporná výhoda oproti psané formě, ve které nemůžeme pomocí například hlasitosti nebo intonace vyjádřit to, co v mluvené formě.

Řeč vzniká v hlasovém traktu člověka. Ten si můžeme rozdělit na hlasové, dechové a artikulační ústrojí. Dechové ústrojí nám slouží jako zdroj energie řeči. Je tvořeno především dýchací cestou, plícemi a s nimi spjatými svaly. Hlasové ústrojí pak označuje celý systém pro tvorbu řeči. Při vydechnutí prochází hlasivkami vzduch, kde se modifikuje a dále vychází ven z těla přes rty člověka v podobě řeči. Hlasivky jsou tvořeny párem řas. Pokud člověk nemluví, zůstávají hlasivky (hlasivková štěrbina) naplno otevřené a vzduch jim prochází beze změny. Pokud člověk mluví, narazí vzduch na cestě ven na překážku vytvořenou hlasivkami. Ty jsou následně rozkmitány a vytváří zvuk. Tento zvuk je nazýván základním tónem hlasivek. Základní tón hlasivek se pohybuje ve frekvenčním rozsahu 60-400 Hz a velice se liší u dítěte a člověka v dospělosti (u dětí tato frekvence dosahuje vyšších hodnot až 600 Hz). Po průchodu vzduchu přes hlasové ústrojí se dostává na závěr do artikulačního ústrojí. Význam artikulačního ústrojí je především ten, že zvládá vytvářet velké množství různých zvuků, díky čemuž se můžeme snadno dorozumívat. Artikulační ústrojí se skládá především z dutin (například nos) a také orgánů (jazyk). Při vytváření řeči se pak především tyto orgány aktivně podílejí na tvorbě řeči. Pomocí kombinace naladění hlasivek, pozice jazyka a rtů dochází k vytváření nejrůznějších zvuků.

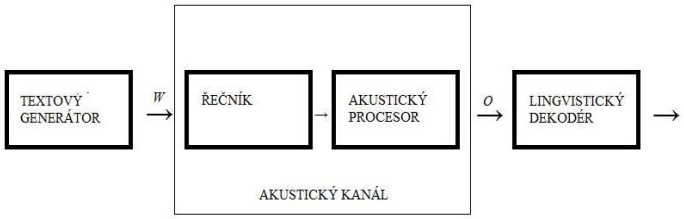
Každá řeč se skládá ze slov, které tvoří slovní zásobu. Jednotlivá slova se pak skládají z fónů. Pro lepší zachycení akustické informace pak využíváme trifóny, které nám vyjadřují daný foném plus jeho pravý a levý kontext.

2.2. Statistické rozpoznávání

První metody rozpoznávání řeči byly vyvíjeny už v sedmdesátých letech minulého století. V té době byly používány metody rozpoznávání na principu porovnávání se vzory. Slovo zde bylo zpracováváno jako celek a bylo klasifikováno do těch tříd, k jejímuž vzorovému obrazu mělo nejmenší vzdálenost. Největším problém těchto metod bylo určení vzdálenosti mezi těmito dvěma obrazy. Tato vzdálenost byla obvykle určována na základě metod dynamického programování, při kterém se hledá taková nelineární transformace časové osy jednoho z obrazů, při níž dojde k porovnání obou obrazů s nejmenší výslednou vzdáleností [1].

Druhá skupina metod je pak založena na statistických metodách. Ty využívají takzvané skryté Markovovy modely, pomocí kterých jsou modelována jednotlivá slova jako celek nebo jako subslovní jednotky. Pod tímto pojmem si můžeme představit modelování slabik, fonémů, trifónů apod., které jsou následně zřetězeny ve výslednou promluvu. Každá subslovní jednotka obsahuje parametry, které jsou nastaveny postupným překládáním trénovacích dat. Neznámá promluva je pak rozpoznána podle toho, jaká posloupnost subslovních jednotek generuje promluvu s největší aposteriorní pravděpodobností.

Základní schéma tohoto přístupu se pak skládá z akustického kanálu a lingvistického dekodéru. Akustický kanál se dále skládá z bloku řečníka a akustického procesoru.



Obrázek 1 Blokové schéma systému rozpoznávání založeném na statistickém přístupu

Na obrázku 1 můžeme vidět základní schéma systému rozpoznávání řeči při použití statistického přístupu. Akustický procesor transformuje řečové kmity produkované řečníkem na posloupnosti vektorů příznaků a lingvistický dekodér překládá i „zkomolené“ řetězce příznaků na řetězce slov. Rozpoznávání je zde formulováno jako problém dekódování s maximální aposteriorní pravděpodobností [1]. Na vstupu přepokládáme, že máme nějakou posloupnost slov $\mathbf{W} = \{w_1\ w_2 \ldots\ w_N\}$ a vektor příznaků získaný z řečového signálu akustickým procesorem $\mathbf{O} = \{O_1\ O_2 \ldots\ O_N\}$. Lingvistický dekodér se pak snaží rozpoznat, co bylo řečeno na základě tohoto vektoru příznaků. Chceme-li vyjádřit tento vztah matematicky, je našim cílem najít nějakou posloupnost slov $\widehat{\boldsymbol{W}}$, která nám bude maximalizovat podmíněnou pravděpodobnost $P(\boldsymbol{W}|\mathbf{O})$. Při využití Bayesova vztahu pak dostaneme:

$$\widehat{\mathbf{W}} = argmax\ P(\boldsymbol{W}|\mathbf{O}) = argmax \frac{P(\boldsymbol{W})P(\boldsymbol{O}|\mathbf{W})}{P(\boldsymbol{O})}$$

kde P(**W**) je apriorní pravděpodobnost posloupnosti slov **W**, P(**O**) je apriorní pravděpodobnosti posloupnosti výstupních vektorů a $P(\boldsymbol{O}|\mathbf{W})$ označuje pravděpodobnost, že při vyslovení slov

W bude vytvořena posloupnost výstupních vektorů **O.** Protože pravděpodobnost P(**O**) není funkcí W, lze ji při hledání maxima ignorovat. Hledanou posloupnost slov $\widehat{\boldsymbol{W}}$ tedy určit maximalizací sdružené pravděpodobnosti P(**W,O**)

$$\widehat{W} = argmax\ P(\boldsymbol{W}, \boldsymbol{O}) = argmax\ P(\boldsymbol{W})P(\boldsymbol{O}|\mathbf{W})$$

Z dané rovnice vidíme, že problém nalezení nejlepší posloupnosti slov k danému vektoru příznaků lze rozdělit na dva úkoly a to na hledání pravděpodobnosti $P(\boldsymbol{O}|\mathbf{W})$ a hledaní pravděpodobnosti P(**W)**. Obě dvě tyto pravděpodobnosti mohou být trénovány nezávisle na sobě. Apriorní pravděpodobnost P(**W**) má v sobě informaci o jazykovém modelu a podmíněná pravděpodobnost obsahuje informaci o akustickém modelu. Obě dvě tyto informace musíme určit ještě před samotným rozpoznáváním a to obvykle za pomoci trénování z řečových a jazykových dat.

Proces rozpoznávání pak spočívá v nalezení takové posloupnosti slov $\widehat{\boldsymbol{W}}$**,** která pro danou posloupnost pozorovaných vektorů příznaků **O** maximalizuje součin pravděpodobností P(**W**) a $P(\boldsymbol{O}|\mathbf{W})$ přes všechny možné posloupnosti slov **W**. Výpočetní náročnost takového vyčerpávajícího hledání je však i pro slovníky menšího rozsahu enormní. V praktických aplikacích jsou proto využívány takové důmyslné suboptimální prohledávací a rozhodovací strategie, které se snaží množství výpočtu účinně redukovat pokud možno s minimálními následky na přesnost rozpoznávání [1].

Z výše uvedeného textu tedy plyne, že úloha statistického rozpoznávání řeči by se dala shrnout v těchto krocích:

1) Akustická analýza řečového signálu, pomocí které získáme posloupnost vektorů příznaků **O.**
2) Tvorba akustického modelu pro určení pravděpodobností $P(\boldsymbol{O}|\mathbf{W})$.
3) Tvorba jazykového modelu pro určení pravděpodobností P(**W**)**.**
4) Nalezení nejpravděpodobnější posloupnosti slov

Tyto čtyři kroky jsou tedy klíčové pro statistické rozpoznávání řeči a dále se na ně podíváme podrobněji.

2.2.1. Akustická analýza řeči

Pro samotné rozpoznávání řeči je nejprve nutné ze zvukové stopy pomocí metod předzpracování získat relevantní informace. K tomu se využívá parametrizace řeči. Základním předpokladem těchto metod je, že hlasivkový trakt si lze představit v dostatečně krátkém časovém úseku jako stacionární. Pro akustickou analýzu řeči se využívá metod krátkodobé analýzy, které tedy nezpracovávají celý akustický signál dohromady, ale pracují s jednotlivými mikrosegmenty jako by to byly oddělené krátké zvuky. Tyto mikrosegmenty se obvykle pohybují v rozmezí 10-30 ms. Výsledkem parametrizace jsou pak vektory parametrů, které popisují jednotlivé mikrosegmenty. Jejich spojením pak dostaneme parametrizovaný celý akustický signál. Než se pustíme do samotné parametrizace promluv, je nejprve nutné provést digitalizaci signálu. Digitalizace signálu se pak skládá ze dvou základních částí a to ze vzorkování a kvantizace spojené s kódováním.

- **Pulsní kódová modulace** je tedy metoda, která převádí analogový zvukový signál na digitální. Výsledkem je pak signál v digitální a tedy dále dobře zpracovatelné podobě pro metody parametrizace.

1) **Vzorkování**
   Vzorkování signálu je proces jeho diskretizace v časové oblasti. Klíčovým parametrem pro vzorkování je perioda vzorkování **T**. Podíváme-li se na obrázek 2, tak periodu vzorkování můžeme vidět ve spodní části jako rozdíl mezi jednotlivými vyznačenými body. Důležitým předpokladem zde je, aby vzorkovací frekvence **f** = 1/**T** splňovala Shannonův vzorkovací teorém. Ten nám říká, že použitá vzorkovací frekvence musí být alespoň dvakrát vyšší, než je nejvyšší harmonická složka vzorkovaného signálu.

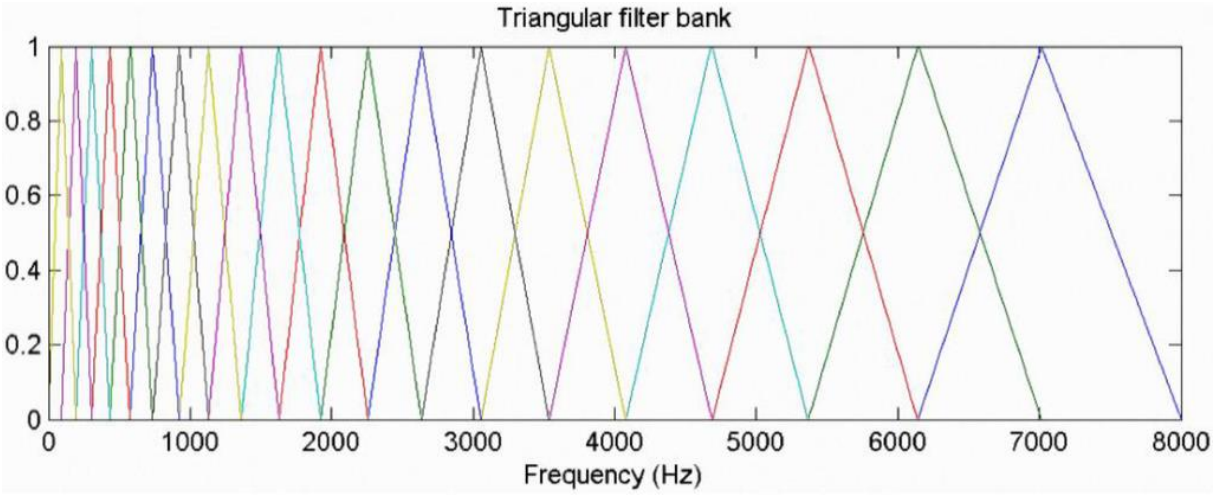
2) **Kvantizace a vzorkování**
   Kvantizace s následným kódováním je přiřazení analogové hodnoty jednomu z konečného počtu číselných hodnot. Počet těchto hodnot se obvykle volí ve tvaru 2^N.
   Tento krok je obvykle prováděn pomocí A/D převodníků. Při přiřazování hodnoty jednotlivých vzorků dochází k zaokrouhlování velikosti signálu na jednu z nejbližších hodnot. Dochází tedy ke ztrátě informace nazývané také jako kvantizační šum.

- **MFCC** je homomorfní metoda zpracování řeči. To znamená, že se hodí pro analýzu signálu, které vznikli konvolucí dvou nebo více složek. Vzhledem k tomu, že způsob vzniku řeči se dá popsat jako konvoluce budícího signálu a impulsní funkce hlasového ústrojí, je použití této metody vhodné. MFCC vychází ze způsobu, jakým člověk vnímá řeč. Změny ve výšce zvuku nejsou vnímány lineárně, ale logaritmicky. Vyskytují se zde kritická pásma slyšení, které představují frekvenční oblasti, kde dochází k maskování zvuku. Z toho důvodu je pro MFFC zavedena stupnice subjektivní výšky zvuku. Ta má jednotky [Mel] a z frekvence v [Hz] se přepočítává pomocí vztahu:

$$f_{mel} = 2595.log_{10}(1 + \frac{f}{700})$$

Průběh parametrizace by se pak dal shrnout do těchto kroků:

1. Aplikace Hammingova okénka na signál rozdělený na segmenty o délce 20-30ms
2. Provedení Fourierovy transformace signálu a vypočtení krátkodobého výkonového spektra
3. Filtrace za pomoci melovských filtrů – ta je realizována pomocí trojúhelníkových pásmových filtrů s rovnoměrným rozložení středních frekvencí jednotlivých filtrů podél frekvenční osy s měřítkem v melovské škále. (viz obrázek 2)
4. Logaritmus jednotlivých energií filtrů
5. Zpětná Fourierova transformace (využívá se Cosinovy transformace)



Obrázek 2 Příklad rozložení trojúhelníkových filtrů

Výsledné kepstrum se pak spočte pomocí vzorce:

$$c_m(j) = \sum_{i=1}^{M} log_{10}\big(y_m(i)\big) * \cos(\frac{\pi}{M}(i - 0.5) * j)$$

kde M je počet filtrů a N je žádaný počet melovských kepstrálních koeficientů.

Dynamické koeficienty delta a delta-delta (akcelerační) nám dále určují dynamiku časové změny vektorů příznaků. Ty se určují za pomoci lineární regrese. Výsledný vektor příznaků se pak skládá z melovských kepstrálních koeficientů, delta a delta-delta koeficientů.

2.2.2. Akustické modelování

Jak již bylo zmíněno, úkolem akustického modelování je co nejrychleji a co nejpřesněji odhadnout pravděpodobnost P(**O**|**W**) pro libovolnou posloupnost slov a pro libovolné vektory příznaků. Akustický model by měl splňovat několik fundamentálních vlastností:

1) **Flexibilita** - ta je důležitá z důvodu, že podmínky, které byly dostupné při trénování, se mohou velice lišit od podmínek, které jsou dostupné při rozpoznávání. Pod těmito podmínkami si můžeme představit například odlišné akustické pozadí, odlišné tempo řeči anebo také zcela jiného řečníka.
2) **Přesnost** – pod heslem přesnost si představme především to, abychom byli schopni odlišit slova, která jsou akusticky velice podobná, avšak znamenají něco naprosto jiného. Jako například slova *nes, pes*.
3) **Účinnost -** tato vlastnost nám zaručuje to, aby byl systém při nasazení v reálných aplikacích schopen rozpoznávat řeč v reálném čase.

Jako velice vhodný nástroj pro řešení této úlohy se ukázalo být použití skrytých Markovových modelů HMM (Hidden Markov Model).

- **Skryté Markovy modely -** HMM byly poprvé využity již v sedmdesátých letech minulého století. Větší popularity se však dočkaly až o dekádu později, kdy se ukázaly být jako nejlepší metoda na rozpoznávání diskrétního diktátu. Modelování řeči pomocí HMM tedy patří do skupiny metod, které využívají děje, při kterém člověk generuje řeč. Na velmi krátký časový okamžik si pak můžeme představit, že hlasové ústrojí je v neměnném stavu, ve kterém produkuje signál závisející na daném stavu. Takovýto kousek signál může být popsán vektorem příznaků (např. MFCC).

  Z představy o vytváření řeči pak vychází i konstrukce klasifikátoru založená na modelování řečového signálu pomocí Markovova procesu. Během toho jsou generovány dvě vzájemně svázané časové posloupnosti náhodných proměnných a podpůrný Markovovův řetěz, které je posloupností konečného počtu stavů a řetězec vektorů příznaků, který reprezentuje spektrální charakter jednotlivých mikrosegmentů řečového signálu. Pro tyto charakteristiky jsou pak vytvořeny náhodné funkce, které pravděpodobnostně ohodnocují vztah charakteristik ke všem stavům [1].

  Abychom tedy co nejlépe odhadli rozdělení pravděpodobnosti P(**O**|**W**), využijeme k tomu skrytých Markovových modelů. Našim úkolem bude odhadnout strukturu a parametry jednotlivých HMM. K řešení tohoto problému se dá v zásadě využít dvou metod a to expertní odhad anebo metodu statistické indukce. V našem případě využijeme expertního odhadu pro určení topologie HMM a metodou statistické indukce pak natrénujeme jeho parametry.

- **Struktura HMM -** Skrytý Markovovův model je stochastický proces, který v diskrétním čase generuje posloupnost vektorů pozorování. V každém časovém okamžiku pak model změní svůj stav ($\mathbf{s}_i$ na $\mathbf{s}_k$) podle předem dané pravděpodobnosti $\boldsymbol{a_{ik}}$. Stav $\mathbf{s}_i$ pak generuje v každém časovém okamžiku vektor příznaků $\mathbf{o}_i$ a to na základě výstupní pravděpodobnosti daného stavu $f_i(\mathbf{o}_t)$.

  Pro pravděpodobnost přechodu pak platí:

$$\boldsymbol{a_{ik}} = P(\boldsymbol{s}(t+1) = \boldsymbol{s_k} \mid \boldsymbol{s}(t) = \boldsymbol{s}_i)$$

  A také:

$$\sum_{1}^{N} \boldsymbol{a_{ik}} = 1$$

  A pro rozdělení výstupní pravděpodobnsti:

$$\boldsymbol{f_k}(\boldsymbol{o_t}) = P(\boldsymbol{o_t} \mid \boldsymbol{s}(t) = \boldsymbol{s_k})$$

kde $\boldsymbol{s_k}$ je stav v čase *t* a P je hustota pravděpodobnosti. Rozdělení této pravděpodobnosti pak musí být zároveň specifické a robustní. Robustní z toho důvodu, aby bylo schopno rozpoznat hlas více řečníků, a specifické aby bylo schopno od sebe oddělit nejrůznější zvuky. V dnešní době je nejčastěji využívaná neuronová síť. Dále je hojně využívané spojité rozdělení se směsí normálních hustotních funkcí, diskrétní rozložení nebo spojité rozdělení se svázanou směsí normálních hustotních funkcí.

Spojité rozdělení se směsí normálních hustotních funkcí – tvar výstupní hustoty pravděpodobnosti je tvořen váženým součtem jednotlivých normálních hustot pravděpodobnosti, z nich je každá určena svým vektorem střední hodnot a svou kovarianční maticí [1]. Parametry jsou navíc určeny ještě váhami jednotlivých složek.

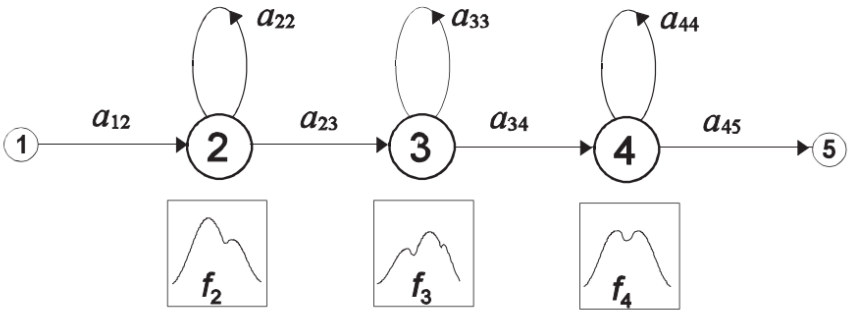
$$f_k(\boldsymbol{o_t}) = \sum_{j=1}^{m} \pi_{kj} N(\boldsymbol{o_t}, \boldsymbol{u_{kj}}, \boldsymbol{\Sigma_{kj}})$$

Tvar hustotní funkce můžeme vyjádřit například výše uvedeným vztahem, kde *m* značí počet složek, $\boldsymbol{o_t}$ je daný vektor pozorování s váhou π_{kj}, se střední hodnotou $\boldsymbol{u_{kj}}$ a kovariancí $\boldsymbol{\Sigma_{kj}}$. $N(\boldsymbol{o_t}, \boldsymbol{u_{kj}}, \boldsymbol{\Sigma_{kj}})$ je tedy vícedimenzionální normálové rozdělení a můžeme ho zapsat také jako:

$$N\left(\boldsymbol{o_t}, \boldsymbol{u_{kj}}, \boldsymbol{\Sigma_{kj}}\right) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma_{kj}})}} \exp[-0.5\left(\boldsymbol{o_t} - \boldsymbol{u_{kj}}\right)^T \boldsymbol{\Sigma_{kj}}^{-1}(\boldsymbol{o_t} - \boldsymbol{u_{kj}})]$$

kde *d* značí dimenzi vektoru příznaků a $\det(\boldsymbol{\Sigma_{kj}})$ je determinant kovarianční matice $\boldsymbol{\Sigma_{kj}}$.

Z hlediska struktury se používají především levo-pravé Markovovy modely. Ty jsou velice vhodné především pro modelování procesů vyvíjejících se v čase. Pomocí levo-pravého Markovova modelu jsou tedy modelovány subslovní jednotky.



Obrázek 3 model fonému s třemi emitujícími stavy

Na obrázku 3 vidíme příklad pětistavového modelu. V tomto případě se jedná o model se dvěmi neemitujícími stavy (1 a 5) a třemi emitujícími stavy, tedy stavy obsahujícími parametry. Velkou výhodou modelování pomocí menších jednotek než celých slov je potřeba menšího počtu trénovacích dat. Pokud bychom totiž trénovali

celá slova, bylo by zapotřebí pro každé slovo několik výskytů a vzhledem k obrovské slovní zásobě českého jazyka by se jednalo o téměř nadlidský úkol. V této diplomové práci je využito převážně modelování za pomoci trifónů. Trifón je tedy kontextově závislý foném. Velkou výhodou je, že využívá pravý a levý kontext a lépe tedy modeluje jednotlivá slova. Do jisté míry je samozřejmě nevýhodou, že je zapotřebí většího množství dat. Tento problém se však dá odstranit shlukováním trifónů s akusticky podobným okolím a je toho využíváno i v diplomové práci.

K trénování těchto modelů je pak používán nejčastěji Baum-Welchův algoritmus. Tento algoritmus byl vyvinut Lenoradem E. Baumem v 70 letech 20 století a slouží tedy k určení parametrů HMM. Tento algoritmus využívá algoritmus forward-backward a jedná se o speciální verzi známého EM algoritmu. Podrobnější popis algoritmu můžeme najít v [1], [2].

- **Shrnutí akustického modelování při běhu na reálném systému -** při běhu na reálném systému je nejprve potřeba „protáhnout" signál blokem na zpracování signálu. K tomu můžeme využít buď to MFCC případně LPC (lineární prediktivní kódování) pro získání vektorů příznaků.

  Dále pak využijeme akustický model pro určení jednotlivých pravděpodobností v každém mikrosegmentu. K tomu využíváme výše zmíněného Gaussovského rozdělení. V praxi nám tento krok provádí blok zvaný labeler, jehož úkolem je spočítat s jakou pravděpodobností vytváří nějaký stav ***s*** příznakový vektor ***o***. Rozpoznávání tedy spočívá v tom, že pro každou posloupnost slov náleží nějaký zřetězený Markovovův model. Model pak má za úkol určit pravděpodobnost, že byla určitá výslovnost slov řečena.

  V praxi pak můžeme narazit na problém ohledně výpočetní náročnosti. Vzhledem k tomu, že se používá deseti tisíce Gaussovských funkcí s dimenzí například 39 je potřeba na reálných systém použít metody pro snížení této náročnosti. To bohužel může vést také ke snížení kvality při rozpoznávání.

2.2.3. Jazykové modelování

Hlavním úkolem jazykového modelování je rychle a přesně poskytnout odhad apriorní pravděpodobnosti P(**W**) pro jakoukoliv posloupnost slov. Vytváření jazykového modelu je pro každý jazyk trošku jiné, neboť každý jazyk funguje na trochu jiném principu. Především se bavíme o slovníku, který vybraný jazyk používá a pak také zákonitosti, kterými se daný jazyk řídí. To znamená, jakým způsobem jsou slova řazena do vět. Jazykový model se toto snaží modelovat, avšak tím na sebe klade také jistá omezení. Prvním z nich je, že nemůže být rozpoznáno jiné slovo, než které má v sobě daný slovník. Druhé z nich pak je, že některé posloupnosti slov nemůžou být vůbec vysloveny. V různých situacích má také stejná posloupnost slov jinou pravděpodobnost. Zde záleží především na kontextu dané řeči a tedy například na tématu, úmyslu řečníka nebo jeho náladě. Při samotné konstrukci jazykového modelu pak řešíme především dva hlavní problémy. Prvním z nich je, aby model správně určoval pravděpodobnost P(**W**) pro libovolnou posloupnost slov **W**. Druhým problémem je,

aby model při samotném rozpoznávání dokázal co nejrychleji již během toho, co daný řečník mluví, poskytovat údaje o pravděpodobnosti P(**W**) a neměl by vyčkávat až do konce promluvy. Pokud by tento požadavek splněn nebyl, pak by v reálném případě rozpoznávání probíhalo velice zpomaleně.

- **Stochastické jazykové modelování -** Stochastický model nám tedy určuje pravděpodobnost pro libovolnou posloupnost slov. Důležité je, aby žádná posloupnost neměla nulovou pravděpodobnost. Protože pokud bychom tak udělali, pak by při výskytu neznámého slova nemohla být promluva nikdy správně rozpoznána. Toto je problémem převážně při rozpoznávání spontánní řeči, kdy se řečník často zasekává a používá nespisovná slova a jedná se i o problém vyskytující se v této diplomové práci. Pravděpodobnost je pak počítána pomocí vzorce:

$$P(\boldsymbol{W}) = \sum_{k=1}^{K} P(\boldsymbol{w_k} | \boldsymbol{w_{k-1}} \dots \boldsymbol{w_1})$$

  kde *k* je počet slov posloupnosti **W**. Pravděpodobnost P(**W**) je tedy podmíněna svojí historií $(\boldsymbol{w_1} - \boldsymbol{w_{k-1}})$. Vzhledem k tomu, že systémy rozpoznávání řeči obvykle obsahují obrovské slovníky slov, pak není možné tyto pravděpodobnosti všechny ocenit. Z toho důvodu pracujeme s jejich aproximací a to taková, že všechny historie slov, které se shodují v n-1 slovech jsou zařazeny do stejné třídy. Takovým to modelům pak říkáme n-gramové modely. V praxi se pak nejčastěji vyskytují bigramové nebo trigramové modely. Velká výhoda těchto modelů je, že se hodí na jazyky s pevnější skladbou věty. Dále pak jejich konstrukce také není příliš složitá, neboť jednotlivé pravděpodobnosti jsou určovány na základě relativní četnosti v trénovacích datech. Největším problémem vytváření n-gramových modelů je pak nedostatek dat. To platí pro český jazyk obzvlášť, neboť kvůli jeho skloňování a mnoho tvarů slov je vytvořit kvalitní jazykový model komplikovanější než pro jiné jazyky.

- **Třídní jazykové modely -** velkým problémem při rozpoznávání řeči je rozpoznávání slov, se kterými se jazykový model zatím nesetkal. V takovém to případě musíme nejprve identifikovat slova, která chybí v jazykovém modelu a také způsob, jakým je do modelu správně přidávat. To se dá řešit pomocí třídních jazykových modelů. K vytvoření takovýchto modelů je potřeba v trénovacích datech nějak označit jednotlivé třídy. Zaměříme-li se například na problematiku sportů, kterou se tato práce zabírá, tak zde se nejčastěji se jedná o jména sportovců, sportovišť, hlavních měst a podobně. Tyto třídy jsou tedy při přípravě dat řazeny do speciálních kategoriích. Další problém je, že český jazyk má navíc několik různých pádů a tedy tvarů jmén. Zaměříme-li se například na jména sportovců, tak pro ty bylo vytvořeno 5-6 různých tříd (máme 7 pádů, ale některé pády mají stejné tvary). Podíváme-li se opět do praxe řešené v této práci, tak máme-li k dispozici třídní jazykový model, stačí před samotným rozpoznáváním doplnit jazykový model o soupisku nových jmén. Ty jsou pak pomocí automatického systému vyskloňovány do všech možných tvarů a přidány do jazykového modelu. Vzhledem k tomu, že i soupisky hráčů v jednotlivých utkání nebo například místa konání, názvy hal, měst jsou k dispozici na internetu, dá se tento

problém téměř automaticky tímto způsobem řešit. Takto vytvořené jazykové modely velmi výrazně snižují množství OOV (out of vocabulary) slov a tím pádem i perplexitu daného jazykového modelu a tím výrazně zvyšují úspěšnost rozpoznávání. Pod pojmem perplexita jazykového modelu si můžeme představit způsob na zjišťování kvality daného jazykového modelu. Jedná se o číslo, která nám uvádí, mezi jakým počtem slov se bude daný systém pro rozpoznávání rozhodovat. Čím je toto číslo menší, tím je jazykový model kvalitnější. Podrobnější popis jazykového modelování můžeme najít v [1][3].

3. Tvorba titulků pro sportovní pořady

V předchozí kapitole bylo ukázáno, z jakých částí se skládá systém pro rozpoznávání řeči a jak jednotlivé části fungují. Dále přejdeme k tomu, jak vlastně funguje rozpoznávání řeči na reálném vysílání. V dnešní době se používají v podstatě dvě přístupy pro tento úkol a to automatické rozpoznávání přímo z televizní stopy anebo použití stínového mluvčího, který poslouchá televizní komentář a přemlouvá ho systému rozpoznávání. Oba tyto přístupy mají své výhody i nevýhody a jsou použitelné pro různé typy přenosů. Hlavní charakteristiky přímého titulkování jsou pak mnohonásobně nižší náklady na titulkování, ale bohužel také větší náchylnost na chyby. To převážně z toho důvodu, že živé přenosy obsahují často vysoký ruch v pozadí, může přes sebe mluvit více komentátorů a i přes použití třídního jazykového modelu se v tomto přenosu objevuje více OOV slov.

V České republice již dlouhá léta běží systém pro titulkování sportovních přenosů, politických debat, zábavných show a podobně. Tento systém vyvíjela na základě spolupráce s Českou televizí Fakulta kybernetiky Západočeské univerzity v Plzni. Úspěšně byly otitulkovány například poslední zimní a letní olympijské hry v Sochi a v Pchjongčchangu. Obě tyto olympiády však byly titulkovány druhým způsobem a to použitím stínového mluvčího. Hlavním úkolem stínového řečníka je přemlouvání promluv, které byly řečeny komentátory, v klidném akustickém prostředí, klidným hlasem při zachování podobného obsahu. Člověk, který provádí tento úkol, tedy sedí ve studiu, poslouchá přenos, který má o pár vteřin dříve než samotní diváci, a snaží se ho přemluvit do co nejlepší podoby. Systém pro rozpoznávání řeči má k dispozici speciální akustický model obvykle natrénovaný na hlas daného řečníka a také speciální jazykové modely připravené pro jednotlivé sporty.

Při použití stínového řečníka je dosaženo velmi dobrých výsledků pohybujících se okolo hranice 95 procent. Důležitou otázkou tedy je, zda-li je možné použít titulkování sportovních pořadů přímo ze zvukové stopy. A pokud ne pro všechny sporty, tak alespoň pro některé. Aby byl systém použitelný, bylo by dobré, aby se hranice úspěšnosti pohybovala nad 90 procenty. Pro ověření tohoto způsobu byl již Západočeskou univerzitou v Plzni proveden experiment, který zkoušel rozpoznávání přímé zvukové stopy na hokeji. K tomu bylo využito modelování pomocí skrytých Markovových modelů za použití 22 Gausovských směsí při 4922 stavech (podrobněji [4]). Z hlediska jazykového modelování pak byly použity třídní jazykové modely doplněny o soupisky jednotlivých sportů. Bohužel výsledky rozpoznávání se ukázaly být velice neuspokojivé. Zatím co u sportů přemlouvaných stínovým řečníkem se úspěšnost

rozpoznávání pohybovala okolo 98 procent, v rozpoznávání z přímé zvukové stopy se pohyboval okolo 65 procent. Bohužel takto vytvořené titulky nejsou příliš použitelné pro praktické použití. Důvodem takto špatných výsledků je opravdu velká přítomnost akustického šumu v pozadí a také spousta přeřeků a skákání si do řeči jednotlivých komentátorů.

Úkolem mé práce tedy bylo zjistit, zda-li při použití specifických akustických modelů pro jednotlivé sporty bude možno alespoň někde přejít na titulkování z přímé zvukové stopy a ušetřit tak nemalé náklady.

Před přechodem k praktické části bych ještě rád zmínil, jakým způsobem funguje titulkování řeči v ostatních jazycích. Ať už se jedná o angličtinu, francouzštinu nebo japonštinu, tak veškeré systémy rozpoznávání fungují na podobném principu. Vzhledem k velké přítomnosti šumu a ne příliš kvalitní přímé zvukové stopě (z hlediska rozpoznávání) je i v zahraničí hojně využíváno stínového řečníka (anglicky re-speaker). Ze zahraničních jazyků mě nejvíce zaujala japonština, kde se rozpoznávání řeči používá například pro titulkování hodinových zpráv. Akustický model je zde natrénován podobně jako v ostatních zemích pomocí HMM. Avšak jazykový model funguje na velice zajímavé bázi, kde stejně jako v ostatních jazycích byl vytvořen podobným způsobem, avšak pro zlepšení výsledků je každý den rozšiřován a adaptován o nová slova. Do jazykového modelu jsou tedy přidávána rozpoznaná slova z posledních několika hodin a těmto slovům je přidávána vysoká váha. Takovýto model se nazývá Time Dependent Language Model (TDLM) a velice kladně přispívá ke snížení počtu OOV slov a snížení perplexity daného jazykového modelu [10], [11].

4. Trénování a evaluace akustických modelů pomocí HTK

4.1. Co je HTK

Název HTK vychází z anglického Hidden Markov Model Toolkit. Je to přenosný nástroj sloužící pro tvorbu a manipulaci se skrytými Markovovými modely. HTK se skládá ze setu knihoven a nástrojů napsaných v jazyce C a slouží především tedy pro analýzu řeči, trénování skrytých Markovových modelů a analýzu výsledků. Nástroj HTK byl vytvořen v 90 letech 20. století na univerzitě Cambridge a je možné ho pro nekomerční využití stáhnout z webových stránek.

Návod obsahuje jednoduché příkazy, které stačí zkopírovat nebo přepsat a vložit do příkazového řádku. Nejobtížnější část je tak připravit vstupní data jednotlivých příkazů do požadované formy. Podrobnější návod spolu s příkazy najdeme v [7], [12].

4.2. Příprava dat k trénování

4.2.1. Soubory nutné pro natrénování akustického modelu

Pro práci s nástrojem HTK je tedy vždy potřeba připravit několik souborů, které musí dodržovat správný formát a musí být umístěný ve správné složce. Důležité je také dodržet stejné značení jednotlivých promluv ve všech potřebných souborech. Pro trénování je potřeba:

1) Referenční přepis *words.mlf.* Jedná se o přepis na úrovni slov a to jak testovacích tak trénovacích vět. Na první řádce souboru musí být veden výraz *#!MLF!#.* Dále jsou pak

uvedeny unikátní označení jednotlivých vět následovaných vždy jejich obsahem. Promluva je ukončena tečkou, která je vždy na samotné řádce.

2) Výslovnostní slovník *dict_sp.txt* a *dict.txt*. Tyto slovníky obsahují výslovnosti všech slov, které jsou obsaženy v referenčním přepisu *words.mlf*. Rozdíl mezi těmito dvěma soubory je pouze ten, že slovník *dict_sp* obsahuje výslovnosti slov pouze s krátkou pauzou _sp_.
3) Soubory *monophones0* a *monophones1*, které obsahují všechny fóny, které používáme při fonetické transkripci. Rozdíl mezi soubory je pouze ten, že v *monophones1* musíme mít symbol krátké pauzy.
4) Seznam trénovacích a testovacích promluv *train.scp* a *test.scp.* Na promluvách, které jsou obsažený v trénovacím seznamu, pak natrénujeme náš akustický model a na testovacích promluvách ho následně otestujeme. Tyto promluvy musí být rozdílné, abychom natrénovali a netestovali na stejných datech.
5) Seznam promluv pro parametrizaci ve formátu:
   wav/bojove_sporty0000veta0001.wav htk/bojove_sporty0000veta0001.htk
   wav/bojove_sporty0000veta0002.wav htk/bojove_sporty0000veta0002.htk
   wav/bojove_sporty0000veta0003.wav htk/bojove_sporty0000veta0003.htk
   wav/bojove_sporty0000veta0004.wav htk/bojove_sporty0000veta0004.htk
   wav/bojove_sporty0000veta0005.wav htk/bojove_sporty0000veta0005.htk
   kde ve složce *wav* je zvuková stopa nařezána na jednotlivé promluvy a do složky *htk* jsou pak ukládány parametrizované výsledky.

4.2.2. Tvorba přepisu na úrovni fonémů

Systém bude založený na modelování malých jednotek. Pro zatím budeme pracovat s modelování pomocí monofónů, ale v dalších krocích se také dostaneme k modelování pomocí složitějších jednotek zvanými trifóny, kde je využíváno také pravého a levého kontextu daného fónu a je tak dosaženo lepších výsledků. Každý monofón bude reprezentován pomocí vlastního skrytého Markovova modelu. Proto kromě transkripce na úrovni slov, kterou máme již vytvořenou, je potřeba vytvořit také transkripci na úrovni monofónů (později trifónů). Tento přepis vytvoříme pomocí programu HLEd z knihovny HTK ze souboru *words.mlf* a výslovnostního slovníku *dict.txt*. Nově vytvořené soubory pak vypadají následovně:

Phones0.mlf:

#!MLF!#
"/badminton0000veta0001.lab"*
*_sil_*
n
a
z
a
C
A
t

phones1.mlf:

#!MLF!#
"/badminton0000veta0001.lab"*
*_sil_*
n
a
*_sp_*
z
a
C
A

k
u
d
r
u
h
*_sil_*
t
k
u
*_sp_*
…

4.2.3. Parametrizace řeči

Prozatím máme nahrávky pouze ve zvukové podobě. Abychom je však mohli použít pro natrénování akustických modelů, je potřeba je převést na posloupnost vektorů pamarametrů. Parametrizovat se bude pomocí MFCC 13 koeficientů + delta + delta delta koeficienty. Výsledkem tedy bude vektor popisující jeden mikrosegmet o velikosti 39.

4.3. Definice a tvorba monofonního modelu:

Nyní se dostaneme do fáze, kde každý fón české fonetické abecedy bude reprezentován jedním skrytým Markovovým modelem. Nejprve je třeba nadefinovat, jak budou jednotlivé Markovovské modely vypadat. Každý model bude mít 5 stavů, které si můžeme představit jako 5 kruhů vedle sebe. Dva krajní kruhy budou takzvané neemitující stavy. Pod pojmem neemitující stavy si můžeme představit, že negenerují žádný vektor parametrů a slouží k propojení jednotlivých modelů. Emitující stavy jsou pak stavy generující vektory parametrů. V praxi pak může model například pro písmeno A vypadat následovně:

~h "A"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
-5.260925e+000 -6.052345e+000 -1.219461e+000 -2.440739e+000…….
<VARIANCE> 39 2.208805e+001 7.596239e+000 4.013688e+000……..
<GCONST> 7.698160e+000
<STATE> 3
<MEAN> 39
-5.260925e+000 -6.052345e+000 -1.219461e+000……
<VARIANCE> 39
2.208805e+001 7.596239e+000 4.013688e+000 2.436403e+000…….
<GCONST> 7.698160e+000
<STATE> 4
<MEAN> 39
-5.260925e+000 -6.052345e+000 -1.219461e+000
<VARIANCE> 39
2.208805e+001 7.596239e+000 4.013688e+000……
<GCONST> 7.698160e+000
<TRANSP> 5

0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 6.000000e-001 4.000000e-001 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 7.000000e-001 3.000000e-001
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

kde vidíme, že za řetězcem *~h* následuje jméno modelu. Další řádek nám pak znační začátek HMM. Řetězec *<NUMSTATES> 5* značí počet stavů, kde stav 1 a 5 jsou stavy neemitujícími a stavy 2,3,4 jsou stavy generující parametry. Vektory parametrů v emitujících stavech se pak řídí Gaussovým rozdělením pravděpodobnosti a u každého z těchto stavů tedy potřebujeme znát kovarianční matici a střední hodnotu. Na konci příkladu pak vidíme parametr *<TRANSP>.* Ten značí údaj o velikosti matice přechodu následovaný samotnou maticí. Je-li parametr nulový, pak není definován přechod z jednoho stavu do druhého. Naopak pokud je parametr nenulový, pak je definován přechod z jednoho stavu do druhého. Vidíme také, že pravděpodobnost na jednotlivých řádcích nám musí dát číslo 1.

Vrátíme-li se nyní k samotnému trénování, tak prvotní model je vytvořen pomocí programu HCompV. Ten nám spočte celkovou střední hodnotu a kovarianci ze všech trénovacích dat.

Zde je důležité, aby byli výstupní adresáře pojmenované např. *hmm0, hmm1…* vytvořené před samotným spuštěním programu. Neuděláme-li tak, program zahlásí chybu. Je dobré si tedy ze začátku trénování vytvořit složky *hmm0* až *hmm9* a pak ještě *hmmA* a *hmmB* pro finální rozpoznávání.

Dále je potřeba v adresáři hmm0 vytvořit takzvaný *Master Marco File*, který bude obsahovat HMM pro jednotlivé monofóny. Program nám neudělá nic jiného, než že rozkopíruje z prvního kroku vytvořené parametry pro každý jednotlivý monofón a pouze změní jeho jméno.

Nyní jsme tedy ve stavu, kdy máme vytvořené jednotlivé modely a je potřeba přistoupit k jejich natrénování z námi dostupných dat. Trénování bude prováděno pomocí programu HERest, který je postaven na Baum-Welchově algoritmu. Pro mnohé může být tento algoritmus znám také jaké forward-backward algoritmus. Tento příkaz si zde přece ukážeme, neboť ho budeme využívat po celou dobu. Trénování provedeme pomocí příkazu:

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

Parametry programu:
-I phones0.mlf – Soubor s monofónní transkripcí, který jsme vytvořili na začátku.
-t 250.0 150.0 1000.0 – Nastaví práh prořezávání Baum-Welchova algoritmu.
-H hmm0/MODELS – Vstupní soubor s modely, které budou reestimovány.
-M hmm1 – Výstupní adresář, kam budou uložené reestimované modely.

Programu jsou tedy postupně předkládány jednotlivé věty z trénovacího seznamu *train.scp* a podle nich jsou trénovány jednotlivé modely. Tento krok je při trénování s HTK

obvykle nejvíce problematický, neboť jsou zde postupně procházeny všechna trénovací data a obsahují-li nějaký problém, pak na něj narazíme obvykle zde. Může se například stát, že v připravených datech se nějakou chybou při zpracování objeví například uvozovky, tečka uprostřed slova apod. Tento druh chyby se obzvlášť v ručně psaných datech objevuje velmi často a je dobré se pokusit tento problém odstranit při přípravě dat, neboť zahlásí-li nám program chybu, tak se velice těžko zjišťuje, které slovo nám dělá problémy. Pokud se však přes tento krok dostaneme, obvykle už nenarazíme v další části trénování na jiné problémy.

Pro zlepšení přesnosti parametrů je třeba reestimovat více než jednou. Pustíme tedy stejný program ještě třikrát pouze s tím rozdílem, že změníme postupně vstupní a výstupní složky. Po čtyřech reestimacích máme tedy co nejlépe přepočteny parametry modelů. Můžeme klidně reestimovat i vícekrát, avšak už při čtvrté reestimaci dojde obvykle pouze k nepatrné změně parametrů.

4.4. Přidání modelu krátké pauzy

Zatím byl během trénování použit pouze model dlouhé pauzy *_sil_*. Pro zlepšení akustického modelu by bylo vhodné umět rozlišit krátkou pauzu od dlouhé a také odchytit všelijaké šumy, které se v nahrávce objeví. Změníme tedy topologii modelu dlouhé pauzy a to tak, že přidáme přechody ze stavu 2 do stavu 4 a naopak. Dále vytvoříme model krátké pauzy, který bude mít pouze jeden emitující stav, který bude svázán s prostředním stavem dlouhé pauzy. Tyto dva modely tedy budou společně sdílet určité parametry.

Po těchto úpravách máme tedy vytvořené dva různé modely pauz, které nám mnohem věrohodněji modelují jednotlivé promluvy. Dále je potřeba opět provést třikrát reestimaci.

4.5. Přerovnání trénovacích dat

Dále se pokusíme zlepšit náš akustický model takzvaným přerovnáním trénovacích dat. Pod pojmem přerovnání si můžeme představit to, že program vytvoří transkripci na úrovni monofónů a pokud je ve slovníků na výběr více variant výslovnosti, vybere tu, která podle doposud natrénovaných modelů odpovídá nejlépe. Z tohoto důvodu jsme na začátku vytvářeli slovník *dict.txt*, který obsahuje vždy dvě varianty výslovnosti slov a to s krátkou a dlouhou pauzou.

Také se ale může stát, že se nepodaří některé věty přerovnat. To se může stát například při vysokém akustickém ruchu v pozadí, nebo mluví-li přes sebe více řečníků anebo je jednoduše věta špatně přepsaná. Je tedy potřeba upravit seznam trénovacích vět a vyhodit z něj ty, které přerovnáním neprošli.

Vytvořený soubor s větami, které se nepodařilo přerovnat, je dobré po vytvoření vždy zkontrolovat, neboť se může stát, že bude obsahovat velké množství vět a je potřeba zjistit proč. Může tomu být opravdu například z velkého akustického ruchu, avšak také se může stát, že při zpracování dat dojde k chybě a věty na sebe nesedí. Je tedy pak potřeba projít trénovací data a najdeme-li chybu, začít s trénováním od začátku. Nyní jsme tedy z trénování odstranili věty, které nám mohou modely trochu kazit a je potřeba opět provést reestimaci. Tu již provedeme s novým monofónovým přepisem a novým seznamem trénovacích vět.

Reestimaci provedeme čtyřikrát a výsledný model, nyní již finální, budeme mít uložen v adresáři hmmB.

4.6. Přidání trifónů

Pro zatím jsme pro modelování používali pouze nejjednodušší rozpoznávací jednotky a to monofóny. Chceme-li dosáhnout lepší úspěšnosti rozpoznávání, je zapotřebí přejít na modelování pomocí trifónů. Ty na rozdíl od monofónů využívají kontextu před i za fonémem. Nevýhodou trifónů je nutnost vyššího počtu trénovacích dat, avšak to je problém, který se dá částečně odstranit.

Chceme-li využít trifónů pro akustický model, využijeme již výše natrénovaný model, který máme uložený ve složce *hmmB*. Trénujeme-li trifónový model pomocí HTK musíme k tomu využít tento model a ne ten, který jsme získali následným přidáváním složek.

Stejně jako pro monofónový model, tak i pro trifóny bude vhodné si nejprve vytvořit referenční přepis na trifónové úrovni a k tomu adresáře *hmmt0* až *hmmt10* pro jednotlivé reestimace. Refrenční přepis vytvoříme pomocí:

*Hled -l * -n triphones0 -i crwtri.mlf mktri.led aligned2.mlf > hled.txt*

Kde, *triphones0* obsahuje všechny existující trifóny, *mktri.led* je soubor obsahující neřečové události promluvy a *crwtri.mlf* je výsledný soubor s transkripcí na úrovni trifónů. Ten může vypadat následovně:

#!MLF!#
"/poker0000veta0002.lab"*
*_sil_*
sil-h+e
h-e+s
e-s+k
s-k+l
k-l+v
*_sp_*
l-v+e
v-e+C

vidíme, že rozdíl oproti monofónovém přepisu je v kontextu u jednotlivých fónů. Poté je potřeba přidat trifóny také do našeho modelu. K tomu využijeme již nám známému programu HHed:

HHed -T 1 -C CF_trif.mfc -H hmmB\models -M hmmt0 mktri.hed monophones1.jpg > hhed.txt

kde *CF_trif.mfc* je konfigurační soubor pro tvorbu modelů. Nově vytvořený trifónový model pak máme ve složce *hmmt0*. Tento model je nyní potřeba reestimovat.

Herest -T 1 -C CF_trif.mfc -I crwtri.mlf -t 250.0 150.0 1000.0 -S aligned.scp -H hmmt0\models -M hmmt1 triphones0

Příkaz pro reestimaci je téměř stejný jako u monofónového modelu s tím rozdílem, že nyní využíváme nově vytvořeného trifónového referenčního přepisu a seznamu trifónů (*triphones0)* namísto seznamu monofónů. Model reestimujeme čtyřikrát a výsledek tedy máme uložený v *hmmt5*.

Dále se pokusíme odstranit hlavní problém trifónů a to nedostatek trénovacích dat. Vzhledem k tomu, že trifónů je obrovské množství, není příliš možné kvalitně natrénovat všechny. Toto úskalí vyřešíme pomocí slučování parametrů více trifónových modelů dohromady. Vycházíme z představy, že více trifónů lze považovat za téměř totožné vzhledem k akustickému kontextu a pomocí jednoho trifónů pak reprezentujeme celou skupinu. K tomu použijeme příkazu:

HHed -C CF_trif.mfc -H hmmt5\models -M hmmt6 tree.hed triphones0 > hhed.txt

Zde je klíčový parametr *tree.hed.* V něm nastavujeme práh a hloubku jak mají být trifóny slučovány. Zde musíme být velice opatrní, neboť nastavení těchto parametrů nám může velice ovlivnit celkovou úspěšnost rozpoznávání a to i o několik procent. Po vykonání tohoto příkazu už jen čtyřikrát reestimujeme a máme výsledný trifónový model, který můžeme dále zlepšovat přidáváním složek.

4.7. Rozpoznávací experiment

Pro zatím jsme tedy pouze trénovali akustický model postupným předkládáním trénovacích dat. Nyní se konečně dostaneme k ověření kvality akustického modelu a to k rozpoznávacímu experimentu. Zde stojí za zmínku to, že rozpoznáváme na jiných větách, než na kterých jsme natrénovali akustický model. Pro rozpoznávání využijeme opět program HVite, který jsme již výše použili také pro přerovnání dat.

4.7.1. Použití rozpoznávací sítě, jazykového modelu

Využíváme-li pro rozpoznávání rozpoznávací síť, musíme si ji nejprve připravit. V nejjednodušším případě síť vytvoříme tak, že do ní dáme pouze slova, která obsahují námi testované promluvy. Síť nám pak bude udávat, s jakou pravděpodobností může dané slovo být po jiném. Samotné rozpoznávání pak pustíme pomocí příkazu:

*HVite -C CF.mfc -H hmmB/models -S test.scp -i výsledek_all.txt -l * -p -60.0 -w wdnet dict.txt monophones1*

Parametry:–
hmmB/models – Složka s výslednými modely.
-S test.scp – Seznam testovacích vět.
-i výsledek_all.txt – Výsledek, který nám dá rozpoznávač.
*-l **- Způsobí, že do jmén souborů v MLF je vložen symbol * místo cesty k souboru.
-p – 60.0 – Tímto parametrem volíme volbu penalizace, čímž můžeme změnit váhu rozpoznávání krátkých slov.
-w wdnet – Rozpoznávací síť.
dict.txt – Zdvojený výslovnostní slovník.
monophones1 – Seznam všech monofónů.

V souboru *výsledek_all.txt* tedy nyní máme výsledek rozpoznávání. Výsledky však nejsou v úplně nejlepší podobě. Pro jejich zpřehlednění využijeme opět program ze sady HTK Hresults. Po jeho zavolání dostaneme výsledek, s jakou úspěšností byly rozpoznány jednotlivé věty. K tomu program využívá referenčního přepisu *words.mlf*, ve kterém je obsaženo, co má být rozpoznáno a porovnává to s reálnými výsledky rozpoznávače. Program pak pustíme v následující podobě:

Hresults -f -I words.mlf monophones1 vysledek_all.txt > vysledek.txt

kde v souboru *vysledek.txt* máme úspěšnost jednotlivých vět. V praxi může výsledek vypadat například takto:

---------- File Results ----------
poker0000veta0002.rec: 91.67(75.00) [H= 11, D= 0, S= 1, I= 2, N= 12]
poker0000veta0003.rec: 89.47(89.47) [H= 17, D= 0, S= 2, I= 0, N= 19]
poker0000veta0004.rec: 83.33(83.33) [H= 5, D= 0, S= 1, I= 0, N= 6]
poker0000veta0005.rec: 89.47(89.47) [H= 17, D= 2, S= 0, I= 0, N= 19]
poker0000veta0006.rec: 83.33(83.33) [H= 5, D= 1, S= 0, I= 0, N= 6]
poker0000veta0007.rec: 93.75(87.50) [H= 15, D= 1, S= 0, I= 1, N= 16]
poker0000veta0009.rec: 100.00(100.00) [H= 17, D= 0, S= 0, I= 0, N= 17]
poker0000veta0011.rec: 70.59(70.59) [H= 12, D= 5, S= 0, I= 0, N= 17]

---------- Overall Results ----------
SENT: %Correct=45.00 [H=45, S=55, N=100]
WORD: %Corr=94.63, Acc=92.56 [H=1322, D=33, S=42, I=29, N=1397]
==========

Na poslední řádce příkladu vidíme, že soubor byl rozpoznán s více než 90 procentní úspěšností.

4.8. Přidání složek do akustického modelu

V adresáři hmmB tedy máme pro zatím nejlepší verzi monofónového akustického modelu. Další způsob, jak zlepšit jeho kvalitu je postupným přidáváním složek. K tomu opět využijeme nástroj ze sady HTK:

HHEd -T 1 -A -C CF.mfc -H hmmB/models -M hmm21 add_next.hed monophones1

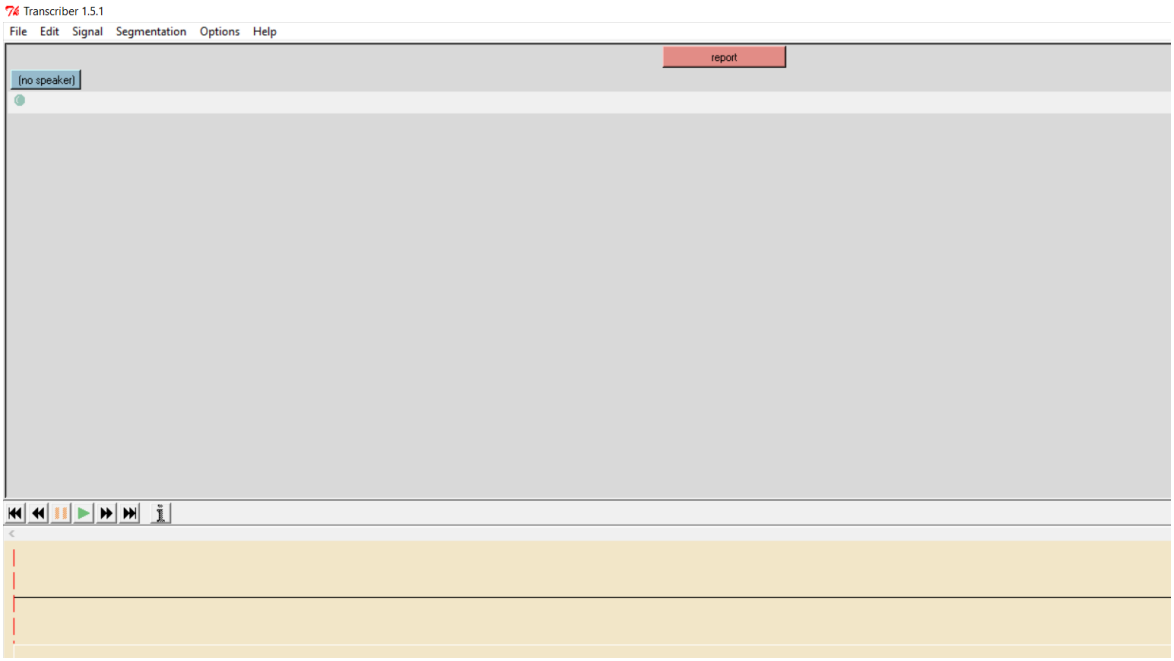
kde soubor *add_next.hed* obsahuje řetězec *MU +1 {*.state[2-4].mix}*, který nám značí přidání složky do modelu. Výsledný model pak bude uložen ve složce *hmm21* a bude potřeba ho opět čtyřikrát reestimovat. Složky přidáváme do modelu tak dlouho, dokud se zlepšuje úspěšnost při rozpoznávání. Je tedy vhodné po jednotlivých reestimacích vždy provést rozpoznávací experiment na testovacích větách, abychom věděli, zda-li stále roste úspěšnost a má tedy cenu přidávat další složky. Přidáváním složek dojde obvykle k velkému zlepšení úspěšnosti. Jediná nevýhoda přidávání složek je postupné prodlužování jednotlivých reestimací a máme-li velké množství trénovacích dat, může jedna reestimace trvat i několik hodin.

5. Tvorba a návod k přepisu

Veškerá data, ze kterých se v této práci trénují akustické modely, jsou ručně anotovaná data. Ty jsou vytvářeny pomocí programu Transcriber a vzhledem k tomu, že jsou základním stavebním kamenem diplomové práce, bylo by nyní dobré si ukázat, jak vlastně vznikají.

5.1. Základní vlastnosti programu Transcriber

Pro anotaci nahrávek byl tedy využit program Transcriber. Dalo by se jistě najít plno dalších programů, avšak Transcriber je volně dostupný na internetu, je velmi jednoduše ovladatelný a je tedy pro tyto účely ideální. Po jednoduché instalaci se na ploše vytvoří ikona programu, pomocí které budeme program pouštět. Po spuštění programu uvidíme následující rozhraní:



Obrázek 4 rozhraní Transcriberu

V horní části vidíme několik základních záložek. V záložce *file* najdeme možnosti pro začátek nového přepisu, otevření již vytvořeného přepisu, dále pak možnost pro otevření zvukové stopy a nakonec záložku *edit episode attributes*, která slouží především k zapsání jména přepisovače, aby bylo poznat, kdo přepis napsal.

V další záložce *edit* jsou nejprve možnosti pro kopírování, vkládání a hledání v textu. Tyto možnosti se dají samozřejmě využívat, ale rychlejší je využít klasických klávesových zkratek. Dále zde najdeme možnosti pro vytváření a hledání témat případně řečníků.

Jak vyplývá z názvu další záložky *segmentation*, tak ta slouží k segmentaci textu na menší části. Opět je možno využít také klávesové zkratky, ale o tom si ještě povíme později.

Předposlední záložka *option* pak obsahuje nejvíce možností. Dá se zde nastavit interval automatického ukládání, který se občas může velice hodit. Dále zde můžeme změnit typ a velikost písmen nebo barvy jednotlivých částí programů. Pak zde najdeme také velice užitečnou záložku *bindings*, kde se dají nastavit klávesové zkratky. Toto se hodí především pro přepisování, kdy značíme jména do různých závorek, neboť psát pro každé jméno otevřenou a uzavřenou složenou případně hranatou závorku je velmi zdlouhavé. Poslední možností v této

záložce pak je možnost načtení konfiguračního souboru. To je důležité pro přepis českého jazyka, neboť pokud nenačteme konfigurační soubor, program nedokáže správně rozpoznávat česká písmena jako například *ř, č* nebo *š* a při znovuotevření souboru bychom pak místo těchto znaků viděli otazníky.

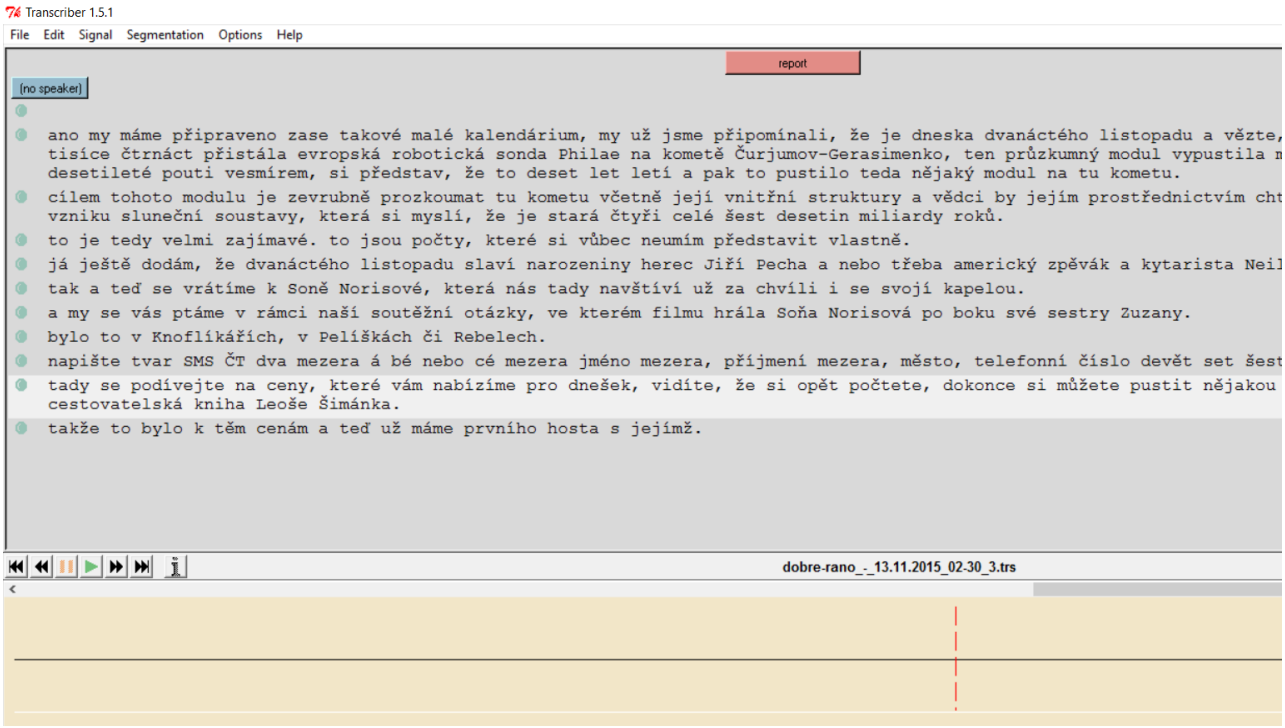
V prostřední části programu pak máme prostor, kde se nám bude postupně objevovat přepsaný text. Také zde je tlačítko *no speaker*, kde můžeme vždy označit mluvčího, který v danou chvíli mluví.

V dolní části obrazovky pak je místo, kde při otevření bude zvuková stopa. Nad ní je pak malá nástrojová lišta s možnostmi pro zastavení a přehrání zvuku, přetočení zvukové stopy o kousek anebo pak ikonka s informacemi, kde můžeme vidět například počet přepsaných slov nebo počet řečníků v celém přepisu.

5.2. Jak přepisovat

Po vysvětlení základních funkcí programu se můžeme pustit do samotného přepisování. Klikneme tedy na možnost *file,* dále pak New Trans a vybereme zvukový soubor, který chceme přepsat. Ihned po načtení nahrávky se nám v dolní části tedy objeví zvuková stopa programu. Tu si můžeme libovolně přiblížit případně oddálit, avšak mně se jako nejlepší ukázala volba, kde je na stránce vidět přibližně půl minutová stopa. Toto je důležité především proto, když při přepisování přeslechneme nějaké slovo a chceme se vrátit v čase, tak stačí kliknout na zvukovou stopu a pustit přehrávání. Samotnou promluvu pak přepisujeme víceméně stejně, jako kdybychom psali normální text snad s pár rozdíly. Ty budou nyní zmíněny spolu se základními klávesovými zkratkami pro ovládání programu:

1) Pro spuštění zvukové stopy můžeme používat tlačítko *play*. Mnohem rychlejší je však využívat klávesu *Tab*, která při stisknutí spustí nebo zastaví zvuk.
2) Psaný text dělíme do segmentů. Ty by neměli být delší než 4 řádky. Obvykle se snažíme, aby každý segment obsahoval jednu větu, avšak může se stát, že komentátor mluví dlouho v kuse a je potřeba pak tedy větu rozdělit uměle. Segmenty vytvoříme pomocí klávesy *Enter* a mažeme pak pomocí *Ctrl + Enter*.
3) Občas se stává, že nám v jednom segmentu mluví přes sebe dva komentátoři. V takové situaci nejprve napíšeme, co řekl jeden, pak napíšeme tečku a za ní zapíšeme promluvu druhého komentátora. Zvláštní situace také může nastat v případě, že se dlouho nemluví nebo nerozumíme řeči, která byla vyřčena. V takové situaci pak využijeme takzvaný prázdný segment. To znamená, že celý časový úsek, kde k takové situaci dojde, bude segment, který nebude obsahovat žádný text.
4) Oproti klasicky psanému textu je zde rozdíl, že věta nezačíná velkým písmenem. Velká písmena píšeme pouze v případě jmen, zkratek a podobně.
5) Číslovky píšeme vždy slovně.
6) Přepis se snažíme psát spisovně. Narazíme-li tedy na situaci, kdy komentátor mluví nespisovně, pokusíme se přepis mírně modifikovat. Typická situace je, když slovo *kterej* napíšeme jako *který.*



Obrázek 5 Ukázka jednoduchého přepisu

Na obrázku můžeme vidět přepis v jeho nejjednodušší podobě. Vidíme, že věty začínají malým písmenem a jsou rozdělené do jednotlivých segmentů. Na začátku přepisu pak vidíme prázdný segment.

5.3. Třídní přepis

Pro zatím jsme si ukázali, jak psát jednoduchý přepis. Často ale narazíme na případ, kdy je potřeba přejít na třídní přepisování. Vzhledem k tomu, že diplomová práce se zaměřuje na sportovní přenosy, tak v tomto případě to znamená, že budeme jednotlivá jména sportovců, sportovišť nebo národností dávat do jednotlivých závorek. Toto zesložitění se dělá z důvodu využití při jazykovém modelování. Kde díky značení jmen v přepisech, pak stačí nalézt aktuální soupisku pro dané utkání, která je vyskloňována a vložena do jednotlivých kategorií, a díky tomu získáme velice kvalitní třídní jazykový model. U každé kategorie bude také tedy uveden pád, ve kterém se dané jméno nachází. Veškerá jména pak můžeme rozdělit do tří hlavních kategorií.

Do první a také nejvíce používané spadnou veškerá jména sportovců. Tato kategorie bude obsahovat tedy jména sportovců, trenérů, manažérů a podobně, které mají co dočinění s daným sportem. Zde je důležité, abychom do závorek nedávali obecná jména, jako například když při moderování hokeje komentátor mimoděk zmíní jméno prezidenta nebo zpěvačky, která v hale zpívala hymnu a podobně. Tuto kategorii budeme dávat do kulatých závorek. Příklad: *to byla nádherná přihrávka na (Jaromíra Jágra 4), kterou na něj vyslal (Pavel Patera 1), škoda že rozhodčí (Fraňo 1) zapískal offside.*

Druhá kategorie a nejméně používaná bude kategorie sportovišť. Tato kategorie se téměř vždy objeví na začátku přepisu, kdy komentátor vítá diváky v dané aréně, sportovišti a pak je občas zmíněna v průběhu přenosu. Opět je zde důležité řadit sem pouze sportoviště

související s danou událostí. Tato kategorie bude v hranatých závorkách. Příklad: *dobrý den, tady je (Robert Záruba 1), vítám vás na zimních olympijských hrách ve [Vancouveru 6], které zatím vypadají hůře než ty v [Athénách 6].*

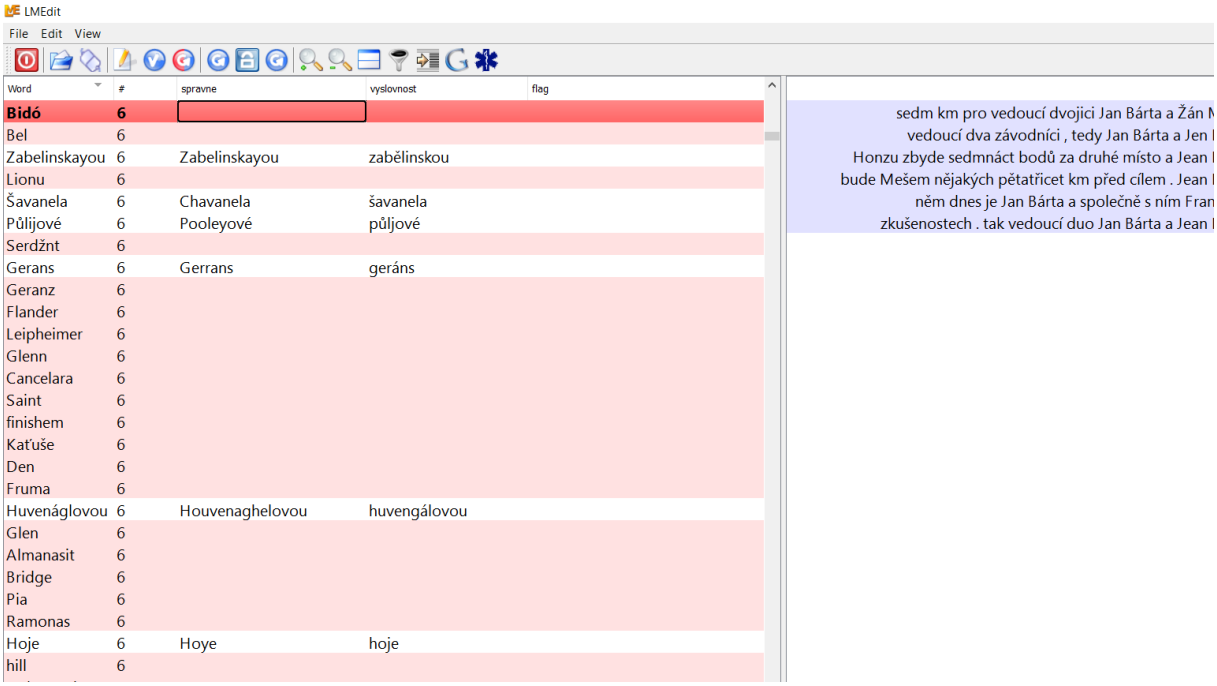
Poslední kategorií pak bude kategorie klubová. Zde budeme řadit všechna jména klubů, jejich „přezdívky“ (viz. Bílý Balet) nebo národnosti jednotlivých hráčů. Značit ji pak budeme pomocí složených závorek. Příklad: *{Real Madrid 1} jinak přezdívaný {Bílý Balet 1} tedy roznesl {italský 4} klub pět jedna.*

Poslední možností, kterou můžeme v přepisování využívat je značení řečníků. Toto je typické u většiny sportů, kde obvykle zápas nebo závod nekomentuje pouze jeden komentátor, ale obvykle zde najdeme jednoho stálého moderátora a druhého hosta, který také přispívá svými poznatky. Z vlastní zkušenosti vím, že přepisovat zkušeného moderátora je obvykle mnohem snazší, neboť mluví daleko spisovněji a zřetelněji než přizvaný host.

Chceme-li tedy označit řečníka, klikneme na modré tlačítko vlevo nahoře, pro zatím označené jako *no speaker*. Zde vyplníme jméno moderátora, který se nám obvykle představí ze začátku pořadu, a klikneme na tlačítko *ok*. Pokud se nám pak objeví další řečník někdy v průběhu přepisu, nemusíme vyjíždět v programu na začátek a hledat tlačítko na vytvoření, ale můžeme využít další klávesové zkratky *Ctrl+t,* která nám opět ušetří práci a plní stejný účel. Značení řečníků je obvykle bezproblémové při přepisování sportovních přepisů. Avšak dostaneme-li se k přepisování nějakých diskuzních pořadů, kde často dochází k přeměně řečníků a často ani nevíme, kdo mluví, pak není až tak důležité znát jména řečníka, ale je důležité stejnému hlasu vždy přiřadit stejný název řečníka.

5.4. Kontrola přepisů

Nejlepší volba, jak zkontrolovat daný přepis, je pustit si ho opět od začátku a porovnávat to, co jsme přepsali s tím, co je řečeno. Další a velice účinná možnost kontroly je pomocí programu Word. To uděláme jednoduše tak, že zkopírujeme celý přepis a vložíme ho do Wordu. Ten nám pak podtrhává chyby, kterých bychom si nevšimli, ani kdybychom přepis četli několikrát. Nejčastěji to jsou překlepy typy *sand* místo *snad* anebo pak různé gramatické chyby. Posledním způsobem kontroly je pak hromadná kontrola pomocí programu LMEdit. Ta probíhá tak, že jsou veškeré přepisy pro daný sport „prohnány“ nějakým souborem známých a zkontrolovaných slov. Každé slovo, které se pak v tomto seznamu nevyskytne, obvykle se jedná o jméno sportovce, překlep nebo anglické slovo, je pak vyextrahováno na základě jeho četnosti do speciální databáze. Ta může vypadat například následovně:



Obrázek 6 Příklad opravování slov pomocí LMEditu

Z uvedeného příkladu vidíme, že v prvním sloupci se vyskytují neznámá slova, ve druhém jejich četnost, ve třetím pak správný tvar, který už musíme doplnit my a ve čtvrtém jejich výslovnost. Pátý sloupec je pak specifický a vyplňuje se pouze v případě, že dané slovo buď to neznáme, pak do něj vložíme písmeno *x* anebo pak písmenko *m* jako multislovo jedná-li se o jméno, které se vyskytuje pospolu. Tedy například *Sparta Praha* nebo *David Lafata*. V pravé části programu pak máme vždy uvedený kontext slova pro všechny jeho výskyty. Ten může přijít velice vhod, pokud se jedná například o překlep, který bychom samotný nikdy neidentifikovali, avšak z kontextu nám snad vyplyne.

Program LMEdit je jinak opět velice intuitivní a má několik zajímavých vychytávek jako je například druhá ikonka zprava a tedy znak Googlu. Při jeho stisknutí se nám neznámé slovo automaticky otevře v internetovém prohlížeči v Googlu a především při hledání správného tvaru jména je toto velice užitečné.

6. Praktická část

Praktická část této práce se skládala z několika kroků. Nejprve tedy bylo potřeba přepsat data pro jednotlivé sporty. Na jejich přípravě jsem se podílel již od studia na střední škole, byl to konec konců i jeden z důvodu studia na této škole, protože mi tato práce ohledně rozpoznávání řeči a obrazu přišla zajímavá a i velice záslužná. Příprava přepisů na diplomovou práci byla časově neuvěřitelně náročná a dělat ji sám během pouze navazujícího studia by nebylo v mých silách. Máme-li například hodinový přepis o obsahu přibližně 4000 slov, jeho přepsání nám zabere okolo 4 hodin. Pro kvalitní akustický model je pak dobré mít ideálně okolo 100 hodin přepisů. Takže pouze připravit přepisy pro jeden jediný sport trvá okolo 20 dní čistého času.

Jednotlivé přepisy bylo potřeba dále zkontrolovat pomocí slovníků a následně zpracovat a natrénovat na každém sportu zvlášť jednoduché akustické modely. S tímto postupem jsem se seznámil již při tvorbě bakalářské práce, kde jsem však provedl trénování pouze na několika sportech. Aby bylo na diplomovou práci z čeho vybírat, bylo zapotřebí zpracovat veškeré sporty, které se v České televizi vysílají. Tato část práce byla opět časově velice náročná a pracoval jsem na ni po celou dobu od ukončení bakalářské práce, neboť sportů, které se v České televizi vysílají je několik desítek. Vzhledem k tomu, že je potřeba zpracovat ručně psaná data, která obsahují bohužel chyby, a to neustále jiné, jednalo se opět o časově velice náročný úkol. Občas zde samozřejmě byl sport, který nebyl problém zpracovat během dne nebo dvou, jako například poker nebo alpské lyžování. Obvykle tomu tak bylo u sportů, která měla k dispozici méně dat a platilo zde tedy méně dat, méně překvapení a chyb. Ale taky zde byly bohužel sporty jako například hokej, kde mi zpracování dat a natrénování jednoduché akustického modelu zabralo pár týdnů.

Po zpracování jednotlivých sportů bylo dále potřeba vybrat reprezentanty, na kterých bude provedeno další rozpoznávání a u kterých by mohlo dobře fungovat rozpoznávání z přímé zvukové stopy. Bylo tedy nutné provést analýzu vysílacího schématu České televize a najít a vybrat vhodné reprezentanty pro přímé titulkování.

Pro všechny vybrané sporty bylo pak nutné ověřit na reálném vysílání jejich výsledky. K tomu bylo potřeba projít archiv České televize, najít vhodnou nahrávku z nedávné doby o délce alespoň 30 minut a tu opět přepsat, abychom měli s čím porovnávat rozpoznané výsledky, a ověřit tedy úspěšnost rozpoznávání jednotlivých sportů. Pro diplomovou práci jsem se rozhodl využít trifónový GMM model. Pro dosažení co nejlepších výsledků a nasazení systému do praxe by bylo samozřejmě vhodné použít také DNN (deep neutral network) a nebo TDNN (time delay neutral network), avšak pro vytipování sportů pro přímé titulkování je použitý trifónového GMM modelu naprosto dostačující.

6.1. Příprava a první natrénování akustických modelů

Pro každý sport bylo tedy nejprve potřeba natrénovat jednoduchý jednosložkový model. Na začátku byly pro jednotlivé sporty vždy k dispozici všechna přepsaná data spolu s jejich zvukovou stopou a dále zpracované slovníky. Tato data byla poskytnuta Fakultou aplikovaných věd Západočeské univerzity v Plzni a na jejich přípravě jsem se tedy dlouhá léta podílel a znal jsem tak velice dobře jejich formát a chyby, se kterými se zde mohu setkat.

6.1.1. Formát přepisů a slovníků

Ze začátku se tedy vycházelo z ručně přepsaných dat v Transcriberu, které byly všechny v podobném, bohužel ne ve stejném tvaru. Zde můžeme vidět, jak vypadaly jednotlivé přepisy při otevření v libovolném textovém editoru.

```
<?xml version="1.0" encoding="CP1250"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="Ondřej Váchal" audio_filename="Box-AUS-FRA" version="1"
version_date="120111">
```

<Episode program="" air_date="">
<Section type="report" startTime="0" endTime="518.28025">
<Turn startTime="0" endTime="518.28025">
<Sync time="0"/>
vítejte v této hale, která byla postavena nejdříve pro mistrovství světa v ping pongu, boxovali (Fleming Paul 1) a (Kedafí Dželkijer 1).
<Sync time="12.142"/>
{Australan 1} proti {Francouzovi 3}, my jsme se dostali do druhého kola, po prvním kole vidíte stav tři jedna ve prospěch boxera v červeném {Australana 2} (Pola Fleminga 2).
<Sync time="498.009"/>
další z výměn v posledních vteřinách a ještě takovou parádičku si připravil (Kedáfi Dželkijer 1), konec utkání třináct devět zvítězil {francouzský 1} boxer.
</Turn>
</Section>
</Episode>
</Trans>

Na prvních dvou řádkách je uvedeno použité kódování a verze xml. Na třetí řádce je pak napsáno jméno člověka, který daný soubor přepisoval, v tomto případě moje, a dále pak jméno souboru. Na páté řádce pak vidíme začátek a konec přepisu, kde časy jsou uvedeny v milisekundách. Především konečný časový údaj byl velice důležitý, neboť byl potřeba pro nařezání zvukové stopy, ale k tomu se ještě dostanu. Zde jsem však narazil na problémy, neboť jak je zmíněno výše, přepisy nebyly vždy ve stejném formátu. Můj program tedy původně bral natvrdo z páté řádky časový údaj o konci přepisu. To fungovalo pro 90 procent dat, avšak některé přepisy obsahovaly také informace o jednotlivých řečnících a zde pak začátek vypadal trochu jinak a program tak musel být upraven. Další část přepisu pak byla víceméně stejná a vždy jeden řádek obsahoval informaci o začátku daného segmentu, druhý pak co v něm bylo řečeno a třetí konec segmentu a začátek dalšího. Konec přepisu pak poznáme podle uzavření jednotlivých atributů vždy ukončeno pomocí *</Trans>.*

Dále pak byly k dispozici jednotlivé slovníky vytvořené v programu LMEdit, které opět při otevření v libovolném textovém editoru byly v následujícím tvaru:

|<s> <s>|Padlý|anděl </s> </s>| Padlý anděl m

|<s> <s> MBS|Olympics|</s> </s>| MBS Olympics em bí es olympiks m

|<s> <s> David|Vincouver|</s> </s>| David Vincour m

|<s> <s> Jevgenin|Plato|</s> </s>| Evgeni Platov jevgenij platov m

|<s> <s>|Clintna|Mansela </s> </s>| Clinta Mansella klinta manžela m

Vidíme, že jednotlivé řádky slovníku obsahovali vždy původní slovo, za co bylo nahrazeno a jeho výslovnost.

6.1.2. Vyčištění a oprava dat

Tato data byla tedy k dispozici na začátku a bylo potřeba je zpracovat. Nejprve bylo vhodné získat z dat pouze důležité informace. Byl tedy vytvořen program, který ze slovníků vždy vytáhne pouze špatný výraz, jeho správná náhrada a jeho výslovnost. Jednotlivé řádky tedy vypadaly následovně:

Ahjo ahoj
akrát akorát
ažž až aš
sand snad snat

Vidíme, že slovník občas obsahuje dva sloupce a občas tři. To z toho důvodu, že u slov se stejnou výslovností, jako je jejich správný tvar, nemusel být třetí sloupec uveden. Podobným způsobem byly pak vyčištěny všechny přepisy a byly z nich odstraněny jednotlivé závorky a čísla označující pády. Poté co byla data vyčištěna, mohlo se přejít k dalšímu kroku. Nejprve bylo potřeba vytvořit program, který postupně prochází jednotlivé řádky v přepisech a jednotlivé řádky ve slovníků a hledá mezi nimi shodné výrazy. Narazí-li na shodu a tedy špatný výraz, provede jeho náhradu za správný z daného slovníku. Provádět náhrady slovo za slovo bylo velice jednoduché, musel zde ale být vyřešen problém, kdy bylo potřeba provést náhradu *m* slov za *n* slov. Tím se však velice zvětšila výpočetní náročnost a nakonec jsem tento problém ořízl na hledání náhrad všech dvojic, trojic a čtveřic, neboť delší výrazy se ve slovníkách nevyskytovaly a ušetřil jsem ti značně čas. Máme-li totiž k dispozici slovník o délce 50 000 slov a například přepis o délce 15000 slov a procházíme v přepisu všechny možné délky slov a hledáme jejich shodu ve slovníku, tak se dostáváme k desítkám miliónům operacím. A máme-li pak takových přepisů 300 pro každý sport, začíná to být problematické.

Tyto náhrady se tedy vždy provedly dvakrát. Nejprve jedním univerzálním slovníkem, který obsahoval desetitisíce slov a pak ještě slovníkem pro daný sport, který vždy vycházel z daných přepisů a bylo potřeba ho před samotným trénováním připravit. Připravit tedy data byť jen pro jediný sport byla otázka stovek hodin. Neboť data musela být nejprve přepsána, pak z nich byl vytvořen slovník neznámých slov a pomocí něj pak tedy byly přepisy upraveny. V bakalářské práci jsem se již zabýval problematikou, jaký vliv má na samotnou úspěšnost rozpoznávání aplikace těchto náhrad a i na jednoduchém jednosložkovém monofónovém modelu byly rozdíly velmi výrazné. Například u házené se úspěšnost zlepšila z 33% na 36%.

6.1.3. Vytvoření referenčního přepisu a výslovnostního slovníku

Nyní tedy byly přepisy ve stavu, kdy byly zbavené chyb, a mohlo se přistoupit k dalším krokům. Přepisy bylo potřeba dále rozdělit na část, která obsahovala pouze informace o začátcích a koncích promluv a pak na čistý obsah promluv. Ten byl v následujícím formátu:

a včetně českého reprezentanta Rastislava Vítka
premiéra i pro nás jako komentátorskou dvojici na plavání
zdravím na komentátorském stanovišti Martinu Kaplanovou
dobrý den a start je klíčovým momentem takového závodu protože plavec…

vidíme, že každá řádka obsahuje jednu větu daného přepisu.

Zaměříme-li se dále na obsah promluv, tak ten byl potřeba pro vytvoření referenčního přepisu pro rozpoznávání a výslovnostního slovníku. Ten vytvoříme tak, že pomocí jednoduchého skriptu zašleme soubor na internetovou stránku https://services.speechtech.cz/welcome/utils/text . Jedná se o stránku, která denně sbírá obrovské textové korpusy z internetových serverů. Díky tomu je pak systém schopen vytvořit velice aktuální, robustní a také spolehlivý jazykový model (podrobněji [14]). Ta nám po chvilce vrátí tři soubory. Dva z nich se týkají výslovnostního slovníku a mají koncovku *check* a *ok*. Soubor s koncovkou *check* obsahuje slova, která jsou v programu zadefinována jako neznáma a je potřeba jejich výslovnost zkontrolovat, neboť jejich výslovnost byla vytvořena na základě pravidel českého jazyka. Druhý soubor s koncovkou *ok* pak obsahuje výslovnost všech ostatních slov, kterých je naštěstí většina. Tyto soubory mají pak následující tvar:

lamy	*l a m i*
sami	*s a m i*
maty	*m a t i*
od	*o t*
od	*o d*
hanu	*h a n u*
alp	*a l p*
alp	*a l b*
devon	*d e v o n*
nevel	*n e v e l*

z formátu vidíme, že v levém sloupci jsou uvedena jednotlivá slova a v pravém sloupci jejich výslovnost. Třetí soubor, který nám vytvoří internetová stránka, pak obsahuje jednotlivé věty oddělené znakem |.

Nyní jsou k dispozici veškerá data pro vytvoření finálního výslovnostního slovníku a referenčního přepisu. Nejprve se zaměříme na vytvoření druhého zmíněného souboru a to vytvoření referenčního přepisu značeného při rozpoznávání jako *words.mlf.*

Zde bylo zapotřebí opět vytvořit jednoduchý skript, který nám načte poslední soubor vygenerovaný internetovou stránkou a to soubor obsahující jednotlivé věty. Z něj jsou postupně načítány jednotlivé věty a vždy když se objeví znak |_ tak víme, že se jedná o novou větu. Každá věta a přepis je pak značena unikátně. Máme-li například sport plavání, pak první věta v prvním přepisu bude označeno takto: *plavani0000veta0000.* Zde jsem při programování kódu ze začátku nevěděl, jak odlišit jednotlivé přepisy, neboť na webovou stránku jsem posílal pouze jeden soubor, který obsahoval všechny věty ze všech přepisů. Nakonec jsem tento problém vyřešil pomocí unikátního textového identifikátoru, v mém případě *ERROR_444*. Tento znak jsem vždy vložil na konec každého přepisu a při následném vytváření referenčního přepisu jsem pak díky tomuto vždy věděl, že končí jeden a začíná druhý přepis a tento znak jsem odstranil.

Při vytváření referenčního přepisu jsem si také vytvářel testovací sadu o rozsahu 100 vět. Ačkoliv se jednalo pro zatím pouze o tvorbu jednoduché monofónového akustického

modelu, tak i přes to bylo dobré vědět, že vše funguje tak, jak má. Zde jsem se snažil, aby věty byly vybrány vždy z celého rozsahu přepisů. To z toho důvodu, že prvních několik přepisů mohlo být namluveno jedním řečníkem a vybrat pouze tyto věty, mohly by být výsledky velice dobré, avšak při vybrání vět ze všech přepisů, můžeme dostat zcela odlišné výsledky. Tyto výsledky pro mě byly důležité i z toho důvodu, abych věděl, jak kvalitní jsou jednotlivé akustické modely u každého sportu. Obvykle výsledky potvrdily mé očekávání, avšak občas jsem byl výsledky překvapen ať už pozitivně nebo negativně.

Dále se zaměříme na tvorbu výslovnostních slovníků. Jak je již výše zmíněno po zaslání obsahu přepisu na webovou stránku dostaneme také soubory s koncovkou *ok* a *check*, ze kterých nyní vytvoříme výslovnostní slovníky. Nejprve je potřeba zkontrolovat soubory ze slovníku *check*, neboť zde vytvořená slova mohou často obsahovat nesprávnou výslovnost a to především u cizích slov. Navíc pokud slova nezkontrolujeme, obvykle pak narazíme také na chybu při použití nástrojů HTK, neboť zde budeme mít monofóny, které HTK nezná a obvykle také neexistují. Pro upřesnění si ukážeme příklad ze slovníku check:

Osace	*o s a c e*
Maldonádovi	*m a l d o n A d o v i*
Kyrgyzchstánu	*k i r g i s x s t A n u*
Mendeánová	*m e n d e A n o v A*
Afrického	*a f r i c k E h o*
te'd	*t e ' d*

z uvedeného příkladu můžeme vidět, že ve slovníku se vyskytují většinou jména anebo překlepy. Podíváme-li se na poslední slovo *te'd*, tak zde se přesně dostáváme do situace, kdy bychom při použití nástroje HTK narazili na problém. To proto, že se nám ve výslovnosti objevuje fón „'" a ten samozřejmě neexistuje. Vzhledem k tomu, že při velkém objemu přepisů jako například u fotbalu nebo u hokeje, kdy bylo k dispozici stovky hodin dat, dosahovala velikost slovníku *check* tisíce a tisíce slov, byl jsem nucen vytvořen další skript, který mi tento problém pomohl odstranit. Vytvořil jsem si soubor, do kterého jsem nadefinoval všechny existující fóny, se kterými se můžeme setkat, ten jsem načetl a porovnával vždy s jednotlivými slovy. Pokud dané slovo obsahovalo znak, který nebyl uveden v mém souboru, toto slovo se mi vypsalo a bylo nutné slovo najít a opravit a to i v referenčním přepisu. Pokud bych tento krok neudělal, docházelo by při první reestimaci akustického modelu k postupnému procházení jednotlivých vět a vždy když by program narazil na cizí slovo, tak by spadl. Toto bylo možné u menších sportů, které obsahovaly řádově několik hodin dat, ale jakmile jsem dorazil datově rozsáhlejším sportům, musel jsem tento problém odstranit jinak než pouze metodou pokus-omyl.

Po odstranění těchto neznámých znaků jsem se konečně mohl vrhnout k vytvoření výslovnostního slovníku. Opravená slova ze souboru *check* a slova ze souboru *ok* jsem vložil do jednoho souboru a ten jsem poté načetl do programu. Dále stačilo nadefinovat další soubor pro zápis, na jehož začátek byly vloženy kvůli programu HTK tři pevně nadefinované řádky, viz

příklad, a dále byly postupně procházeny jednotlivé řádky ze souboru a na jejich konec byl přidáván znak pauzy *_sp_*. Výslovnostní slovník značený v HTK *dict.sp* pak mohl vypadat následovně:

*_END_ [] _sil_*
*_SIL_ [] _sil_*
*_START_ [] _sil_*
*plus p l u s _sp_*
*a a _sp_*
*a A _sp_*
*se s e _sp_*
*na n a _sp_*
*v v E _sp_*
*v f _sp_*
.....

Stejným způsobem bylo potřeba také vytvořit výslovnostní slovník *dict.txt*. Ten kromě pauzy *_sp_* obsahuje také pauzu *_sil_*, aby si program mohl vybrat, kterou pauzu použije, zda-li krátkou nebo dlouhou. Tento slovník byl vytvořen podobným způsobem a to tak, že jednotlivé slovo bylo vždy použito dvakrát a bylo tedy na jednom řádku s pauzou *_sil_* a na druhém řádku s pauzou *_sp_*.

Ze začátku mé práce byly po tomto kroku výslovnostní slovníky ve finální podobě. Po vypracování bakalářské práce jsem však experimenty zjistil, že je do výslovnostního slovníku vhodné přidat ještě výslovnosti, které máme uvedené ve slovníku z programu LM Edit. Tyto slovníky obsahovali tedy tři sloupce, první se špatným výrazem, druhý jeho správnou náhradu a třetí pak výslovnost výrazu. Tento třetí sloupec bylo potřeba vyextrahovat a obdobně jako obsah přepisů ho zaslat skriptem na webovou stránku. Tím jsme dostali výslovnosti slov v podobě, kterou potřebujeme pro přidání výrazů do výslovnostního slovníku. Nyní však bylo ještě potřeba slova zpět spárovat s jejich původním výrazem pro upřesnění:

pojjď pojď pojť

zde můžeme vidět jednotlivý řádek slovníku, po zaslání výslovnosti na webovou stránku dostáváme:

pojť p o j T

a tento výraz je pak potřeba spojit s druhým sloupečkem našeho původního slovníku a výsledek pak vypadá následovně:

pojď p o j T

takovéto výrazy pak spolu se znaky krátkých a dlouhých pauz přidáme do obou výslovnostních slovníků *dict a dict_sp*. Tím konečně dostáváme finální podobu těchto souborů.

6.1.4. Zpracování zvukové stopy

Z obsahové části přepisů tedy získáme referenční přepis *words*.mlf a výslovnostní slovníky *dict_sp.txt a dict.txt.* Pro správné nařezání zvukové stopy potřebujeme z přepisů dostat jednotlivé informace o začátcích a koncích jednotlivých segmentů. Tato hodnota je uváděna v sekundách, jak můžeme vidět na příkladu:

<Sync time="52.015"/>
v dráze číslo jedna v tomhle finále uvidíme domácí závodnici Hanah Majliovou
<Sync time="59.791"/>
ve dvojce Maďarka Katinka Hosuová mistryně Evropy na krátké polohovce angažmá
<Sync time="84.161"/>
ve čtyřce Ješven závodně plave od roku dva tisíce sedm to znamená pět let ona sama tvrdí že její výkony vycházejí z vědeckého přístupu čínských trenérů k plavání
<Sync time="103.987"/>
….

Našim účelem tedy bude získat z přepisu začátek a konec každé věty a k tomu je ještě potřeba si každou větu unikátně pojmenovat. Toto pojmenování vět nám musí sedět s pojmenováním v referenčním přepisu *words.mlf*. Zvukovou stopu pak nařežeme pomocí programu *WaveCutter*. Data do tohoto programu byla potřeba v této podobě:

Vstup=Hokej-muzi-OH2010_Ctvrtfinale_Finsko_Cesko_01.wav
0.000 5.008 hokej0000veta0001.wav
5.008 11.804 hokej0000veta0002.wav
11.804 13.259 hokej0000veta0003.wav
13.259 21.582 hokej0000veta0004.wav
21.582 33.72 hokej0000veta0005.wav
33.72 46.091 hokej0000veta0006.wav
46.091 51.815 hokej0000veta0007.wav
51.815 59.61 hokej0000veta0008.wav
59.61 66.192 hokej0000veta0009.wav

kde na prvním řádku je uveden název zvukové stopy, která musela být vždy ve stejné složce jako soubor se seznamy vět. Na dalších řádkách jsou pak vždy jednotlivé věty spolu s informacemi o jejich časovém začátku a konci. Jak je již uvedeno výše, značení jednotlivých vět muselo korespondovat s jejich označením v referenčním přepisu. Pro lepší ilustraci opět uvedeme příklad referenčního přepisu:

#!MLF!#
"/hokej0000veta0001.lab"*
tak
a
je
potřeba
se
pořádně

nabudit
protože
ten
velký
zápas
za
chvíli
přijde
.
"/hokej0000veta0002.lab"*

já
myslím
…

První věta označena tedy jako *hokej0000veta0001,* která začíná v čase 0 a končí v čase 5.008, obsahuje text uvedený výše. Stejným způsobem na sebe musely samozřejmě sedět veškeré věty s jejich zvukovou stopou. Pro jistotu jsem vždy před samotným trénováním pustil pár náhodných vět a zkontroloval, jestli jejich zvuk sedí na to, co je skutečně řečeno. Protože se stále jednalo pouze o ručně psaná data a ačkoliv jsem v průběhu času dělal skripty robustnější a robustnější, občas se stejně objevila nějaká nova chyba, která byla objevena až v průběhu trénování. Taková situace je samozřejmě nejhorší, neboť je pak potřeba veškerá data na začátku opravit a začít s celou přípravou od začátku. Pokud by se například objevila chyba v tomto kroku a neseděly by na sebe jednotlivé věty, zjistili bychom to až téměř u konce trénování HTK a to při přerovnání jednotlivých vět a vytvoření souboru s větami, které nám výrazně kazí natrénování parametrů modelu (soubor *ne.scp*). V takovémto případě by se v tomto souboru nevyskytlo pouze několik vět, ale soubor by obsahoval stovky a stovky vět a bylo by tedy podezření, že na sebe některé věty nesedí s jejich zvukovou stopou.

6.1.5. Vytvoření souborů train.scp, test.scp a param.scp

Poslední, co je ještě potřeba vytvořit, jsou soubory obsahující seznamy trénovacích a testovacích vět a seznam promluv pro parametrizaci. Formát těchto souborů je uveden návodu pro natrénování HTK a je potřeba se jej držet. Skript pro vytvoření těchto souborů byl vcelku jednoduchý a byla to jedna z mála částí, ve které se nikdy nevyskytl problém. Jediné, na co by zde možná bylo dobré upozornit je, aby soubory obsahující trénovací a testovací věty obsahovaly odlišné věty. Tedy abychom testovali na jiných větách, než na kterých natrénujeme akustický model. Uvedeme-li opět krátký příklad například již k výše uvedenému hokeji, pak trénovací a testovací soubor vypadá takto:

htk/hokej0000veta0001.htk
htk/hokej0000veta0002.htk
htk/hokej0000veta0003.htk
htk/hokej0000veta0004.htk
htk/hokej0000veta0005.htk

a soubor pro parametrizaci pak:

wav/hokej0000veta0001.wav htk/hokej0000veta0001.htk
wav/hokej0000veta0002.wav htk/hokej0000veta0002.htk
wav/hokej0000veta0003.wav htk/hokej0000veta0003.htk
wav/hokej0000veta0004.wav htk/hokej0000veta0004.htk
wav/hokej0000veta0005.wav htk/hokej0000veta0005.htk

z příkladů vidíme, že stejně jako u referenčního přepisu *words.mlf* i zde bylo důležité držet stejné pojmenování, aby promluvy zůstaly propojené mezi všemi soubory. Před každou větou vidíme, že je ještě uveden začátek *wav/* nebo *htk/*. To z toho důvodu, že skript, který později využívá tyto věty, je pouštěn ve stromové struktuře o úroveň výše a obsahuje podsložky *wav*, která obsahuje jednotlivé promluvy nařezané podle jednotlivých vět a ve složce *htk* jsou pak jednotlivé věty parametrizované.

Pro parametrizaci jsme tedy využili metody MFCC spolu s CMN (cepstral mean normalization). CMN slouží k odstranění efektu, který vznikl konvolucí signálů. Výsledkem parametrizace jsou pro jednotlivé mikrosegmenty vektory příznaků obsahující 3 x 13 parametrů (melovské kepstrální koeficienty, delta koeficienty, delta-delta koeficienty).

6.1.6. Automatizace trénování s HTK

Po přípravě výše uvedených souborů byla k dispozici veškerá data nutná pro trénování s nástrojem HTK. Vzhledem k tomu, že sportů, které byly potřeba natrénovat, bylo několik desítek a trénování akustických modelů s HTK vždy stejné, bylo vhodné se pokusit tento krok nějak zautomatizovat. Při mém začátku na této práci jsem nejprve myslel, že bych se mohl pokusit zautomatizovat celý výše uvedený proces. Tedy že na vstupu programu budou přepisy a slovník pro uvedený sport, já vypíšu nějaký inicializační soubor, kde zadám základní informace jako jméno sportu, počet testovacích vět a podobně a celý proces dále bude fungovat na zmáčknutí jednoho tlačítka a výsledkem dále bude natrénovaný akustický model. Ačkoliv by se možná našel jeden nebo dva sporty, kde by tyto myšlenky bylo možné aplikovat, velmi brzy jsem zjistil, že celý proces se mi nikdy zautomatizovat nepovede. Neboť v přepisech se vždy objevilo nějaké překvapení, které mi celý proces shodilo a hledat pak chybu nebylo vůbec jednoduché.

Nakonec jsem se tedy rozhodl zautomatizovat pouze část trénování akustického modelu pomocí nástroje HTK. Po několika natrénovaných sportech jsem zjistil, že když dojde k vytvoření přepisu na úrovni fonémů a následně provedení první reestimace, obvykle pak už nedojde k žádné další chybě. A od tohoto kroku jsem se tedy rozhodl celý proces zautomatizovat. Pro tyto účely jsem vytvořil jednoduchý powershellový script, kde jednotlivé příkazy vypadaly například následovně:

#CMD.EXE /C "HERest -T 1 -C CF.mfc -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm1/MODELS -M hmm2 monophones0.jpg"

#CMD.EXE /C "HERest -T 1 -C CF.mfc -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm2/MODELS -M hmm3 monophones0.jpg"

#CMD.EXE /C "HERest -T 1 -C CF.mfc -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm3/MODELS -M hmm4 monophones0.jpg"

#python model_pauz.py

….

Vidíme tedy, že po použití klíčového slova *#CMD.EXE* stačilo vkládat jednotlivé příkazy z HTK, které byly postupně vykonávány. V powershellovém scriptu se dá také snadno zavolat kód vytvořený v jiném programu, čehož jsem také využil, což můžeme vidět na příklad *#python model_pauz.py.* Kde jazyk python má pro mě nejjednodušší syntaxi a v něm jsem tedy vytvořil prográmek, který mi automaticky přidával model pauzy do seznamu modelů.

Tento skript tedy obsahoval první reestimace, následně přidání modelu pauzy, další reestimace, přerovnání dat, finální reestimaci, vytvoření rozpoznávání sítě a následně puštění samotného rozpoznávání na testovacích větách. Tím bylo tedy dosaženo finálního natrénování jednosložkového monofónového akustického modelu. V této fázi se skript zastavil a čekal na potvrzení, zda-li je vše správně. Pokud ano a tedy testovací věty byly rozpoznány na přibližně očekávané hodnoty a všechny ostatní soubory vypadaly, pustil jsem skript dále. Poslední část skriptu obsahovala přidávání jednotlivých složek do akustického modelu a následně opět čtyři reestimace parametrů. V této fázi jsem zatím modely trénoval pouze na osm složek, i když samozřejmě pro lepší výsledky by bylo vhodně dále pokračovat, avšak pro zatím toto nebylo potřeba. Dále jsem složky nepřidával samozřejmě především z časových důvodů, neboť jednotlivé reestimace se při přidání každé složky časově prodlužovali a u sportů jako hokej nebo fotbal už přidání pouze osmi složek trvalo více než den.

Díky tomuto skriptu jsem si tedy ušetřil spousty psaní do příkazové řádky a také spoustu času, neboť skript fungoval ve většině případů velmi spolehlivě a nebylo potřeba do celého procesu nějak zasahovat. Samozřejmě se zde vyskytly výjimky, jako například když při přidávání složek vyhodilo HTK chybu z důvodu příliš velkého množství dat. Ale jinak tedy došlo k poměrně velké časové úspoře.

7. Analýza vysílacího schéma ČT Sport

V předchozí části jsme si ukázali, jak jsou zpracovány jednotlivé sporty. K tomu bylo ale samozřejmě potřeba vybrat vhodné reprezentanty, na kterých bude přímé titulkování vyzkoušeno. Nejprve však bylo potřeba zanalyzovat vysílací schéma kanálu ČT Sport, na kterém bude přímé titulkování prováděno, abychom věděli, jak jsou jednotlivé sporty často vysílány a zkoušeli přímé titulkování především na častěji se vyskytujících se sportech.

Nejprve bylo potřeba v nějaké zpracovatelné podobě získat program ČT Sport. Aby byla analýza co nejpřesnější a nejaktuálnější rozhodl jsem se zanalyzovat celý rok 2017. Program byl pak získán v následující podobě:

2017.01.01;06:00;Hokej;Kanada - USA;Záznam utkání na MS v ledním hokeji hráčů do 20 let v Kanadě
2017.01.01;10:00;;Běžkotoulky;Vydejte se s námi za těmi nejkrásnějšími místy v České

republice. Tentokrát navštívíme Zadov. Tak neváhejte, nazujte běžky a pojeďte s námi
2017.01.01;10:10;Plavání;Otužilci na Vltavě;Reportáž z tradiční vánoční plavecké soutěže otužilců ve Vltavě
2017.01.01;10:30;;Hokejový mušketýr Jan Palouš;Patřil mezi české hokejové mušketýry, proslavil se jako střelec první branky hokejového mužstva Čech a získal dva tituly mistra Evropy

kde jako první byl vždy uveden datum vysílání, dále pak začátek pořadu, sport, který se vysílá a nakonec stručný popis. Jednotlivé údaje byly odděleny středníkem. Analýzu jsem se rozhodl řešit tak, že byl postupně procházen řádek po řádku, ze kterých byly získávány pouze relevantní informace k analýze. Bylo vždy potřeba zjistit, o jaký sport se jedná, a dále pak jeho začátek a konec. Podle začátku a konce daného sportu se pak jednoduše dala určit jeho doba vysílání. Tento údaj byl pak převeden na minuty a zapsán do pole k danému sportu. Na konci pak byly jednotlivé doby trvání u každého sportu nasčítány a převedeny na procenta. To bylo poměrně jednoduché, neboť byl analyzován celý rok 2017 a stačilo pouze sečíst údaje u jednotlivých sportů. Celkem se jednalo o 525444 minut vysílání (celý rok má 525600 minut, avšak první pořád nezačínal přesně o půlnoci). Jednotlivé údaje pak byly zapsány do následující tabulky.

Sport	minuty	procenta
Fotbal	93015	17,70217
Hokej	57636	10,96901
Zprávy	56032	10,66374
Ostatní	44501	8,469218
Basketball	22731	4,326056
Cyklistika	21366	4,066275
Volejbal	18804	3,578688
Motorismus	18120	3,448512
Biatlon	17807	3,388943
Atletika	15650	2,978433
Florbal	14155	2,693912
Tenis	12791	2,434322
Golf	10141	1,929987
Horská kola	9320	1,773738
Klasické lyžování	8475	1,612922
Alpské lyžování	8330	1,585326
Cyklokros	6822	1,298331
Jezdectví	6325	1,203744
Házená	5690	1,082894
Futsal	5600	1,065765
Boby	5545	1,055298
Poker	4865	0,925884
Krasobruslení	4825	0,918271

Zákulisí	4564	0,868599
Letecký sport	4551	0,866125
Akrobatické lyžování	4172	0,793995
Jachting	3795	0,722246
Diskuse	3315	0,630895
Tanec	3070	0,584268
Kanoistika	2895	0,550963
Plážový volejbal	2815	0,535737
Hokejball	2811	0,534976
Veslování	2305	0,438677
Skeleton	2214	0,421358
Curling	2060	0,392049
Vodní slalom	1975	0,375873
Cyklotoulky	1611	0,306598
Rychlobruslení	1604	0,305266
Americký fotbal	1340	0,255022
Pozemní hokej	1320	0,251216
Bojové sporty	1268	0,24132
Skoky na lyžích	1195	0,227427
Triatlon	1135	0,216008
Šachy	1050	0,199831
Snowboarding	1010	0,192218
Plavání	990	0,188412
Baseball	875	0,166526
Badminton	650	0,123705
Ragby	625	0,118947
Dostihy	563	0,107147
Gymnastika	300	0,057095
Softball	290	0,055191
Požární sport	220	0,041869
Sportovní aerobik	220	0,041869
Vzpírání	90	0,017128
Celkem	525444	

Tabulka 1 Analýza vysílacího schématu české televize

Dle očekávání se na prvních dvou místech objevil fotbal a hokej. Následovaly pak zprávy, kde kromě klasických zpráv anebo například „Branek, bodů, vteřin“ byly zahrnuty také sestřihy a souhrny sportovních událostí. Na čtvrtém místě v tabulce můžeme vidět kategorii Ostatní, do které byly zahrnuty sporty s velmi malou četností, jako například dřevorubectví (25 hodin) a dále pořady jako „Běžkotoulky“ nebo „Zákulisí“, kterých se na ČT Sport k mému překvapení vysílá poměrně dost. Velice často vysílány pak byly také míčové sporty jako basketball, volejbal nebo florbal. Hodně vysílacího času na ČT Sport zabírá také cyklistika (silniční a dráhová) a to i přes to, že do ní nebyl započten cyklokros a horská cyklistika.

8. Výběr vhodných reprezentantů pro přímé titulkování

Po provedení analýzy vysílacího schématu ČT Sport bylo potřeba vybrat vhodné reprezentanty s ohledem na kvalitu akustického signálu, pozadí a počtu řečníků. Tyto aspekty bylo důležité zkombinovat také s četností vysílání jednotlivých sportů, neboť by bylo hezké, kdyby nám přímé titulkování fungovalo na takovém softballu, avšak z hlediska využití ne příliš praktické. Z hlediska týmových sportů, které se ve vysílání objevují nejčastěji, jsem se rozhodl vybrat hokej, basketball a cyklistiky. U těchto sportů nebyly očekávány příliš dobré výsledky, neboť sporty obvykle namlouvá více řečníků, kteří si skáčou do řeči a často také při poměrně hlasitém šumu v pozadí (skandování diváků). Výjimku zde tvořila možná cyklistika, kde jsem očekával poměrně dobré výsledky, neboť každoročně sleduju pořady jako Tour de France, které vím, že jsou z velké části namlouvány jedním řečníkem, Tomášem Jílkem, a jsou namluveny velice kvalitně snad z výjimkou povzbuzování diváků a hluku aut a vrtulníků v pozadí.

Vzhledem k tomu, že účelem bylo obsáhnout celou škálu pořadů, byly dále vybrány zimní sporty a to biatlon, který je v České republice poslední léta velice populární a dále pak alpské lyžování vzhledem k úspěchům Ester Ledecké na posledních olympijských hrách. Dalšími velmi populárními sporty v Čechách jsou také atletika a tenis a proto nemohly chybět ani v tomto výběru. Vzhledem k tomu, že u atletiky jsem ve vysílání narazil na velmi časté sestřihy, rozhodl jsem se u tohoto sportu vybrat pro experimenty jak živě namluvený přenos tak také sestřih namluvený v klidnějším prostředí obvykle jedním člověkem.

Aby bylo dále zabrané celé spektrum sportů vysílaných na ČT Sport, rozhodl jsem se také vybrat motorismus, plavání a golf, které jsou také poměrně často vysílané, a byl jsem velice zvědav, jak dopadne rozpoznávání u těchto druhů sportů, neboť vybírat samé míčové sporty jako basketball, volejbal nebo házenou tak věřím, že výsledky by byly velice podobné. Nakonec bylo potřeba vybrat také sport, u kterého se dalo čekat dobrých výsledků. Z tohoto ohledu byl vybrán curling, neboť se jednalo o sport, který jsem si téměř celý přepsal sám a věděl jsem, že byl poměrně kvalitně namluven, téměř bez akustického šumu a byl to tedy jeden z mých favoritů na dosažení nejlepších výsledků.

9. Provedení rozpoznávacího experimentu

9.1. Optimální volba počtu stavů a složek

V předchozích krocích tedy byla provedena analýza vysílacího schématu České televize a z ní následně vybrány sporty pro přímé titulkování a to z hlediska četnosti vysílání, množství řečníků a předpokládané kvality zvukové stopy.

Dalším krokem bylo natrénovat akustické modely jednotlivých sportů na ideální počet stavů a složek. Jako první nástřel bylo zvoleno 24 složek. Během přidávání složek jsem u každého sportu prováděl také jednoduché rozpoznávání s rozpoznávací sítí na 100 větách, aby bylo vidět, zda stále roste úspěšnost rozpoznávání. Tato rozpoznávací síť představuje zerogramový jazykový model, který obsahuje pouze slova, která bude obsahovat rozpoznávána předloha. Všechny tyto slova pak mají stejnou pravděpodobnost. Při malém počtu testovacích vět by pak byla také velmi malá neurčitost tohoto jazykového modelu a je tedy dobré rozpoznávat na větším množství vět. Výsledky těchto testů najdeme v tabulce 2.

Podle těchto výsledků se pak přibližně dalo zjistit, jaký je pro daný sport optimální počet složek. Postupným rozpoznáváním jsem zjistil, že to především záleží na množství dat a počtu řečníků. Pokud sport namlouval pouze jeden řečník a bylo k dispozici menší množství dat, stačilo trénování zastavit někde na patnácté složce. Pro obsáhlé sporty jako byl fotbal nebo hokej pak úspěšnost neustále rostla i okolo pětadvacáté složky. Na následujícím obrázku můžeme vidět úspěšnost rozpoznávání pro biatlon s postupným přidáváním složek.

Obrázek 7 Úspěšnost rozpoznávání biatlonu

Z obrázku je krásně vidět, že ze začátku nám úspěšnost roste poměrně skokově. Ale okolo 15 složky se úspěšnost téměř zastavila a dále pak osciluje okolo 73 procent. Optimální počet složek byl tedy pro každý sport trochu jiný. Nejsnazším řešením by samozřejmě bylo natrénovat všechny sporty například na 30 složek. Avšak v takovém to případě by mohlo dojít, a také v mé práci došlo, u mnoho sportů k přetrénování a výsledky pak naopak vychází o dost hůř.

Obdobný problém jako s počtem složek byl i s počtem stavů. Na začátku trénování jsem se snažil, aby tento počet byl mezi 3000-4000. Přibližně na takový počet byly trénovány modely v již dříve provedených experimentech [4]. Bohužel ani tento parametr jsem nakonec nemohl zvolit univerzální. Neboť i zde jsem postupnými experimenty zjistil, že záleží především na množství trénovacích dat. Na sportech s nejvíce daty (okolo 100 hodin) jako byla atletika nebo hokej byl optimální počet okolo 4500. Na sportech se středním počtem dat (okolo 40 hodin) jako například biatlon, alpské lyžování byl optimální počet okolo 3000. A na sportech s nejméně daty (méně než 35 hodin), jako například curling, pak okolo 2500.

Sport	Objem trénovacích dat	Úspěšnost na zerogramu	Počet řečníků	Počet složek	Počet stavů

Alpské lyžování	42 h	78,1%	3 a více	23	3079
Atletika_souhrn	117 h	88,6 %	3 a více	23	3998
Atletika_živě	117 h	70,1 %	3 a více	23	3998
Basketball	82 h	76,3 %	3 a více	24	4939
Biatlon	48 h	88,2 %	2	22	3239
Curling	32 h	86 %	2	17	2525
Cyklistika	62 h	85%	3 a více	22	4545
Golf	45 h	84,78	2	19	3783
Hokej	145 h	65%	3 a více	24	4575
Motorismus	37 h	79%	3 a více	22	3431
Plavání	43 h	87,2%	1	21	3006
Tenis	38 h	92 %	2	17	3001

Tabulka 2 Natrénované doménové akustické modely

Po natrénování akustických modelů na optimální počet stavů a složek bylo na čase přejít k přímému rozpoznávání na televizním přenosu. Ke každému sportu byl tedy vybrán a přepsán pořad z České televize o délce 20 minut. Tento přepis byl pro jistotu dvakrát zkontrolován, neboť se jednalo o klíčovou informaci, ze které byla zjištěna úspěšnost rozpoznávání a tak bylo velice nežádoucí, aby obsahoval chyby. Po přepsání souboru bylo potřeba vytvořit referenční přepis pro rozpoznávání. Ten obsahoval vždy nejprve název souboru a dále pak vždy na každém řádku jedno slovo. Důležité také bylo „prohnat“ tento přepis přes program *remotejmzw*, který často spojí jména v multislova (Jaromír Jágr => Jaromír_Jágr) nebo nahradí některá slova za zkratky (plus => + , takzvaný => tzv.). Po vytvoření reference bylo dále pro zlepšení úspěšnosti nutno do každého třídního jazykového modelu dodat soupisky(třídy). Jednalo se hlavně o jména sportovců, se kterými se daný jazykový model zatím nesetkal. K těmto slovům bylo dále potřeba vytvořit správnou výslovnost. K tomu se znovu dal použít program *remotejmwz*, avšak vzhledem k tomu, že se ve většině případů jednalo o jména nebo cizí slova, bylo lepší tuto výslovnost pro každý sport vytvořit ručně. I přes malý počet těchto slov však vytvoření kvalitní fonetiky pro všechny sporty byla časově poměrně náročná záležitost. Pokud bychom tato slova však nepřidali, program by je pak nikdy nemohl rozpoznat a úspěšnost rozpoznávání by pak byla samozřejmě o pár procent horší.

Když byly vytvořeny soupisky pro všechny sporty, nic už nebránilo tomu pustit se do samotného rozpoznávání. Pro každý sport byl vybrán nejlepší akustický model spolu s jeho seznamem trifónů, dále pak referenční přepis, soupiska jmen a nakonec unikátní jazykový model. Z hlediska jazykového modelování byly použity třídní trigramové jazykové modely. Každý model byl specifický a obsahoval jiný počet slov:

Sport	Množství slov v jazykovém modelu
Alpské lyžování	553272
Atletika_souhrn	527736
Atletika_živě	527736
Basketball	524899
Biatlon	553125
Curling	553292
Cyklistika	545256
Golf	524565
Hokej	553662
Motorismus	524733
Plavání	525185
Tenis	524590

Tabulka 3 Obsah jednotlivých jazykových model

Pro rozpoznávání bylo nutné ještě nastavit dva parametry a to penaltu vložení slova a váhu jazykového modelu. Nejsnazším řešením bylo vytvořit skript, do kterého byl vložen příkaz pro rozpoznávání s různým nastavením těchto parametrů a pro finální výsledek pak vybrat nejlepší z nich. I tyto parametry bylo potřeba pro každý sport nastavit jinak. Rozpoznávání pak probíhalo na procesoru počítače a běželo tak rychle, kolik jader a jak výkonných bylo k dispozici. Můj testovací počítač obsahoval přetaktovaný 8 jádrový Ryzen 1700, který disponuje velmi solidním výpočetním výkonem a tak rozpoznávání obvykle trvalo pouze okolo 40 % reálného času.

Po rozpoznání všech sportů na reálném televizním vysílání tak byly k dispozici konečně první opravdové výsledky. U některých sportů se výsledky pohybovaly podle očekávání, avšak u několika sportů byly výsledky horší, než jsme původně předpokládali. Ne příliš dobré výsledky byly především u sportů, kde jsme čekali nejlepší úspěšnost rozpoznávání a to například u tenisu nebo u curlingu. Po analýze těchto sportů bylo zjištěno, že obsahují často dlouhé pauzy, ve kterých se nemluví a bylo by tedy dobré využít detektor ticha (silence detector). Ten slouží k tomu, aby se nerozpoznávalo v případech, kdy se nemluví. Jinými slovy říká rozpoznávači kdy má a kdy nemá rozpoznávat. Dalším způsobem jak zlepšit úspěšnost rozpoznávání bylo zvolit vyšší prořezávací práh při přerovnávání trénovacích vět. Díky tomu byly z rozpoznávání vyhozeny téměř všechny věty, které mohly obsahovat například prohozená slova nebo větší ruch v pozadí a mohly tedy zhoršovat úspěšnost rozpoznávání. U všech sportů tak tento krok úspěšnost zlepšil. Například u plavání, kde se nejprve úspěšnost pohybovala někde okolo 82 %, bylo nyní dosaženo 87 %. Podobně tomu bylo i u tenisu, kde pomohla především aplikace detektoru ticha, neboť tento sport často obsahoval nejvíce ze všech dlouhé úseky, kde se vůbec nemluvilo.

9.2. Analýza rozpoznaného textu

V současné chvíli jsme tedy měli rozpoznaný text spolu s referencí a bylo potřeba je porovnat. Vzhledem k tomu, že se jedná o rozpoznávání řeči, které slouží k titulkování pořadů, objeví se zde plno chyb, které sice program rozpozná jako chybu, avšak z hlediska srozumitelnosti jsou naprosto irelevantní jako například:

koneckonců---------- konec konců
Křišťálový_Glóbus----------křišťálový glóbus
sezóně----------sezoně
jedenadvacet----------jeden a dvacet
neúčastnil----------nezúčastnil
Garcia----------García

Z uvedených příkladů je jasné, že z hlediska srozumitelnosti je velký rozdíl jestli rozpoznávač rozpozná špatně naprosto jiné slovo anebo dojde k rozpoznání výrazů uvedených výše. Často také referenční přepis obsahoval spojky jako *a, i,* které na první poslech neslyší často ani člověk, avšak z hlediska srozumitelnosti textu nejsou úplně důležité a tak i ty mohli být z kategorie chyb vypuštěny. Veškeré rozpoznané texty bylo tedy potřeba porovnat s referencí, projít slovo po slovu a případně opravit tyto méně vážné chyby. V následující tabulce pak můžeme vidět maximální dosaženou úspěšnost u jednotlivých sportů a to před i po úpravě rozpoznaného textu.

Sport	Rozpoznaný text	Upravený text
Alpské lyžování	79,53%	84,74%
Atletika_souhrn	89,08%	91,71%
Atletika_živě	76,06%	79,20%
Basketball	74,92%	78,70%
Biatlon	73,32%	78,35%
Curling	77,37%	80,86%
Cyklistika	73,52%	77,52%
Golf	81,36%	86,16%
Hokej	69,72%	71,24%
Motorismus	70,54%	72,51%
Plavání	88,08%	91,91%
Tenis	91,36%	94,95%

Tabulka 4 Výsledky rozpoznávání všech sportů

Z výše uvedené tabulky vidíme, že původní úspěšnost rozpoznávání se pohybovala od 69 do 92 %. Po analýze jednotlivých chyb pak došlo ke zlepšení v rozmezí 2-5 %. Zajímavá je také statistika počtu slov, které byly rozpoznány, kterou můžeme vidět v tabulce 5. Dále se na každý sport podíváme o něco podrobněji a řekneme si, proč dané výsledky vyšly tak, jak vyšly a zdali je vhodné je použít pro přímé titulkování.

Sport	Počet rozpoznaných slov	Počet unikátních slov
Alpské lyžování	2218	1058
Atletika_souhrn	1851	902
Atletika_živě	1778	768
Basketball	1618	765
Biatlon	2004	831
Curling	2226	851
Cyklistika	2525	1120
Golf	1707	853
Hokej	1904	893
Motorismus	2760	1074
Plavání	1163	608
Tenis	725	438

Tabulka 5 Statistika rozpoznaného textu

Tabulka 5 nám ukazuje množství slov v jednotlivých sportech, vidíme, že zdaleka nejméně slov obsahoval tenis následovaný plaváním.

9.2.1. Alpské lyžování, biatlon

Oba tyto sporty patří mezi nejčastěji vysílané zimní sporty a bylo tedy nezbytné zařadit je do experimentu. Při analýze těchto sportů jsem očekával, že biatlon vyjde o něco lépe, neboť všechny jeho přenosy mluví dva poměrně stálý komentátoři, kteří jsou občas doplněni rozhovorem s šéftrenérem biatlonistů nebo některým ze závodníků. Alpské lyžování pak namlouvají většinou také dva mluvčí s tím, že jeden je stálý a druhý se střídá přenos od přenosu. Podíváme-li se na výsledky tak nakonec biatlon dosáhl 78,4 % a alpské lyžování 84,7 %. Při analýze výsledků jsem zjistil, že alpské lyžování je rozpoznáváno přibližně pořád stejně, zatím co biatlon má úseky, kde rozpoznávání funguje prakticky bez chyb a naopak úseky, kde je téměř vše rozpoznáno špatně. To se děje především při střelbě biatlonistů, kdy je z ruchových mikrofonů do zvukové stopy přimíchán veliký ruch diváků a i komentátoři zde velice často mluví, nebo i křičí přes sebe. U alpského lyžování tyto problémy nejsou, pouze výjimečně je o něco větší ruch v pozadí například při startu domácího závodníka nebo překonání vítězného času a z toho důvodu je zde tedy dosaženo lepších výsledků.

9.2.2. Hokej, basketball

Další velkou kategorií byly kolektivní sporty. Zde se již od začátku daly očekávat jedny z nejhorších výsledků rozpoznávání. To především proto, že každý přenos je obvykle namlouván někým jiným, komentátoři často přenosy velmi prožívají, dochází zde k přeřekům a v neposlední řadě tyto sporty obsahují velké množství šumu v pozadí. Výsledky těchto sportů dopadly podle očekávání a úspěšnost se zde pohybovala okolo 75 %. Bohužel v tomto případě jediný způsob, jak by bylo možné vylepšit úspěšnost rozpoznávání, by dle mého názoru bylo získat kvalitnější zvukovou stopu od České televize. To znamená získat zvuk, který neobsahuje šum na pozadí. Bohužel ani toto řešení by nebylo možné vždy, protože velmi často jsou tyto sporty moderovány přímo na stadionu a v takovém případě není příliš možné tento šum odstranit.

9.2.3. Motorismus

Motorismus byl pro tento experiment vybrán především z důvodu jeho vysoké četnosti ve vysílání. Také zde byly očekávány velice špatné výsledky. Při poslechnutí několika testovacích vět jsem dokonce očekával zdaleka nejhorší úspěšnost rozpoznávání, neboť data obsahovala nejhorší akustický šum ze všech sportů. Akustické pozadí velmi často obsahovalo hlasité zvuky motorů a dokonce i pro člověka byla často řeč moderátora ne příliš srozumitelná. Navíc zde byly smíchány opět hlasy několika řečníků dohromady a nebyl zde tedy příliš důvod doufat v lepší výsledky. Veškeré tyto předpoklady se také potvrdily a úspěšnost rozpoznávání na testovacím souboru zde byla zdaleka nejnižší (okolo 70 %). Stejně jako kolektivní sporty ani motorismus tedy není příliš vhodný sport pro přímé titulkování.

9.2.4. Atletika

Sportem, který také nemohl chybět v našem experimentu, byla královna sportů atletika. Zde jsem si netroufl příliš odhadovat výsledky. Trénovací sada se totiž skládala ze dvou podobně zastoupených typů dat. První z nich byly klasické atletické přenosy jako mistrovství světa, diamantové ligy apod. Druhou skupinu pak tvořily sestřihy, které byly namluveny v klidu, ve studiu a pouze občas obsahovaly rozhovor s některým ze závodníků. Zde jsem se tedy rozhodl vybrat na rozpoznání oba dva tyto druhy přenosů. Výsledky rozpoznávání zde dopadly podle očekávání. Studiový přenos se pohyboval okolo velice přijatelných 92 %, což bylo o přibližně 13 procent více než výsledky u přenosu přímo ze závodu. U studiového přenosu pak byl krásně vidět rozdíl mezi tím, kdy mluvil profesionální komentátor a rozpoznávání pak probíhalo velice solidně a naopak, kdy se ve zvukové stopě objevil rozhovor se sportovcem a rozpoznáno nebylo téměř nic. I přes tyto obtíže je však atletika sportem, kde by mohlo dojít k nasazení přímého titulkování a to alespoň u těchto souhrnů namluvených z klidného prostředí.

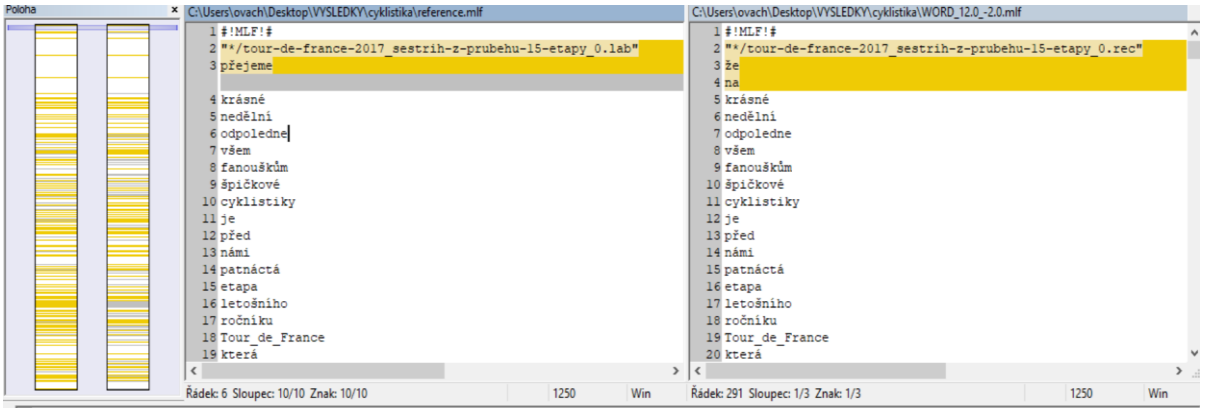
9.2.5. Golf

Dalším sportem, u kterého jsme očekávali velice dobrých výsledků, byl golf. Tento sport jsem z velké části přepisoval a věděl jsem, že byl namluven pouze dvěma mluvčími a to velice kvalitně. Jediné obavy jsem zde měl, že moderátor Jaromír Bosák se v některých fázích přenosu velice vciťoval do dané situace a při rozhodujících úderech mluvil velice ale velice potichu. Podíváme-li se na výsledky rozpoznávání tak úspěšnost zde byla dosažena 86 %. To pro mě bylo mírným zklamáním, avšak při kontrole reference a rozpoznaného textu jsem zde narazil na velice zajímavý jev. Byly zde úseky, které fungovali téměř na 100 %, následovány naopak částmi, kde se úspěšnost pohybovala velice nízko. Při poslechnutí zvukové stopy jsem zjistil, že se jedná o úseky, které obsahují buď to velmi hlasitý potlesk diváků, při kterém nebylo správně rozpoznáno téměř nic anebo pak úseky, kde v pozadí hraje hudba a úspěšnost zde pak byla okolo 50 až 60 %. Pokud by se tedy u golfu podařilo získat zvukovou stopu, která obsahuje pouze namluvený přenos bez těchto přidaných efektů, jednalo by se o sport, který je jeden z nejvhodnějších kandidátů pro přímé titulkování.

9.2.6. Cyklistika

Pokud byl v diplomové práci sport, u kterého jsem si ani při nejmenším netroufl odhadovat výsledky, byla to právě cyklistika. Jakožto velký fanoušek každoročně sleduji etapový závod Tour de France. Věděl jsem tedy, že sport pravidelně každý rok namlouvá Tomáš Jílek, který mluví velmi zřetelně a plynule, avšak vzhledem k tomu, že se jedná o velice dlouhé přenosy,

obvykle 5-6 hodin, společnost mu vždy dělají minimálně dva další hosté. Obvykle se jedná o bývalé závodníky, kteří nemluví úplně nejlépe. Takže jsem se velice těšil, jak dané výsledky rozpoznávání dopadnou. Podíváme-li se na výsledky, vidíme, že celková úspěšnost se blížila k hranici 80 %. To samo o sobě není příliš zajímavé, avšak při procházení rozdílů mezi referencí a rozpoznaným textem se zde ukázal být enormní rozdíl mezi profesionálním komentátorem a hosty.



Obrázek 8 Výstup z programu WinMerge pro porovnání textových souborů

Na obrázku můžeme vidět porovnání obou souborů pomocí programu WinMerge. V levé části pak vidíme dva sloupečky, kde bílá část značí shody v textech a žlutá chyby. Vidíme, že ze začátku souboru je přepis téměř bezchybný. Podobné úseky téměř bez chyb jsou pak ještě uprostřed a na konci, avšak už menšího rozsahu. Při poslechnutí jsem zjistil, že tyto velice dobré úseky jsou všechny namluveny Tomášem Jílkem. Jakmile pak dojde ke změně mluvčího, úspěšnost okamžitě padá rapidně dolů. Bohužel toto je problém, který asi nebude možné odstranit, neboť do takto dlouhých přenosu budou vždy pozváni nějací další hosté. Samozřejmě pokud bychom vybrali úsek, který je z velké části namlouván pouze profesionálem, úspěšnost rozpoznávání by se pohybovala okolo 90 %, avšak také bychom mohli narazit na úseky, kde by úspěšnost rapidně poklesla a titulky by tak byly zcela nepoužitelné. Z tohoto důvodu tedy cyklistiku nemohu doporučit pro přímé titulkování.

9.2.7. Tenis, plavání

Další kategorií sportů pak byl tenis a plavání. Ty reprezentovaly sporty namlouvané pouze jedním řečníkem. Při analýze těchto sportů jsem měl pouze mírné obavy, že ačkoliv je namlouvá jeden řečník, tak obsahují občas výraznější akustický šum a mohlo by tedy dojít k pokažení výsledků. Na oba tyto sporty byl nakonec nasazen detektor ticha, neboť zde často byly úseky, kde komentátoři téměř nemluvili. Podíváme-li se na úspěšnost rozpoznávání tak u obou těchto sportů bylo dosaženo vůbec nejlepších výsledků. Po úpravách se tenis dostal téměř k 95 % a plavání pak k 92 %. Těmito výsledky jsem byl mile překvapen a můžeme tedy říci, že oba tyto sporty a pravděpodobně tedy i ostatní sporty namlouvané jedním řečníkem, jsou velice vhodné pro přímé titulkování.

9.2.8. Curling

Největším zklamáním celého experimentu byl pro mě curling. Jednalo se o sport, který jsem z 90 % přepsal celý sám. Ačkoliv ho namlouvali dva řečníci, věřil jsem, že zde dosáhneme velice slušných výsledků, neboť se jedná o klidnější sport, který neobsahuje téměř fandění diváků.

Bylo zde k dispozici i poměrně slušné množství dat a tak jsem mohl přistoupit k většímu prahu při přerovnání vět. I výsledky při mém zběžném testování vypadaly velice dobře a pohybovaly se okolo 90 %. Bohužel na reálném televizním vysílání, ať jsem se snažil sebevíc, nejlepších výsledků, kterých jsem dosáhl, bylo necelých 81 %. Takto špatného výsledku bylo dosaženo ze dvou důvod. Zaprvé od roku 2010, ze kterého pocházela většina testovacích dat, bohužel došlo k výměně jednoho z komentátorů. Na něj tedy nebyl akustický model vůbec připraven. Druhým problémem se ukázaly být samotní hráči. Při odehrávání jednotlivých kamenů velice hlasitě křičí na metače, co mají dělat a ať už se jedná o instrukce v angličtině, švédštině nebo v češtině, nutí to bohužel systém rozpoznávat slova navíc. Pokud by tedy byla získána čistá zvuková stopa a přidáno do trénovací sady pár přepisů od nového komentátora, pak i zde by bylo možné doporučit přímé titulkování.

9.3. Finální zhodnocení

Z výše uvedených analýz lze tedy dojít k poměrně jasným výsledkům. Přímé titulkování televizních přenosů je určitě možné. Avšak ne u všech sportů. Jeho aplikace je za dosavadních podmínek možná u dvou typů sportů. Prvním z nich jsou sporty, které jsou namlouvány pouze jedním řečníkem za přítomnosti ne příliš vysokého akustického šumu. Z výše uvedených sportů můžeme do této kategorie zařadit tenis (95 %), plavání (91 %) a studiovou atletiku (92 %). Druhou kategorií, kde je možné uplatnit přímé titulkování, jsou sporty, které namlouvají z velké části dva stálí řečníci, avšak za přítomnosti malého množství šumu. Sem můžeme zařadit golf či alpské lyžování. Poslední kategorií, kde by bylo přímé titulkování možné v budoucnu, jsou sporty jako curling nebo biatlon. Tedy sporty, kde v okamžicích klidnějšího pozadí dosahujeme velice slušných výsledků, avšak dojde-li ke zvýšení akustického šumu, výsledky rozpoznávání velice upadají. Tyto sporty by tedy pro přímé titulkování potřebovaly od České televize čistou zvukovou stopu bez zvuku získaného ruchovými mikrofony.

Naopak kategorií, která pro přímé titulkování vhodná nikdy nebude, jsou kolektivní sporty. Zde se každý zápas střídají různí komentátoři, zápasy jsou často moderované přímo ze stadionu, takže ani možnost kvalitnějšího zvuku zde do budoucna není příliš možná. Na následujícím grafu pak můžeme vidět sporty, které se na základě této diplomové práce dají doporučit k přímému titulkování.

Obrázek 9 Sporty s nejlepšími výsledky rozpoznávání z přímé zvukové stopy

10. Závěr

Hlavním cílem této diplomové práce tedy bylo vybrat vhodné sporty pro přímé titulkování. Teoretická část práce se nejprve zabývá začátky rozpoznávání řeči a hlavními problémy, se kterými se při rozpoznávání řeči musíme potýkat. Dále je práce zaměřena na statistické metody pro rozpoznávání řeči. Problém je zde rozdělen do čtyř hlavních kroků, do kterých lze statistické rozpoznávání shrnout. Největší pozornost je pak věnována akustickému modelování, které je založeno na skrytých Markovových modelech.

Druhá část teorie je pak zaměřena na trénování akustických modelů pomocí nástroje HTK. Jsou zde ukázány a vysvětleny jednotlivé kroky doplněné o vlastní zkušenosti s jednotlivými problémy, se kterými se můžeme při trénování setkat. Je zde ukázán jak návod pro jednoduchý monofónový model tak i následný přechod na trifónový model.

Poté se práce dostává k samotné tvorbě trénovacích dat. Je zde ukázán program Transcriber, pomocí kterého byla všechna data vytvořena a v pár jednoduchých bodech je ukázán i správný návod, jak přepisy tvořit. Je zde ukázána jak možnost jednoduchého přepisu, tak také možnost značení jednotlivých tříd sloužících dále pro jazykové modelování. Nakonec je zmíněn také program LMEdit, pomocí kterého probíhá hromadná kontrola všech přepisů.

Praktická část práce pak začíná časově zdaleka nejnáročnějším úkolem a to zpracováním trénovacích dat pro veškeré dostupné sporty. Nejprve je provedeno vyčištění přepisů od nepotřebných informací a dále jsou vykonány opravy pomocí dříve vytvořených slovníků. Z těchto modifikovaných přepisů jsou pak vytvořeny veškeré potřebné soubory pro trénování akustických modelů s HTK.

Dále je provedena akustická analýza vysílacího schématu České televize za rok 2017. Pro každý sport je vypočtena jeho četnost vysílání na kanálu ČT Sport. Poté je proveden výběr vhodných reprezentantů pro přímé titulkování a to jednak na základě četnosti vysílání ale také z hlediska zkušeností získaných během let přepisování. Jsou vybrány sporty obsahující jednoho, dva nebo více řečníků a také sporty s klidnějším akustickým pozadím a naopak s výraznějším šumem, aby bylo obsaženo celé spektrum možností pro titulkování.

Nakonec je proveden rozpoznávací experiment. Nejprve jsou vykonány testy zjišťující nejvhodnější počet složek a stavů pro jednotlivé sporty. Poté jsou porovnány jednotlivé reference s jazykovými modely a na základě toho jsou vytvořeny seznamy OOV slov. Dále jsou spuštěny jednotlivé rozpoznávací experimenty s různými parametry pro zjištění nejlepší možné konfigurace. A nakonec je provedena analýza všech výsledků, kde jsou porovnána jednotlivá očekávání s reálnými výsledky a u některých sportů jsou uvedena doporučení pro přímé titulkování.

Výsledkem této práce je, že pro přímé titulkování jsou vhodné sporty namluvené buď to jedním řečníkem anebo pak sporty obsahující velmi kvalitní zvukovou stopu. Pro provedení přímého titulkování na více sportech by pak bylo zapotřebí získat lepší akustiku od České televize bez aditivního šumu.

11. Seznam literatury

1. Psutka, J., Müller, L., Matoušek, J. a Radová, V. : Mluvíme s počítačem česky, Academia, Praha, 2006.
2. People.cs.umass.edu. 2004 [9.3.2018]. Dostupné na World Wide Web: https://people.cs.umass.edu/~mccallum/courses/inlp2004a/lect10-hmm2.pdf.
3. Pražák, A., Ircing, P., Müller, L. : Language Model Adapation Using Different Class-Based Models, ZČU Plzeň.
4. Psutka, V., J., Pražák, A., Psutka, J., Radová, V. : Captioning of Live TV Commentaries from the Olympic Games in Sochi, ZČU Plzeň.
5. Psutka, J.: Komunikace s počítačem mluvenou řečí. Academia, Praha, 1995.
6. Ucnk.ff.cuni [online]. 2009 [cit 27.2.2012]. Dostupné na World Wide Web: <http://ucnk.ff.cuni.cz/oral/>.
7. Young, S. et al., The HTK book (for HTK version 3.4). Cambridge University Press, 2006.
8. Python.org [online]. 2008 [cit 3.12.2008]. Dostupné na World Wide Web: https://www.python.org/download/releases/3.0/.
9. Cs.wikipedia.com [online]. 2006 [31.5.2016]. Dostupné na World Wide Web: < https://cs.wikipedia.org/wiki/Extensible_Markup_Language>.
10. Brousseau, J., Beaumont, J., Bouliannne, G., Cardinal, P., Chapdelaine, C., Comeau, M., Osterrath, F., Oullet, P. : Automated close-captioning of live TV broadost news in French, CRIM Montreal.
11. Imaki, T., Kobayashi, A., Sato, S., Homma, S., Onoe, Kobayakawa, T. : Speech recognition for Subtlitling Japenese Live Broadcasts.
12. Váchal, O. : Automatický postup zpracování ručně anotovaných dat pro tvorbu akustických modelů, Bakalářská práce (Bc.), ZČU Plzeň, Katedra kybernetiky, 2016.
13. Kky.zcu.cz [online]. 2011 [7.1. 2011]. Dostupné na World Wide Web: http://www.kky.zcu.cz/cs/sw/jmzw.
14. Švec, J., Hoidekr, J., Soutner, D., Vavruška, J. : Web Text Data Mining for Building Large Scale Language Modelling Corpus, ZČU Plzeň.

Přílohy:

Příloha 1 – přepis vytvořený pomocí Transcriberu

<?xml version="1.0" encoding="CP1250"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="Denisa Müllerová" audio_filename="rallye-dakar-2018_dakarske-ozveny.wav" version="5" version_date="180505">
<Episode program="" air_date="">
<Section type="nontrans" startTime="0" endTime="18.84">
<Turn startTime="0" endTime="18.84">
<Sync time="0"/>
</Turn>
</Section>
<Section type="report" startTime="18.84" endTime="1515.989">
<Turn startTime="18.84" endTime="1515.989">
<Sync time="18.84"/>
krásný dobrý den vám přejeme při sledování letošních třetí Dakarských ozvěn
<Sync time="30.006"/>|
dnes se budeme věnovat kategorii kamionů která je z historického hlediska pro Českou republiku nejúspěšnější kategorií na Dakarské Rallye
<Sync time="39.06"/>
no a tím historicky nejúspěšnějším českým pilotem je šestinásobný vítěz Dakarské Rallye v kategorii kamionů Karel Loprais
<Sync time="46.563"/>
Karle přeji krásný den
<Sync time="48.328"/>
dobrý den přeji
<Sync time="49.838"/>
letošní čtyřicátá Dakarská Rallye nám teprve začala máme za sebou dvě etapy
<Sync time="55.575"/>
kamiony teď vyráží na start etapy třetí
<Sync time="59.538"/>
co zatím napovídá ta kategorie kamionů o nadějích českých jezdců líbí se vám to
<Sync time="65.69"/>
tak kategorie kamionů se dostala hned po startu do písku
<Sync time="69.895"/>
což je vynikající věc že celej ten písek začíná hned ze začátku a že se na něm všichni dostanou a užiou si ho pěkně
<Sync time="79.324"/>
ten písek si budou užívat i naši diváci dostatek písku si užívá i náš reportér Jan Rouec

<Sync time="85.358"/>
který je přímo na místě takže dejme mu slovo
<Sync time="88.531"/>
Dakarská karavana se stěhuje z města Pisko a vyráží do své třetí etapy

Příloha 2 – ukázka souboru obsahující OOV slova

viděném pt=viDenEm
kolorovaly pt=kolorovali
Si_Woo_Kima pt=sivUkima
Finau pt=finY
Graceovi pt=grejsovi
Dukeovi pt=dukovi
Cauleyho pt=kYliho
Brian_Gays pt=brajengajs
Kenu_Dukeovi pt=kenudukeovi
Greg_Norman pt=greknormen
Heritage pt=heritedZ
El_Clásica pt=elklasika
even pt=even
putt pt=pat
Grega_Normana pt=greganormena
Hilton_Head pt=hiltnhet
Cauley pt=cYli
puttem pt=patem
MacKenziem pt=mekenzijem
Jimmyho_Walkera pt=dZimihovolkera
Ianem_Poulterem pt=Ajenempyltrem
John_Huh pt=dZonhux
puttu pt=patu
Ryanu_Mooreovi pt=rajenumUrovi
Alamu pt=alamu
Valero pt=valero
Jonattan_Vegas pt=dZonatanvegas
Gracea pt=grejsa
Alamo pt=alamo
putty pt=pati
Bud_Cauley pt=batkYli
Reedovi pt=rIdovi
Keegan_Bradley pt=kIgnbredli
Brooks_Koepka pt=brUkskopka
Duke pt=djUk
Ken_Duke pt=kendjUk
Travelers pt=trevelrs

puttoval pt=patoval
Si_Woo_Kim pt=sivUkim

Příloha 3 – referenční přepis words.mlf

#!MLF!#
"/hokej0000veta0001.lab"*
tak
a
je
potřeba
se
pořádně
nabudit
protože
ten
velký
zápas
za
malou
chvíli
přijde
.
"/hokej0000veta0002.lab"*
já
myslím
že
nabuzení
ale
potřebují
i
čeští
reprezentanti
protože
ten
výkon
proti
Lotyšsku
žádná
sláva
.
"/hokej0000veta0003.lab"*
vlastně
postup
.