VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

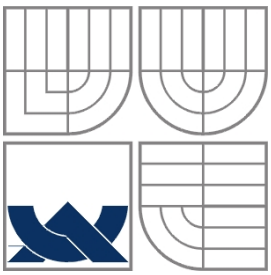
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

HLASOVÉ OVLÁDÁNÍ

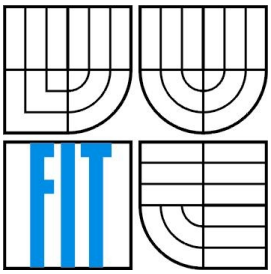
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE PETR SÁDECKÝ
AUTHOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

HLASOVÉ OVLÁDÁNÍ

VOICE CONTROL

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

PETR SÁDECKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. JAN ČERNOCKÝ

BRNO 2010

Abstrakt

V této bakalářské práci jsou studovány techniky pro rozpoznávání řeči a detekci klíčových slov s pomocí skrytých Markovových modelů. V první části jsou popsány techniky parametrizace signálu, trénování modelů a algoritmy pro rozpoznávání řeči. V druhé části je provedena detekce klíčových slov s několika modely natrénovanými z řečových signálů zachycených v různých vzdálenostech a výsledky těchto testů jsou následně vyhodnoceny.

Klíčová slova

skryté Markovovy modely, rozpoznávání řeči, detekce klíčových slov, The Hidden Markov Model Toolkit (HTK), trénování skrytých Markovových modelů

Abstract

This Bachelor's thesis deals with the techniques of the speech recognition and key words spotting using hidden Markov models. The first part of my thesis contains a description of the techniques of parameterization of the signals, models training and algorithms of speech recognition. In the second part, the detection of key words with several models trained on speech signals recorded in different distances is investigated and the results of the tests are evaluated.

Keywords

Hidden Markov Model (HMM), speech recognition, Keyword spotting (KWS), The Hidden Markov Model Toolkit (HTK), HMM training

Citace

Petr Sádecký: Hlasové ovládání, bakalářská práce, Brno, FIT VUT v Brně, 2010

Hlasové ovládání

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Jana Černockého. Další informace mi poskytl Ing. Petr Schwarz, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Petr Sádecký
31. července 2010

Obsah

1Úvod....3
1.1 Nastínění problému....3
1.2 Struktura práce....3
2 Zpracování řečového signálu....5
2.1 Vznik řečového signálu....5
2.2 Parametrizace řečového signálu....5
2.2.1 Mel-frekvenční kepstrální koeficienty – MFCC....7
3 Skryté Markovovy modely....8
3.1 Úvod do skrytých Markovových modelů....8
3.2 Baum-Welchova re-estimace....10
3.3 Viterbiho algoritmus....13
3.4 Rozpoznávání spojité promluvy....14
4 Nastínění detekce klíčových slov....17
4.1 Metoda skrytých Markovových modelů....17
5 Detekce klíčových slov modely fonémů....20
5.1 Diskuze problému....20
5.2 Použité nástroje a zdroje dat....20
5.2.1 HTK....20
5.2.2 Databáze SPEECON....21
5.2.3 Další nástroje....21
5.3 Způsoby vyhodnocení....21
5.4 Trénovací sety....22
5.5 Testovací sety....23
5.6 Klíčová slova....23
5.7 Použité sady modelů....24
5.8 Parametrizace a trénování modelů....26
5.8.1 Příprava....26
5.8.2 Parametrizace řečových nahrávek....27
5.8.3 Příprava modelů....28
5.8.4 Trénování modelů....28
5.9 Vlastní detekce klíčových slov....29
5.9.1 Příprava....29
5.9.2 Detekce....31

5.9.3 Získávání výsledků....32
5.10 Výsledky jednotlivých sad modelů....32
5.10.1 Testovací sada head_test....32
5.10.2 Testovací sada close_test....33
5.10.3 Testovací sada far_test....33
5.10.4 Test far_set na sadě far_test....34
5.10.5 Zhodnocení výsledků....34
6 Závěr....36

1 Úvod

Jedním z nejčastěji používaných prostředků pro komunikaci mezi lidmi je řeč. Jedná se o nejpřirozenější způsob dorozumívání. Mozek je evolučně přizpůsoben k tomu, abychom dokázali tuto cestu komunikace rozvíjet už od narození. Sdělení pomocí mluveného slova je pro nás velmi snadné a rychlé. Mnohem rychlejší než třeba napsání nebo nakreslení stejné informace. Snaha o vytvoření rozhraní, pomocí kterého by bylo možné komunikovat se strojem formou dialogu, je logická a lidstvo investuje velké úsilí k realizaci tohoto rozhraní.

1.1 Nastínění problému

Naše mluvené slovo je naneštěstí zatíženo velkým množstvím informací, které jsou pro identifikaci konkrétních slov zbytečné. Je to například naše nálada v danou chvíli, náš věk a pohlaví nebo aktuální zdravotní stav. Díky tomu se stává rozpoznávání řeči obtížnější. Vezmeme-li dále v potaz, že by bylo záhodno rozpoznávat řeč bez určení konkrétního mluvčího, obtížnost našeho úkolu se mnohonásobně zvýší. Pokud navíc chceme slova rozpoznávat v běžném prostředí, potýkáme se ještě s kontaminací řečového signálu velkým množstvím šumů z okolního prostředí. Pro některé tyto problémy rozpoznávání řeči bylo již řešení nalezeno. Jiné se stále řeší. V dnešní době se pro rozpoznávání řeči používají nejvíce systémy založené na skrytých Markovových modelech. Je nutno podotknout, že zatím žádná metoda pro rozpoznávání řeči se neukázala jako nejlepší a správná.

Zde se ovšem nebudeme zabývat rozpoznáváním celé promluvy, ale zaměříme se na detekci klíčových slov, kde se snažíme v řeči nalézt určitá slova, která jsou pro nás důležitá. Příkladem by mohlo být ovládání spotřebičů domácnosti, kdy se systém snaží detekovat povely. Nebo se může jednat o detekování jistých informací v běžném hovoru. Případně by mohlo být detekování klíčových slov použito pro automatický informační portál po telefonu, kdy systém pomocí klíčových slov rozhodne, jakou informaci hledáme, místo toho, abychom používali tlačítkovou volbu.

1.2 Struktura práce

V následující kapitole se budeme zabývat úpravou řečového signálu a následným převodem na parametry. Řekneme si něco o Mel-frekvenčních kepstrálních koeficientech a krátkodobé energii. Ve třetí kapitole si povíme základní informace ke skrytým Markovovým modelům. Ukážeme si, jakým způsobem je lze použít při rozpoznávání řeči. Nastíníme Baum-Welchovu re-estimaci a její použití pro trénování modelů. Podíváme se na Viterbiho algoritmus, který lze využít

pro rozpoznávání. Nakonec si povíme něco o rozpoznávání spojité řeči. Čtvrtá kapitola se bude věnovat detekci klíčových slov. Nastíníme si metody využívané k tomuto účelu a zaměříme se na metodu skrytých Markovových modelů. V páté kapitole si představíme nástroje a data, která jsou použita k provedení experimentu. Zamyslíme se nad způsobem, jakým bude vše provedeno a jak ze získaných poznatků vyvodíme patřičné závěry. Popíšeme přípravu a postup vlastního experimentu. V závěru shrneme informace, které jsme získali.

2 Zpracování řečového signálu

Aby bylo možné rozpoznávat řeč, je nutné nejprve popsat řečový signál způsobem, při kterém budou zachovány pouze potřebné informace. Navíc takovým způsobem, který bude dobře zpracovatelný strojem. Tuto přípravu nazýváme parametrizace a jedná se v podstatě o vyjádření řečového signálu omezeným množstvím hodnot, skalárů nebo vektorů. Parametrizace je často založena jak na poznatcích o zpracování signálu, tak na znalostech o tvorbě řeči v řečovém ústrojí.

V této kapitole si řekneme něco o způsobu, jakým vzniká řeč. Popíšeme si operace, které jsou provedeny před vlastní parametrizací. Představíme si Mel-frekvenční kepstrální koeficienty, což jsou parametry, kterými budeme řečový signál popisovat.

2.1 Vznik řečového signálu

Lidská řeč je souvislý časově proměnný proces. Vzniká s pomocí artikulačního traktu a je unikátní pro každého člověka. Jedná se o sekvenci zvuků, které jsou výsledkem ovlivňování základního hlasivkového tónu nebo samotného výdechového proudu vzduchu vytlačovaného z plic nastavením hlasového ústrojí. Tím vznikla jistá různorodost, kde kromě informace o obsahu promluvy nese řeč navíc i barvu, intonaci a jiné složky.

Vycházíme z předpokladu, že řeč je tvořena navazujícími segmenty. To jsou nejmenší foneticky odlišné jednotky a nazýváme je fonémy. Z fonémů pak můžeme poskládat jednotlivé slabiky a z těch následně celá slova. Hlasovým projevem fonému je hláska. Právě rozpoznáváním jednotlivých fonémů je možné rozpoznávat řeč. Musíme si ovšem uvědomit, že řeč se utváří změnou parametrů hlasového traktu a tyto změny nevznikají skokové, ale mají jistou setrvačnost. Ta je dána silou svalů a setrvačností při přechodech traktu z jednoho stavu do druhého. Z toho by nám mělo být jasné, že jednotlivé hlásky jsou ovlivněny v závislosti na předchozích hláskách a zároveň mohou být ovlivněny hláskami, které budou následovat. Tento jev nazýváme koartikulace a ta způsobuje, že stejné hláska díky vazbě s předchozí hláskou mohou znít pokaždé jinak. Proto lze vedle fonémů použít při rozpoznávání jednotky, které se snaží tyto vlivy postihnout. Jedná se především o trifóny, což je foném závislý na levém a pravém sousedovi.

2.2 Parametrizace řečového signálu

Pro rozpoznávání se nepoužívá řečový signál v původní podobě. Převádíme ho na soubor parametrů, který zachovává pro nás důležité informace. V první řadě je signál ustředněn. Stejnosměrná složka signálu nenese pro rozpoznávání žádnou užitečnou informaci. Tato složka

by mohla být dokonce při dalším zpracování signálu rušivá. Ustřednění provedeme odečtením střední hodnoty. Pokud se jedná o dostatečně krátký a ukončený signál, můžeme spočítat střední hodnotu prostým průměrováním po ukončení signálu vztahem:

$$\overline{s} = \frac{1}{N} \sum_{n=1}^{N} s[n] \tag{1}$$

Pokud ovšem signál není ukončený anebo je příliš dlouhý, musíme střední hodnotu odhadovat rekurzivně:

$$\overline{s}[n] = \gamma\, \overline{s}[n-1] + (1-\gamma)\, s[n], \tag{2}$$

kde $\gamma \rightarrow 1$.

Dalším krokem je rozdělení signálu na malé segmenty. To má jednoduchý důvod. Řeč je signálem náhodným. Pro metody odhadu parametrů by měl být stacionární. Pokud dojde k rozdělení signálu na malé úseky, může být v tomto úseku považován za stacionární. Délka segmentu by měla být dostatečně malá, aby byl signál v segmentu stacionární, ale na druhou stranu musí mít také určitou délku pro přesný odhad požadovaných parametrů. Délka jednoho rámce se obvykle pohybuje v rozmezí 10 – 30 milisekund.

Pro výběr signálu do rámců používáme tzv. okénkové funkce. Mezi dvě nejpoužívanější patří pravoúhlé okno a Hammingovo okno. Při aplikaci okénkových funkcí se mění spektrum vybíraného signálu, protože vynásobením signálu v časové oblasti dochází vlastně ke konvoluci spektra řeči se spektrem okna. Pravoúhlé okno je selektivnější než Hammingovo okno, ale na druhou stranu zatíží signál většími vysokofrekvenčními komponentami. Hammingovo okno je naopak mnohem čistší, ale je také méně selektivní.

Vypočítáme krátkodobou energii. Ta se využívá při rozdělování hlásek na znělé, které jsou vysokoenergetické, a neznělé, které mají malou energii. Může nám také posloužit jako detektor řečové aktivity, ale je nutné si uvědomit, že pro detekci řeči a ticha v šumu nebude fungovat, protože nízkoenergetické hlásky budou s tímto šumem splývat a tudíž zůstanou zamaskovány. Výpočet můžeme provést následovně:

$$E = \frac{1}{l_{ram}} \sum_{n=0}^{l_{ram}-1} x^2[n] \tag{3}$$

Tento parametr je logaritmován a je proveden v každém rámci.

Další parametr by mohl být počet průchodů nulou. Tento počet určuje, kolikrát za jeden rámec projde signál nulou. Parametr je možné použít pro rozlišení znělých a neznělých hlásek, kdy neznělé mají hodně průchodů nulou a znělé málo. Je ovšem použitelný pouze tehdy, jestliže máme signál správně ustředněn. Počet průchodů nulou je citlivý na šum a zde ho používat nebudeme. Využijeme jiné parametry a těmi budou Mel-frekvenční kepstrální koeficienty.

2.2.1 Mel-frekvenční kepstrální koeficienty – MFCC

Pokud bychom zkoumali lidské ucho, zjistili bychom, že má lepší rozlišovací schopnost na nízkých frekvencích než na frekvencích vysokých. U MFCC na frekvenční osu rozmístíme nelineárně filtry. Měříme energii na jejich výstupu a tu pak použijeme při výpočtu kepstra. Při konstrukci filtrů můžeme také nelineárně upravit frekvenční osu a pak na ni rozmístit filtry rovnoměrně. Pokud použijeme nelineární úpravu, musíme převést Hertzy na Mely s použitím vztahu:

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}). \tag{4}$$

Pro výpočet energií z jednotlivých frekvenčních pásem použijeme diskrétní Fourierovu transformaci, umocníme, vynásobíme trojúhelníkovým oknem a sečteme. Provedeme zpětnou Fourierovu transformaci logaritmů těchto energií. Inverzní Fourierovu transformaci nahradíme pomocí diskrétní kosinové transformace:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos(\pi \frac{i}{N}(j - 0{,}5)), \tag{5}$$

kde N je počet kanálů filtrů. Výsledkem jsou pak Mel-frekvenční kepstrální koeficienty. V takto volené parametrizaci si musíme najít kompromis mezi počtem kepstráních koeficientů. Větší počet těchto koeficientů lépe popíše řečový signál, ale je obtížnější je generalizovat. Pro popis rámce často volíme dvanáct kepstrálních koeficientů, přičemž nultý vypočtený koeficient občas nebývá použit. Tento koeficient vyjadřuje energii rámce. Pro zvýšení kvality popisu rámce se přidává aproximace první a druhé časové derivace každého koeficientu. Tyto nově vzniklé koeficienty se nazývají delta a delta-delta koeficienty.

Kepstrální koeficienty mají ještě jednu nepříjemnou vlastnost. Koeficienty vyšších řádů mají numericky malé hodnoty, což má negativní efekt na pozdější klasifikaci. Z tohoto důvodu bývá v dalším kroku sada koeficientů vyrovnána pomocí funkce:

$$c'_n = (1 + \frac{L}{2} \sin \frac{\pi n}{L}) \cdot c_n. \tag{6}$$

Parametr L volíme s ohledem na počet zvolených koeficientů. Operaci, kterou takto provedeme, nazýváme *liftering*.

Mel-frekvenční kepstralní koeficienty jsou používány v mnohých aplikacích realizujících rozpoznávání. Umožňují totiž množství efektivních manipulací a mají dobré rozlišovací schopnosti.

3 Skryté Markovovy modely

Touto kapitolou si představíme skryté Markovovy modely a ukážeme si algoritmy, které se spolu s nimi využívají, aby bylo možné s jejich pomocí realizovat rozpoznávání řeči. Navíc si zde ukážeme některé úpravy, které nám umožní snadněji přejít k rozpoznávání spojité řeči.

Skrytý Markovův model je matematickým aparátem. Odpovídá statistickému modelu automatu s konečným počtem stavů. Můžeme ho použít pro popis nestacionárních signálů, třeba řeči. Použití skrytých Markovových modelu při rozpoznávání řeči vychází z představy, že řečový signál lze popsat posloupností úseků. Tyto úseky lze považovat za stacionární a jsou navzájem propojeny přechodovými oblastmi. Stavy skrytého Markovova modelu, popsané spojitou směsí gaussovských rozložení hustot pravděpodobnosti, představují jednotlivé úseky řečového signálu, kterým mohou odpovídat například části slov či části jednotlivých hlásek. Doby trvání stavu jsou popsány pravděpodobností setrvání ve stavu. Přechody mezi stavy charakterizuje pravděpodobnost přechodu mezi stavy.

Tato metoda je v současné době jednou z nejpoužívanějších metod rozpoznávání řeči. Používáme ji při rozpoznávání izolovaných slov i souvislých promluv. Jako vzory mohou být použity celá slova nebo i menší úseky, jako jsou fonémy. Rozpoznávání může být na mluvčím jak nezávislé, tak závislé. Modely vzorů jednotlivých úseků právě vyjadřují statistický popis rozložení parametrů, k jejichž odhadu je potřeba dostatečně velké množství trénovacích dat.

3.1 Úvod do skrytých Markovových modelů

Pro jednoduchost si ukážeme princip skrytých Markovových modelů na rozpoznávání izolovaných slov. Následující odvození je z dokumentace k HTK. [1]

Slovo je reprezentováno sadou řečových vektorů parametrů $\boldsymbol{O}$.

$$\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \ldots \boldsymbol{o}_T, \tag{7}$$

kde $\boldsymbol{o}_t$ je řečový vektor odpovídající času t. Rozpoznávání izolovaných slov pak můžeme vyjádřit následujícím výpočtem

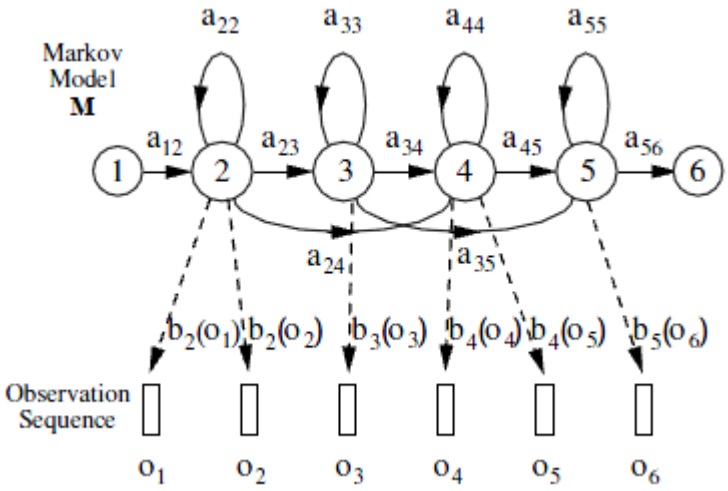
$$\arg\max_i \{P(\omega_i|\boldsymbol{O})\}, \tag{8}$$

kde ω_i je i-té slovo ve slovníku. Tuto pravděpodobnost nejsme schopni spočítat přímo. Můžeme ovšem využít Bayesovo pravidlo a pak dostaneme:

$$P(\omega_i|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|\omega_i)P(\omega_i)}{P(\boldsymbol{O})}, \tag{9}$$

kde $P(\omega_i)$ je pravděpodobnost vlastního slova a je předem dána. Ve výsledné pravděpodobnosti slova má největší váhu $P(\boldsymbol{O}|\omega_i)$.

Při ilustraci použití skrytých Markovových modelů předpokládáme, že řada vektorů parametrů je vygenerovaná podle následujícího obrázku:



obr. 1: Schéma generování vektorů parametrů markovovským modelem.

Převzato z [1]

Pravděpodobnost sekvence $\boldsymbol{O}$ generované modelem M pohybem přes posloupnost stavů X, pak spočítáme jednoduchým součinem výstupních stavů pravděpodobnosti $b_i(\boldsymbol{o}_t)$ s přechodovými pravděpodobnostmi a_{ij}. Pro situaci vyobrazenou výše tedy platí následující:

$$P(\boldsymbol{O}, X|M) = a_{12} b_2(\boldsymbol{o}_1) a_{22} b_2(\boldsymbol{o}_2) a_{23} b_3(\boldsymbol{o}_3) \ldots, \tag{10}$$

přičemž je nám známa pouze sekvence $\boldsymbol{O}$, zatímco posloupnost stavů X pro nás zůstává neznámá. Z tohoto důvodu se skrytým Markovovým modelům říká skryté. Protože neznáme posloupnost stavů X, vyjádříme požadovanou pravděpodobnost jako sumu přes všechny sekvence stavů $X=x(0),x(1),...x(T)$.

$$P(O|M) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(\boldsymbol{o}_t) a_{x(t)x(t+1)}, \tag{11}$$

kde vstupní stav $x(0)$ a výstupní stav $x(T+1)$ jsou neemitující stavy.

K výpočtu pravděpodobnosti přes všechny stavy můžeme také použít aproximaci, která využívá nejvíce pravděpodobnou posloupnost stavů:

$$\hat{P}(\boldsymbol{O}|M) = \max_X \left(a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(\boldsymbol{o}_t) a_{x(t)x(t+1)} \right). \tag{12}$$

Přímý výpočet těchto dvou rovnic není jednoduchý, ale naštěstí existuje rekurzivní způsob výpočtu, který je velmi efektivní. Tento způsob vychází z dynamického programování.

Vyřešíme-li vztah (8), je problém rozpoznávání ukončen. Toto řešení nalezneme pomocí vztahu (9). K množině modelů M_i, které odpovídají slovům ω_i, jsme dostali k analyzovanému signálu nejpravděpodobnější vzor. Vše platí za předpokladu:

$$P(\boldsymbol{O}|\omega_i)=P(\boldsymbol{O}|M_i) \tag{13}$$

Zároveň předpokládáme, že pro každý model M_i jsou nám známe parametry a_{ij} a $b_j(\boldsymbol{o}_t)$.

Právě zde leží síla skrytých Markovových modelů. Skupinou trénovacích vzorů a s pomocí robustní a velmi účinné re-estimační metody jsme schopni tyto parametry získat. Výstupní parametry $b_j(\boldsymbol{o}_t)$ jsou popsány směsí gaussovských rozložení hustot pravděpodobností:

$$b_j(\boldsymbol{o}_t)=\sum_{m=1}^{M_{js}} c_{jm} N(\boldsymbol{o}_t;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm}). \tag{14}$$

Tento vztah pro výpočet pravděpodobnosti stavu pro pozorovaný vektor parametrů $\boldsymbol{O}$ vzhledem k části modelu charakterizovaného střední hodnotou $\boldsymbol{\mu}$ a varianční maticí $\boldsymbol{\Sigma}$ je sumou přes všechny směsi gausovských hustot rozložení pravděpodobnosti. $N(\boldsymbol{o})$ představuje právě tyto směsi:

$$N(\boldsymbol{o};\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{o}-\boldsymbol{\mu})}. \tag{15}$$

3.2 Baum-Welchova re-estimace

Pokud definujeme strukturu modelů, pak trénováním získáváme pravděpodobnosti přechodů a_{ij} a parametry rozložení hustot pravděpodobnosti pro stavy modelů $b_j(\boldsymbol{o}_t)$

$$b_j(\boldsymbol{o}_t)=\frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}_j|}} e^{-\frac{1}{2}(\boldsymbol{o}_t-\boldsymbol{\mu}_j)'\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_j)}. \tag{16}$$

Za předpokladu, že by měl model jen jeden jednoduchý stav, můžeme zredukovat výpočet parametrů rozložení na jednoduché spočítání průměru přes všechny příslušné realizace. Vektor středních hodnot by pak byl:

$$\hat{\boldsymbol{\mu}}_j=\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{o}_t \tag{17}$$

a kovarianční matice by vypadala takto:

$$\hat{\boldsymbol{\Sigma}}_j=\frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{o}_t-\boldsymbol{\mu}_j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)'. \tag{18}$$

Ve skutečnosti ovšem máme modely obsahující několik stavů a neexistuje možnost přímého přiřazení jednotlivých stavů modelu a pozorovaných vektorů, protože se jedná o neznámou sekvenci. V prvním kroku zatím žádné natrénované modely nemáme, můžeme trénovací data rovnoměrně rozdělit takovým způsobem, aby měl každý ze stavů stejný počet vektorů. Pomocí vzorců (17) a (18) provedeme první odhadu parametrů modelu. V dalším kroku využijeme Viterbiho algoritmus, který

bude přesněji popsán dále. V podstatě dojde k přeskupení pozorovaných vektorů mezi jednotlivými stavy takovým způsobem, jímž budou maximalizovány výsledné pravděpodobnosti, jaké daný model vygeneroval pro dané vektory. Následně přepočteme jednotlivé parametry a celý proces opakujeme, dokud se budou měnit odhady výstupních parametrů a dokud bude výsledná pravděpodobnost významně růst.

Existuje ještě druhý postup, kdy je každý vektor přiřazen ke stavu modelu. Použijeme při tom ovšem takovou váhu, kterou by model vygeneroval v případě, že by se do stavu s tímto vektorem dostal. Pokud tedy je $L_j(t)$ pravděpodobnost bytí ve stavu j v čase t, změníme vztahy (17) a (18) takovým způsobem, že vzniknou vážené průměry:

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{t=1}^{T} L_j(t)\,\boldsymbol{o}_t}{\sum_{t=1}^{T} L_j(t)} \tag{19}$$

a

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{t=1}^{T} L_j(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)'}{\sum_{t=1}^{T} L_j(t)}. \tag{20}$$

Právě tento způsob se nazývá Baum-Wechlova re-estimace. Samozřejmě musíme ještě vyčíslit $L_j(t)$. K tomu využijeme tzv. Forward-Backward algoritmus. Dopředná pravděpodobnost $\alpha_j(t)$ pro model M se stavy N je definována jako:

$$\alpha_j(t) = P(\boldsymbol{o}_1 \boldsymbol{o}_2 \ldots \boldsymbol{o}_t, x(t) = j \mid M) \tag{21}$$

a jedná se o pravděpodobnost pozorování prvních t vektorů řeči ve stavu modelu j v čase t. Výpočet dopředné pravděpodobnosti provedeme s pomocí následující rekurze:

$$\alpha_j(t) = [\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij}] b_j(\boldsymbol{o}_t). \tag{22}$$

Tím vlastně vyjadřujeme fakt, že pravděpodobnost bytí ve stavu j v čase t a pozorovaného vektoru $\boldsymbol{o}_t$ je zredukována na sumu všech možných předcházejících stavů, které jsou váženy přechodovou pravděpodobností a_{ij}. Do sumy nejsou zahrnuty první a poslední stavy, protože se jedná o stavy neemitující. Počáteční podmínky pro rekurzi jsou nastaveny na:

$$\alpha_1(1) = 1 \tag{23}$$

a

$$\alpha(1) = a_{1j} b_j(\boldsymbol{o}_1) \tag{24}$$

pro $1 < j < N$. Koncová podmínka je nastavena následovně:

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)\, a_{iN}. \tag{25}$$

Dle definice dopředné pravděpodobnosti $\alpha_j(t)$ dostaneme:

$$P(\boldsymbol{O}|M) = \alpha_N(T) \tag{26}$$

a navíc získáme výpočtem dopředné pravděpodobnosti pravděpodobnost $P(\boldsymbol{O}|M)$.

Zpětná pravděpodobnost $\beta_j(t)$ je definována:

$$\beta_j = p(\boldsymbol{o}_{t+1}\boldsymbol{o}_{t+2}\ldots\boldsymbol{o}_T, x(t) = j|M). \tag{27}$$

I tato pravděpodobnost je počítána rekurzí:

$$\beta_i(t) = \sum_{i=2}^{N-1} a_{ij}\, b_j(\boldsymbol{o}_{t+1})\beta(t+1) \tag{28}$$

s počátečními podmínkami:

$$\beta_i(T) = a_{iN} \tag{29}$$

pro $1 < j < N$ a koncovou podmínkou danou:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j}\, b_j(\boldsymbol{o}_1)\beta_j(1). \tag{30}$$

Můžeme si všimnout, že dopředná pravděpodobnost je pravděpodobností vzájemnou, zatímco zpětná pravděpodobnost je podmíněná pravděpodobnost. Tato asymetrická definice nám umožňuje výpočet pravděpodobnosti obsazení stavu jako součin dvou pravděpodobností. Vycházíme z definice:

$$\alpha_j(t)\beta_j(t) = P(\boldsymbol{O}, x(t) = j|M). \tag{31}$$

Tím pádem tedy:

$$L_j(t) = P(x(t) = j|\boldsymbol{O}, M) \tag{32}$$

$$L_j(t) = \frac{P(\boldsymbol{O}, x(t) = j|M)}{P(\boldsymbol{O}|M)} \tag{33}$$

$$L_j(t) = \frac{1}{P}\alpha_j(t)\beta_j(t) \tag{34}$$

kde $P = P(\boldsymbol{O}|M)$.

Postup trénování modelů je následující:

1. alokujeme paměť,
2. spočítáme dopředné a zpětné pravděpodobnosti pro všechny časy t a všechny jednotlivé stavy j,
3. na základě předcházejících parametrů modelů přepočteme parametry a nově spočteme hodnotu $L_j(t)$,
4. spočítáme novou hodnotu $P(\boldsymbol{O}|M)$,

5. pokud se hodnota $P(\boldsymbol{O}|M)$ nezvětšila oproti předchozí hodnotě, zastavíme iteraci. V případě, že se hodnota zvětšila, pokračujeme v dalším cyklu k získání nových lepších parametrů.

Postup a vztahy počítají s tím, že pro jeden model máme pouze jednu trénovací promluvu. Pokud máme trénovacích dat více, postup není nikterak komplikován. Příslušné hodnoty jsou pouze akumulovány přes všechny tyto realizace a pak jsou spočteny příslušné pravděpodobnosti obsazení stavů $L_j(t)$.

3.3 Viterbiho algoritmus

V předchozí kapitole jsme mluvili o Baum-Welchově re-estimaci parametrů Markovova modelu. Při výpočtu dopředné pravděpodobnosti, kterou jsme tam zmiňovali, jsme použili rekurzivní algoritmus. Algoritmem jsme mimo jiné získali také hodnotu celkové pravděpodobnosti $P(\boldsymbol{O}|M)$. Tím pádem by tento algoritmus mohl být použit pro hledání modelu s maximální pravděpodobností $P(\boldsymbol{O}|M_i)$ a lze ho tedy využít při realizaci rozpoznávání. Následující postup je opět převzat z dokumentace HTK [1].

V praxi se nám ovšem více hodí výpočet maximální pravděpodobnosti sekvence stavů, protože to následně vede k jednoduššímu rozšíření na rozpoznání spojité řeči. Tato pravděpodobnost se počítá stejným způsobem jako dopředná pravděpodobnost. Provedeme pouze jednu malou úpravu, a tou je nahrazení sumy hledáním maxima. Pro daný model M nechť reprezentuje $\Phi_j(t)$ maximální pravděpodobnost pozorování vektoru parametrů $\boldsymbol{o}_1$ až $\boldsymbol{o}_t$ při dosažení stavu j v čase t. Tato částečná pravděpodobnost může být spočítána rekurzivně:

$$\phi_j(t) = \max_i \left\{ \phi_i(t-1)\, a_{ij} \right\} b_j(\boldsymbol{o}_t), \tag{35}$$

kde

$$\phi_1(1) = 1 \tag{36}$$

a

$$\phi_j(1) = a_{1j} b_j(\boldsymbol{o}_1) \tag{37}$$

pro $1 < j < N$. Maximální pravděpodobnost je pak dána:

$$\phi_N(T) = \max_i \{ \phi_i(T)\, a_{iN} \}. \tag{38}$$

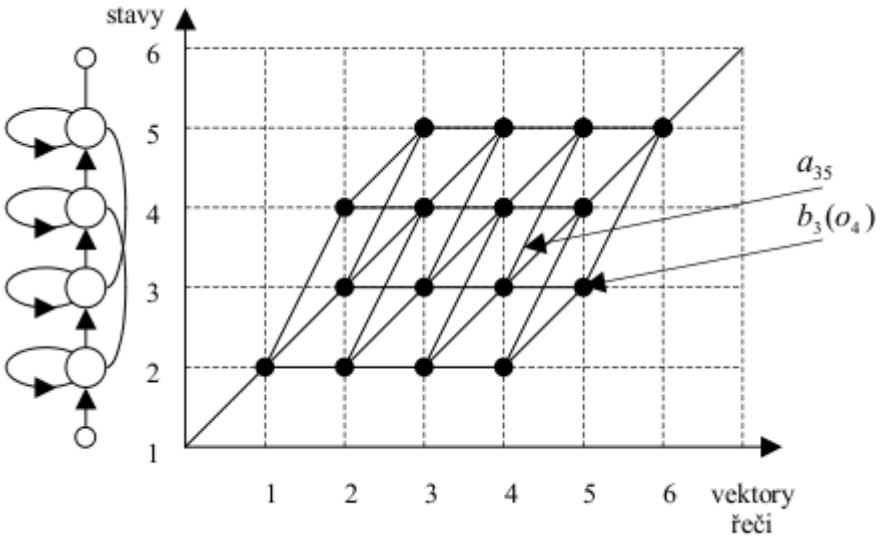
Logaritmické vyjádření algoritmu vede k lepším numerickým hodnotám:

$$\psi_j(t) = \max_i \{ \psi_i(t-1) + \log(a_{ij}) \} + \log(b_j(\boldsymbol{o}_t)). \tag{39}$$

Tato rekurze se nazývá Viterbiho algoritmus. Na obrázku 2 vidíme vizualizaci tohoto algoritmu. Můžeme si ho představit jako hledání nejlepší cesty maticí, kde vertikální osa představuje stavy ve skrytém Markovově modelu a horizontální osa jsou řečové rámce, posloupnost vektorů řečových koeficientů, tzn. čas. Každý bod v grafu reprezentuje výstupní logaritmickou pravděpodobnost,

že příslušný řečový rámec v daném čase je modelován příslušným stavem modelu. Každá úsečka, která spojuje tyto body, představuje logaritmickou přechodovou pravděpodobnost. Logaritmická pravděpodobnost kterékoli cesty je jednoduše spočítána jako součet logaritmických přechodových pravděpodobností a logaritmických výstupních pravděpodobností směrem zleva doprava. Pro každé t je pak známa každá parciální cesta $\psi_i(t-1)$ pro každé i. Rovnice (39) může být tudíž použita pro výpočet $\psi_i(t)$ a tedy pro posunutí v cestě o jeden časový rámec.

Tento koncept pro nalezení nejlepší cesty je velmi důležitý, protože jej lze zobecnit a použít v případech, kdy se snažíme rozpoznávat spojitou promluvu.



obr. 2: Viterbiho algoritmus. Převzato z [2]

3.4 Rozpoznávání spojité promluvy

Zatím jsme si vše ukazovali na modelech celých slov. Za každým modelem se skrývá velké množství úkonů, které je nutné provést. Jedním z nich je například získání dostatečného množství akustických dat. Dále také vytvoření jejich transkripce, definice modelů a nakonec jejich natrénování a otestování. Potom konečně získáme rozpoznávač. Ten bude ovšem omezen pouze na rozpoznávání slov, která jsou ve slovníku rozpoznávače a pro která byly dané modely natrénovány. Při rozšiřování bychom museli provést znovu kroky potřebné pro vytvoření modelů nových slov. Nutno zmínit fakt, že kdybychom chtěli rozpoznávat opravdu s velkým slovníkem, bylo by vytvoření modelů pro všechna slova velmi problematické. Bylo by vhodnější, kdybychom vytvořili modely, ze kterých bychom byli schopni nová slova poskládat. Pro tento účel zvolíme menší stavební kameny. Těmi jsou fonémy. Při rozpoznávání nepoužíváme slovo „foném“ ve významu fonetiky. Fonémem myslíme prvek z množiny všech fonémů bez ohledu na význam ve slově. Pakliže máme celou sadu modelů

fonému pro daný jazyk, jsme schopni snadno skládat tyto modely do slov, popřípadě i do sousloví nebo celých vět.

Modely fonémů se tvoří obdobně jako modely celých slov. Počet stavů modelů fonémů je většinou pro všechny fonémy stejný. Pro práci s fonémy je také nutné vytvořit velmi přesnou fonetickou transkripci slov. Musíme si uvědomit, že výslovnost se může od psané varianty výrazně lišit.

Následně jsme schopni na základě fonetických přepisů natrénovat příslušné markovovské modely fonémů. Je nutné si ovšem uvědomit, že pokud chceme vytvořit univerzální modely fonémů, musíme mít velké množství dat získané z co nejširšího množství populace. Mělo by se jednat o spojitou řeč.

Vlastní trénování modelů fonémů probíhá podle podobných principů a pravidel, jako v případě když tvoříme modely celých slov. Narazíme ovšem na odlišnosti, které je nutné překonat. V první řadě budeme mít problém s časovou transkripcí pro fonémy. Pro malé množství dat bychom mohli být schopni tuto transkripci vytvořit, ale fonémy jsou celkem malé úseky a proto nebudeme schopni nadefinovat pro jednotlivé fonémy přesné hranice. Časová náročnost pro větší množství dat by byla navíc neúnosná. Zvolíme tedy trochu jiný přístup. Takový, který je založen na Baum-Welchově re-estimaci. V tomto případě ovšem neprovádíme odhad pro každý model zvlášť, ale pracujeme se všemi modely najednou. Vzniklý algoritmus bude vypadat následovně:

1. alokujeme místo pro všechny parametry všech modelů,
2. vezmeme další trénovací promluvu,
3. sestavíme složený Markovovský model odpovídající spojení modelů odpovídajících symbolům v transkripci trénovací promluvy,
4. spočítáme dopředné a zpětné pravděpodobnosti a na jejich základě získáme také pravděpodobnost obsazení stavu.
5. využijeme dopředné, zpětné a okupační pravděpodobnosti v každém časovém rámci k aktualizaci příslušných sum pro všechny modely,
6. pokud máme nějaká další trénovací data, přejdeme ke kroku 2,
7. na základě hodnot, které jsme získali, vypočítáme nové odhady pro parametry modelů.

Tento postup můžeme opakovat. Většinou ho provádíme, dokud roste pravděpodobnost $P(O|M)$. Můžeme si také povšimnout, že již není vyžadována časová transkripce pro jednotlivé fonémy. Vystačíme si pouze s jejich sekvencemi pro jednotlivé promluvy.

Abychom mohli rozpoznávat spojitou promluvu, musíme provést celkem významnou změnu na Viterbiho algoritmu. Tato modifikovaná metoda se v angličtině nazývá *Token Passing Model.* Pracujeme s představou, že každý stav j skrytého Markovova modelu má v čase t právě jednu strukturu obsahující mimo jiné i informaci o částečné logaritmické pravděpodobnosti. Tuto strukturu

může stav předat dál a struktura představuje míru shody mezi promluvou $\boldsymbol{o}_1\boldsymbol{o}_2...\boldsymbol{o}_T$ a modelem promluvy s omezením, že model se nachází ve stavu *j* v čase *t*. Do této struktury pak v každém stavu, kterým prošla, akumulujeme hodnotu logaritmické pravděpodobnosti $\psi_i(t)$. Vlastní algoritmus bude vypadat následovně:

1. do každého vstupního stavu modelu umísti strukturu. Hodnota ve struktuře bude nula,
2. pro časy $t = 1 \ldots T$ pak v každém stavu *i*, který obsahuje strukturu s pravděpodobností, zkopíruj tuto strukturu do všech napojených stavů *j* a zvětši hodnoty logaritmické pravděpodobnosti v každé kopii o $log[a_{ij}]+log[b_i(\boldsymbol{o}(t))]$. Pokud se v nějakém stavu nachází více struktur, nech si tu, která obsahuje největší hodnotu s logaritmickou pravděpodobností, a ostatní smaž,
3. na konci ze všech stavů spojených s výstupním stavem zkopíruj strukturu a k hodnotě pravděpodobnosti přidej $log\ a_{iN}$. V posledním stavu si vyber tu strukturu, která obsahuje hodnotu s nejvyšší logaritmickou pravděpodobností, a ostatní zahoď.

Důvodem, proč zde mluvíme o *Token Passing modelu*, je snadné využití tohoto modelu v případě rozpoznávání spojité řeči. Předpokládejme, že se jedná o velkou síť s konečným počtem stavů, kde se v každém stavu schovává Markovovský model reprezentující jednotlivé fonémy. Pak jsme v této síti schopni poskládat z modelů požadovaná slova, případně i celé věty. V konečném důsledku se vlastně jedná o skrytý Markovovský model, který v jednotlivých stavech obsahuje další modely. V tuto chvíli nám chybí poslední věc a tou je informace o sekvenci stavů, ve které jsme modelem prošli. Tu můžeme zjistit jednoduchým přidáním informace do struktury používané v *Token Passing modelu*. Tato informace se týká cesty, kterou se struktura v modelu pohybovala. V praxi se zaznamenává opouštěný stav a čas, kdy jsme tento stav opustili. Pak jsme schopni zjistit cestu modelem a tím i identifikovat řečové úseky ve zkoumané promluvě.

4 Nastínění detekce klíčových slov

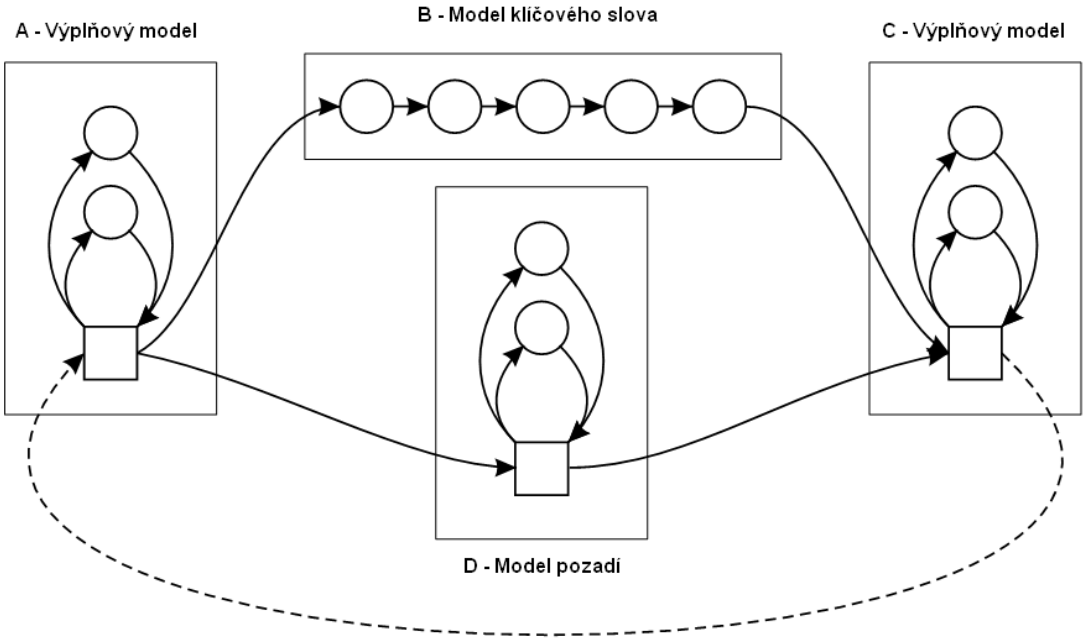
Do této chvíle jsme mluvili stále jen o rozpoznávání, které ovšem není cílem této práce. My se zde chceme zajímat o detekci klíčových slov. Přesto jsou výše napsané kapitoly pro nás důležité. Důvod je zřejmý. Detekce klíčových slov se zakládá na podobných metodách a principech, které se využívají při klasickém rozpoznávání. Rozdíl mezi detekováním a rozpoznáváním slov spočívá v tom, že zatímco rozpoznávač se snaží v řečovém signálu zjistit celou informaci, detektor pouze posuzuje, zdali se v promluvě nevyskytuje pro nás zajímavý úsek řeči, což nemusí být jednotlivá slova, ale může se jednat o celé fráze. Při detekci nám může dělat problém hledání neznámých slov, která často mohou nést velmi důležitou informaci. Příkladem jsou třeba jména. Při realizaci hlasového ovládání nás ovšem problém s neznámými slovy nezatěžuje.

V dnešní době se k detekci klíčových slov používají nejvíce metody založené na skrytých Markovových modelech.

4.1 Metoda skrytých Markovových modelů

Víme, že klíčové slovo je obklopeno okolím, které není pro naše účely zajímavé. S pomocí skrytých Markovových modelů jsme schopni tento fakt simulovat. Do systému, který realizuje detekci klíčových slov, vstupují parametrizované řečové signály. Výstupem je hodnota, která určuje, s jak velkou jistotou se jedná o klíčové slovo. Podle této hodnoty nakonec rozhodneme, zda se o jedná klíčové slovo či nikoliv. V této kapitole vycházíme z [10].

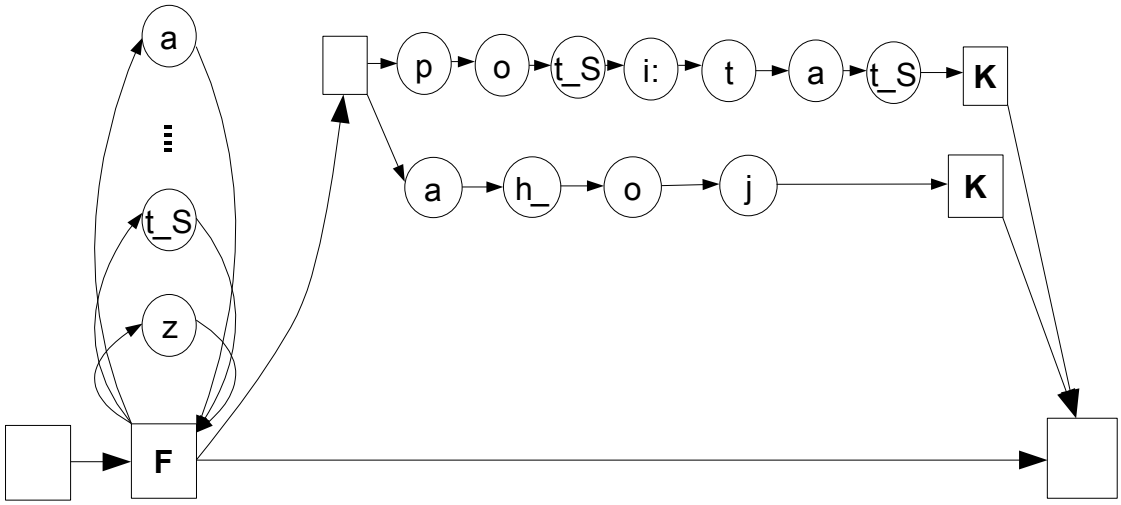
Představme si systém založený na celých slovech. Vytvoříme modely jednotlivých klíčových slov a k nim utvoříme model, který bude zahrnovat všechna ostatní nezajímavá slova. Model výplně a model pozadí je natrénován z promluv, které neobsahují klíčová slova. Problémem tohoto přístupu je, že některá klíčová slova mohou být nepřímo obsažena v neklíčové části promluv. Například pokud bychom měli klíčové slovo „*ranč*“ a v neklíčové části promluvu „pome*ranč*ový džus“. Další nevýhodou je závislost modelů neklíčové části na trénovacích datech.



obr. 3: Systém detekce klíčových slov. Převzato z [10].

Pokud použijeme systém, kde budou základním stavebním kamenem menší prvky, než jsou slova, mohli bychom některé problémy, které celá slova mají, odbourat. K tomuto účelu se hodí modely fonémů. Pak k trénování používáme promluvy obsahující i klíčová slova. Výplňový model a model pozadí tvoříme fonémovou smyčkou. Klíčová slova jsou natrénována zvlášť anebo poskládána z modelů fonémů. Detektor klíčových slov tvořený pomocí fonémů je na obrázku 3. Na začátku a na konci je fonémová smyčka, která představuje výplňový model. Tím jsou modelovány neklíčové části promluv. Model klíčového slova, obsahující modely jednotlivých fonémů, je mezi vyplňovými modely. Paralelně k modelu klíčového slova je model pozadí. Ten tvoří stejná fonémová smyčka jako výplňový model. Promluva je modelována modelem ABC s modely A, B, C a modelem ADC s modely A, D a C. Budeme předpokládat, že modely B a D budou zpracovávat oba stejnou část promluvy. V takovém případě se hodnoty, dané průchodem modely ABC a ADC, liší pouze díky modelům B a D. V případě, že část promluvy v modelu B není klíčové slovo, je hodnota malá, zatímco model pozadí dá hodnotu větší, protože dobře „sedí“ na fonémovou smyčku. Pokud promluva obsahuje klíčové slovo, pak model B generuje vysokou hodnotu. Stejně tak model D generuje vysokou hodnotu. V ideálním případě budou obě generované hodnoty stejné. Obě získané hodnoty porovnáme. Ve výsledku detekce pak získáme možné detekce klíčového slova, kde bude každá z nich ohodnocená takzvaným skóre klíčového slova. Pokud se v dané části promluvy vyskytuje klíčové slovo, je tato hodnota ideálně rovna nule. Vlastní skóre se pohybuje v záporných hodnotách, přičemž čím nižší hodnota skóre, tím menší je podle našeho rozpoznávače šance, že se v daném místě nachází hledané slovo. Tato hodnota vychází záporná, protože porovnávání provádíme

pomocí rozdílu výstupů modelů ABC a ADC. Obrázek 4 představuje realizaci detekce klíčových slov. Klíčová slova jsou paralelně poskládána napravo. Pokud dané slovo má více výslovnostních variant, musíme zahrnout všechny možnosti. Výplňový model je nalevo. Tento model zároveň představuje i model pozadí.



obr. 4: Síť pro detekci klíčových slov založených na modelech fonémů.

Takto realizovaná detekce klíčových slov ohodnotí promluvu pro každé klíčové slovo. Proto je třeba nastavit hranici, kdy ještě považujeme slovo za klíčové. Z hlediska celkového výsledku je nutné zvolit dobrý kompromis mezi počtem falešných poplachů a úspěšně zachycenými slovy. Tato hranice se většinou nastavuje na základě testů provedených na natrénovaných modelech a také na okolnostech, při kterých bude systém použit. V některých různých případech je nezachycení či zachycení slova různě důležité. Pokud budeme mít systém, který bude ovládat osvětlení místnosti, pak nám nebude vadit, že ho musíme o rozsvícení nebo zhasnutí světel požádat dvakrát, ale rozhodně nám bude vadit, že světla se samovolně rozsvěcí a zhasínají, když vedeme běžný dialog s nějakou osobou v místnosti. Pokud budeme mít bezpečnostní systém hledající klíčové fráze v telefonních hovorech, budeme mít nastavený systém takovým způsobem, aby zachytil raději více falešných poplachů. Na druhou stranu tím zvětšíme počet správně zachycených slov a zmenšíme pravděpodobnost, že mineme pro nás klíčovou frázi.

5 Detekce klíčových slov modely fonémů

Do této chvíle jsme si představovali teorii, která je nutná k realizaci rozpoznávání řeči a detekce klíčových slov. V této kapitole si představíme experiment. Rozhodneme o způsobu, jakým bude realizován. A popíšeme jeho realizaci. Zhodnotíme výsledky experimentu. Také si zde povíme, jaké nástroje k experimentu použijeme.

5.1 Diskuze problému

Naším cílem je s použitím metody skrytých Markovových modelů natrénovat několik sad modelů a diskutovat jejich úspěšnost. Tyto sady modelů mají být natrénovány na promluvách zachycených z různých vzdáleností od mluvčího. Pak máme diskutovat úspěšnost těchto sad modelů. K dispozici jsou trénovací sety získané z databáze SPEECON. K trénování modelů a jejich testování využijeme nástroj HTK.

Před trénováním modelů si můžeme zvolit, zda budeme pracovat s modely celých slov anebo použijeme modely fonému. Bylo rozhodnuto použít univerzálnější modely, a proto budou trénovány modely fonémů, ze kterých budeme později skládat celá slova a výplňový model. Díky tomu se nebudeme muset tak pečlivě zabývat výběrem slov, která budeme detekovat, protože později můžeme poskládat slova jiná.

Provedení experimentu by mohlo vypadat následovně. Z dat, která máme k dispozici, vezmeme tři trénovací sady. Tyto sady budou obsahovat řečová data se stejnou transkripcí, ale budou zachycena třemi různými mikrofony nacházejících se v různých vzdálenostech. Na každé z této sad natrénujeme modely fonémů, které následně použijeme k detekci klíčových slov na třech různých sadách testovacích dat. Tyto sady budou také pocházet ze SPEECONU. Tyto testovací sady budou obdobně jako sady trénovací získány mikrofony v různých vzdálenostech. Obsah promluv testovacích sad bude stejný. Poté vytvoříme ještě dvě sady modelů určených k detekci. Tyto sady budou natrénovány na promluvách obsahujících data z dvou různých vzdáleností. Nakonec zhodnotíme všechny výsledky a vyvodíme z nich patřičné závěry.

5.2 Použité nástroje a zdroje dat

5.2.1 HTK

Hidden Markov Model Toolkit je sadou nástrojů z univerzity Cambridge. Slouží k definici, trénování a rozpoznávání řeči. Obsahuje také moduly provádějící parametrizaci, vyhodnocování

výsledku, práci s výslovnostními slovníky, tvorbu transkripce a další. Skládá se z jednotlivých bloků, které mají specifické funkce. Více o tomto nástroji lze vyčíst z dokumentace HTK [1].

5.2.2 Databáze SPEECON

Jedná se o rozsáhlou databázi českých promluv, která vznikla ve spolupráci FEL ČVUT v Praze a FIT VUT v Brně. Obsahem jsou promluvy od velkého spektra mluvčích. Jedná se jak o spontánní řeč, tak o čtenou předlohu. Promluvy obsahují samostatná slova i celé věty. Data jsou získána z různých mikrofonů. Více informací se lze dočíst v dokumentaci [8].

5.2.3 Další nástroje

Navíc byly při této práci použity moduly SLRatio_new verze 0.6 a SResults verze 0.1. Oba byly vytvořeny na FIT VUT v Brně. SResults byl použit pro vyhodnocování výsledků a SLRatio_new bylo použito při detekci klíčových slov. SLRatio_new vlastně implementuje postup, který byl popsán v kapitole 4.1.

5.3 Způsoby vyhodnocení

Získané informace je nutné nějakým způsobem vyhodnotit a porovnat. Výstupem z detekce klíčových slov bude soubor hodnot, ze kterých se pak dále budeme podle nastaveného prahu rozhodovat, zda se jedná o klíčové slovo či nikoliv. Tento výstup následně porovnáme s předlohou, která obsahuje slova, jež jsou v testovacích promluvách obsažena. Při tom může nastat několik variant. Pokud slovo označíme jako slovo klíčové, mohou nastat dvě možnosti. V daném místě se klíčové slovo opravdu nachází a my ho označili, pak jsme se „trefili“ a jedná se o tzv. *hit*. V případě, že na tomto místě klíčové slovo není a detektor tvrdí opak, jedná se falešný poplach (tzv. *False Alarm*). K těmto dvěma případům existuje ještě další možnost, tou je přehlédnutí klíčového slova. Tento případ nazýváme minutí (tzv. *miss*).

Pro naše potřeby je nutné nějakým způsobem ohodnotit kvalitu detekce daných modelů na dané testovací sadě. To navíc potřebujeme provést způsobem, který by se dal snadno porovnávat. Využijeme proto parametr zvaný *Figure of Merit,* zkratka FOM. Jeho výhodou je, že se jedná o jedinou hodnotu, která je vyjádřena v procentech a bude se velmi snadno porovnávat. Jde o průměrnou úspěšnost detekce na intervalu od jednoho do deseti falešných poplachů na klíčové slovo a hodinu.

5.4 Trénovací sety

K trénování použijeme pětici sad, jejichž data jsou z databáze SPEECON. Sady jsou poskládány z promluv s foneticky bohatými větami, kde z novin a několika knih od klasických českých autorů byly zvoleny foneticky bohaté věty. Při výběru se vyřazovaly hlavně některé velmi dlouhé, pravopisně chybné či gramaticky divné věty. Ve výsledku obsahuje tato část SPEECONU 3300 různých vět a každá tato věta se opakuje pětkrát. Dalším typem promluv jsou promluvy s foneticky bohatými slovy. Obsahem těchto promluv je 300 různých slov. Ta byla volena s ohledem na fonémy, ze kterých se skládají. Všechna použitá slova jsou korektní česká slova, i když některá jsou méně používaná a některá se používají pouze v určitých specializovaných oborech. Ve SPEECONU se každé ze zvolených slov objevuje nejvýše v deseti případech. Třetím typem promluv, ze které byla volena trénovací data, jsou promluvy se spontánní řečí, kdy mluvčí odpovídal podle určitého scénáře. Celkově se jedná o 30 témat, která lze rozdělit do čtveřice skupin. První je ovládání mobilního telefonu či PDA. Druhou je výzva pro automatický informační stánek. Třetí skupina se týká televize či Hi-Fi, jejich pořadů a ovládání. Toto téma také hovoří o oblíbeném filmu či sportovci. Do poslední skupiny jsou řazeny všechny ostatní scénáře. Zaznamenána je pouze reakce respondenta nikoli položená otázka. Nahrávající v případě nejasností vysvětlil respondentovi jeho úkol, takže všechny promluvy obsahují vhodnou reakci na danou situaci. Promluvy jsou zachyceny v kanálech nula až tři (to platí pro všechny promluvy, které byly použity pro sestavení trénovacích sad, ale u některých promluv v databázi tomu tak není). Sady jsou tvořeny z kanálů nula, jedna a tři. Kanál dva není použit. Kanálu nula odpovídají data z náhlavního mikrofonu. Kanál jedna jsou data z mikrofonu blízko před mluvčním a třetí kanál jsou data z mikrofonu dále od mluvčího. Více informací o datech a jejich způsobu zaznamenání lze získat z dokumentace SPEECON [8].

Každá ze sad obsahuje 21057 promluv. Promluv s foneticky bohatými větami je 12610. Promluv s foneticky bohatými slovy je 2732 a promluv se spontánní řečí je 5716. První set je postaven z promluv získaných z náhlavního mikrofonu. Druhý set se skládá z promluv nahraných mikrofonem, který se nachází blízko před mluvčím. Třetí set je vytvořen z dat zachycených z mikrofonu dále od mluvčího. Všechny tyto sety obsahují stejná slova i stejné věty a liší se pouze vzdáleností, ze které byly mikrofonem zaznamenány. Čtvrtý set vznikl spojením části prvního setu s částí setu druhého. Poslední set jsme poskládali z části druhého a třetího setu. Viz tabulka č. 1 uvedená dále.

5.5 Testovací sety

K otestování modelů postavíme tři sady. Ty jsou, stejně jako trénovací data, získána z databáze SPEECON. Každá ze tří sad obsahuje 1317 promluv, jejichž obsahem je přibližně 140 minut řečového signálu. Z toho je 303 promluv spontánní řeči, 146 promluv s fonémově bohatými slovy a 868 s fonémově bohatými větami. Jsou použita data z kanálů nula, jedna a tři. Kanál dva není použit.

První testovací set je poskládán z promluv získaných z náhlavního mikrofonu. Druhá sada je sestavena z dat zaznamenaných mikrofonem, který se nachází blízko před mluvčím. Třetí sadu máme nahranou mikrofonem umístěným dále od mluvčího. Viz v tabulce č. 1. Obdobně jako u trénovacích sad i v tomto případě promluvy testovacích sad obsahují stejná slova a stejné věty.

Aby se nám jednotlivé sady od sebe lépe odlišovaly, pojmenujeme si první testovací sadu **head_test**. Druhou nazveme **close_test** a třetí **far_test**. Testovací sada **far_test** představuje promluvy zachycené ze vzdálenosti, ve které by mohl operovat skutečný systém pro ovládání spotřebičů hlasem.

Trénovací sady		**Testovací sady**	
Sada fonémů	**Data z kanálu**	**Sada**	**Data z kanálu**
head_set	nula	head_test	nula
close_set	jedna	close_test	jedna
far_set	tři	far_test	tři
comb_set	nula a jedna		
clofar_set	jedna a tři		

Tab. 1 – Data jednotlivých trénovacích a testovacích sad

5.6 Klíčová slova

Abychom mohli otestovat úspěšnost systému, je třeba zvolit klíčová slova. Pro náš případ bylo vybráno 56 klíčových slov, která se v testovacích promluvách objevují v různých počtech. Jsou různě dlouhá a některá z nich si jsou navzájem podobná. Testovací sada obsahuje celkem 691 výskytů všech klíčových slov. Seznam klíčových slov a počty jejich výskytů jsou v tabulce č. 2. Klíčová slova v tabulce jsou uspořádána podle abecedy.

Klíčové slovo	Počet výskytů klíčového slova	Klíčové slovo	Počet výskytů klíčového slova	Klíčové slovo	Počet výskytů klíčového slova
ahoj	10	nádraží	18	příroda	4
automobilu	6	nahoru	4	ráda	19
byla	22	nahrávání	5	rádio	6
července	4	nakonec	5	skutečně	3
další	13	nejoblíbenější	4	sportovec	5
děkuji	10	několik	5	televize	12
dobrý	65	nervózní	6	televizi	21
doprava	9	nikdo	9	třeba	19
dvacátého	8	oblíbený	11	úplně	4
hlavní	11	odpoledne	5	večer	9
hodin	9	otevřely	4	video	11
chtěla	33	počítač	10	vlastně	9
inkoust	3	podle	12	vodovod	0
jsem	85	podzimní	5	všechno	9
jsme	29	pokoj	21	vybledlý	3
který	11	poslední	6	vysvětlit	3
maminka	18	potřebovala	9	zajímalo	3
město	6	program	15	zarezervovat	7
milión	4	prosím	34		

Tab. 2 – Seznam použitých klíčových slov

5.7 Použité sady modelů

Jako nejvhodnější jednotka, se kterou budeme pracovat, byl zvolen foném. Z toho nám plynou některé důsledky. V první řadě nejsme omezeni žádným předem daným slovníkem. Díky tomu, kdyby to bylo potřeba, můžeme snadno zaměnit detekovaná slova, která jsme si zvolili pro náš experiment a jsou uvedena v tabulce č. 2, za libovolná jiná a přesto nebudeme muset sadu modelů měnit. Výplňový model a model pozadí bude vytvořen z paralelně postavených modelů fonémů, takže nemusíme trénovat speciální model.

Celkem budeme mít vytvořeno pět sad modelů. Každou z nich natrénujeme na jednom trénovacím setu. Tím nám vznikne jedna sada natrénovaná na promluvách z náhlavního mikrofonu. Tuto sadu nazveme **head_set**. Druhou sadou budou modely natrénované na datech z mikrofonu blízko před mluvčím. Sada modelů je označena jako **close_set**. Třetí je vytvořena za pomocí dat z mikrofonu dále od mluvčího. Tuto sadu označíme jako **far_set**. Sada **comb_set** je čtvrtou a je trénována na promluvách z náhlavního mikrofonu a mikrofonu blízko před mluvčím. Poslední sadu

nazveme **clofar_set** a je tvořena kombinací promluv z mikrofonu blízko před mluvčím a mikrofonu dále od mluvčího. Přehledně vyobrazeno v tabulce č. 1 uvedené výše.

Počet fonémů v sadě bude korespondovat s počtem symbolů, který popisuje česká SAMPA (jedná se o zkratku „*Speech Assessment Methods Phonetic Alphabet*“ a tyto symboly odpovídají fonémům české abecedy nebo spíše jejich výslovnostnímu ekvivalentu). Abychom mohli využít nástroje HTK, musíme tyto symboly trochu upravit. HTK totiž nepracuje s lomítkem. Proto ho nahradíme, tam kde je to potřeba, podtržítkem (v tabulce č. 3 jsou uvedeny symboly SAMPA včetně úpravy a příkladu). Každý tento symbol bude představovat jeden model fonému. Navíc budeme mít dva modely ticha.

SAMPA	Přepis pro HTK	Příklad slova	Příklad přepisu	SAMPA	Přepis pro HTK	Příklad slova	Příklad přepisu
i	i	myš	m i S	f	f	forma	f o r m a
e	e	les	l e s	v	v	vak	v a k
a	a	pas	p a s	s	s	sen	s e n
o	o	rok	r o k	z	z	zub	z u p
u	u	kus	k u s	Q\	Q_	tři	t Q_ i
i:	i:	pít	p i: t	P\	P_	řád	P_ a: t
e:	e:	lék	l e: k	S	S	šaty	S a t i
a:	a:	rád	r a: d	Z	Z	žal	Z a l
o:	o:	móda	m o: d a	j	j	jas	j a s
u:	u:	půl	p u: l	x	x	chata	x a t a
o_u	o_u	mouka	m o_u k a	h\	h_	had	h_ a t
a_u	a_u	auto	a_u t o	r	r	ret	r e t
p	p	pes	p e s	l	l	led	l e t
b	b	bota	b o t a	m	m	mák	m a: k
t	t	tam	t a m	n	n	noc	n o c
d	d	dům	d u: m	N	N	banka	b a N k a
c	c	tito	c i t o	J	J	nic	J i t_s
J\	J_	děd	J_ e d	l=	l=	vlk	v l= k
k	k	krk	k r k	m=	m=	osm	o s m=
g	g	kde	g d e	r=	r=	krk	k r= k
t_s	t_s	cíl	t_s i: l	F	F	tramvaj	t r a F v a j
d_z	d_z	leckdy	l e d_z g d i	G	G	abych byl	a b i G b i l
t_S	t_S	čas	t_S a s	e_u	e_u	euforie	e_u f o r i e
d_Z	d_Z	léčba	l e: d_Z b a	@	@	DTW	d @ t @ v @

Tab. 3 - Symboly SAMPA s přepisem pro HTK. Převzato z [9].

Modely fonémů, s nimiž pracujeme, mají pět stavů. První a poslední stav jsou tzv. neemitující. Tyto stavy nám slouží pouze k napojení na další nebo předcházející model. Také budeme mít natrénovány modely dlouhého a krátkého ticha. Model dlouhého ticha má pět stavů, první a poslední jsou stavy neemitující. Navíc ještě mezi čtvrtým a druhým existuje přechod tam i zpět. Tím se tento

model stává univerzálnější pro různé druhy šumů. Model krátkého ticha je pouze třístavový. První a třetí stav slouží k napojování. Prostřední stav tohoto modelu je napojen na třetí stav modelu dlouhého ticha.

5.8 Parametrizace a trénování modelů

Při přípravě modelů budeme u každé sady používat stejnou sekvenci kroků. Postup se bude lišit pouze trénovacími promluvami, které při tvorbě modelů použijeme. Výsledkem tohoto postupu bude vždy jedna sada modelů.

5.8.1 Příprava

Než přistoupíme k parametrizaci a trénování, musíme učinit některé kroky. Na začátku máme připraven seznam trénovacích dat. Nejprve potřebujeme udělat textový přepis řeči, kterou obsahují trénovací promluvy. Tento přepis jsme schopni získat z databáze SPEECON. Obsahem totiž nejsou pouze řečová data, ale nacházejí se zde k nim i soubory informací o daném signálu. Z těchto souborů získáme textový přepis řečového signálu. Ten upravíme na tzv. *master label file*. V něm je specifickým způsobem rozepsán textový přepis řečového signálu. Na začátku je cesta k souboru vzniklého parametrizací z řečového signálu. Následně jsou rozepsána všechna slova v promluvě, každé na samostatném řádku. Celý přepis je zakončen tečkou. Navíc je před první a za poslední slovo umístěna značka *sil*. Předpokládáme, že na začátku a na konci každé promluvy bude chvilka ticha. Právě tato místa se používají při přípravě modelu dlouhého ticha. Ukázka z takového souboru je uvedena na obrázku 5 (kde zatím nejsou přidány symboly *sil*):

```
"/homes/eva/xs/xsadec01/tmp/data/BLOCK00/SES000/SA000F16.lab"
čaj
hemenex
rohlík
.
```

obr. 5: master label file s obsahem slov

Z transkripce slov vytvoříme jiný *master label file*. V něm budou tato slova rozepsána na jednotlivé fonémy. K tomu potřebujeme slovník, seznam slov s jejich výslovnostním přepisem na symboly SAMPA. Ty musí být mírně upraveny, protože nástroj HTK neumožňuje použít při zápisu symbolu lomítko (všechna lomítka v symbolech SAMPA nahradíme podtržítkem). Pokud dané slovo obsahuje vícero výslovnostních variant, jsou tyto varianty ve slovníku uvedeny také. V tuto chvíli máme vše potřebné. Použijeme slovník s *master label file* obsahující celá slova a HLEd (jedná se o nástroj HTK, který slouží k úpravě *master label file* souborů). Tím vznikne *master label file* s obsahem

fonémů (viz krok 5 v readme na přiloženém CD). Příklad takové části souboru je na obrázku 6, kde jsou rozepsána slova „*čaj*“, „*hememex*“ a „*rohlík*“.

```
"/homes/eva/xs/xsadec01/tmp/data/BLOCK00/SES000/SA000F16.lab"
sil
t_S
a
j
h_
e
m
e
n
e
k
s
r
o
h_
l
i:
k
sil
.
```

obr. 6: master label file obsahující slova přepsaná na fonémy

Dále vytvoříme ještě druhý *master label file*. V něm budou mezi jednotlivá slova přidány symboly *sp*. Ty představují místo krátké pauzy, kterou děláme po každém slově, a budou použity k natrénování modelů krátké pauzy (viz krok 4 v readme na přiloženém CD). V příkladu na obrázku 6 by byly symboly *sp* přidány mezi *j* a *h_*, *s* a *r*. Nyní máme připravenou transkripci, kterou budeme potřebovat při trénování. Pro všechny sady trénovaných modelů nám stačí transkripce, kterou jsme právě vytvořili. Nakonec ještě vytvoříme soupisku všech modelů, které budeme trénovat. Ta bude ve dvou variantách jedna s modelem krátkého ticha a jedna bez něj.

5.8.2 Parametrizace řečových nahrávek

Máme připraven soubor obsahující cesty, kde se nacházejí vlastní zvukové záznamy. Připravíme si adresářovou strukturu (viz krok 1 v readme na přiloženém CD), ve které budou parametrizované soubory uloženy, a vytvoříme soupisku. Zde bude na jedné straně cesta k zvukovému záznamu a na straně druhé cesta, kam máme uložit soubor s parametry (viz krok 2 v readme na přiloženém CD). Také budeme potřebovat soubor s konfigurací. K parametrizaci použijeme část HTK s názvem HCopy (viz krok 3 v readme na přiloženém CD). Tento modul umí kopírovat. Navíc při kopírování umožňuje provést konverzi. Ta je závislá na obsahu konfiguračního souboru, který jsme si připravili. V našem případě parametrizaci na Mel-frekvenční kepstrální koeficienty. Na obrázku 7 níže můžeme vidět část parametrizovaného souboru. V tuto chvíli máme připraveny trénovací soubory s parametry. Také si připravíme seznam s cestami k parametrizovaným souborům. Kroky potřebné k parametrizaci musíme provézt pro každé trénování znovu, protože každou sadu modelů vytváříme na jiné trénovací sadě.

```
------------------------------------ Samples: 0->-1 -------------------------------------
0:      -4.282  -0.808   0.336   8.701   8.599   6.624   3.568   3.081   7.647   0.258
        -1.813  -3.581  -5.723
1:      -4.818   1.473   4.193   9.558   6.336   8.125   3.591   1.886   7.937  -2.036
        -2.747  -5.180  -6.762
2:      -4.302   4.324   5.354   6.168  10.829  14.631  10.052   1.555   3.725   0.025
        -0.306  -2.460  -7.251
3:      -3.655   2.774   4.121   5.861   3.536   8.916   3.743   2.105   3.784   0.084
         2.304  -2.780  -6.746
4:      -2.822   1.074   3.879   9.040   4.254   8.056   2.047  -0.130   4.131  -3.037
        -6.089  -4.427  -6.377
5:      -5.286   1.343   4.885   5.598   7.565  11.162   3.090  -1.441   6.002   1.598
         0.057   3.081  -7.272
```

obr. 7: Část souboru vzniklého při parametrizaci

5.8.3 Příprava modelů

Po přípravě transkripce a parametrizaci následuje příprava modelů, které budeme trénovat. Nejprve utvoříme prototyp modelu. Ten bude obsahovat pouze nulové hodnoty a jednoduchou matici přechodů naplněnou základními hodnotami. Následně z tohoto prototypu vytvoříme inicializovaný model. K tomu využijeme další z nástrojů HTK, a to HCompV. Tím provedeme inicializaci parametrů „vysílacích“ hustot rozdělení pravděpodobnosti ve stavech modelu (viz krok 7 v readme na přiloženém CD). Dalším krokem bude rozkopírování tohoto prototypu do jednotlivých modelů (viz krok 8 v readme na přiloženém CD). V tuto chvíli jsme získali modely pro fonémy. Jsou zatím všechny naplněny stejnými hodnotami. Přípravu modelů musíme pro každou sadu provést zvlášť, protože jejich inicializace probíhá na základě parametrizovaných promluv. Ty se pro každou sadu liší.

5.8.4 Trénování modelů

Máme vše potřebné k trénování modelů a můžeme začít trénování. Použijeme nástroj HTK s názvem HERest. Ten provádí výpočet „měkké“ funkce přiřazení jednotlivých vektorů ke stavům s následným přetrénováním parametrů.

V prvním kroku zahájíme základní trénování pomocí HERest (viz krok 9 v readme na přiloženém CD). Jako transkripci k parametrizovaným souborům v tuto chvíli používáme *master label file*, ve kterém jsou obsaženy fonémy a dlouhá pauza. Tím připravíme jednotlivé modely fonémů a model dlouhého ticha. V dalším kroku si připravíme model krátkého ticha. V tom je třeba propojit jeho prostřední stav s prostředním stavem modelu dlouhého ticha. K tomu použijeme HHEd (Jeho základní funkcí je načtení souboru obsahujícího skryté Markovovy modely a jejich následná editace. Ta je provedena na základě příkazů uvedených v řídícím souboru. Ten je parametrem modulu.), kterým provedeme toto propojení. Zároveň v tomto kroku utvoříme v modelu dlouhého ticha přechod z druhého stavu do čtvrtého a naopak (viz krok 10 v readme na přiloženém CD). Po provedení těchto úprav a vytvoření modelu krátkého ticha zahájíme druhou sérii trénování modelu. Znovu používáme modul HERest, ale v tuto chvíli trénujeme i oba modely ticha. Proto

používáme druhý *master label file*. Ten, který navíc obsahuje mezi jednotlivými slovy krátké pauzy (viz krok 11 v readme na přiloženém CD). Trénování provádíme tak dlouho, dokud dochází ke zlepšování výsledků modelů. Tyto operace je také nutné provést nad každou sadou modelů zvlášť. Po provedení těchto kroků jsou modely připraveny.

5.9 Vlastní detekce klíčových slov

Pokud máme natrénovanou sadu modelů, můžeme přistoupit k vlastní detekci. Nejprve je potřeba připravit některé soubory a provést určité operace.

5.9.1 Příprava

V první řadě si musíme připravit testovací data. Pro každou ze sad testovacích promluv provedeme parametrizaci. K tomuto účelu použijeme modul HCopy. Nejprve si uděláme adresářovou strukturu, ve které budou parametrizovaná data (viz krok 12 v readme na přiloženém CD). Poté potřebujeme seznam, kde bude umístění testovací promluvy a za ní bude následovat umístění parametrizovaného souboru. Parametry budou Mel-frekvenční kepstrální koeficienty (viz krok 13 v readme na přiloženém CD). Výsledkem je sada parametrizovaných testovacích souborů. Vytvoříme seznam s cestami k těmto souborům. Dále bude naše činnost vyžadovat soubor se soupisem klíčových slov. Zvolená klíčová slova jsou uvedena v tabulce 2. Nakonec bude ještě nutné vytvořit specifickou síť, kterou budeme používat při detekování. Tato síť modeluje vlastně situaci na obrázku 4, který je na konci kapitoly čtyři. Soubor se sítí obsahuje na začátku smyčku všech fonémů (část této smyčky, kterou obsahuje soubor se sítí můžeme vidět na obrázku 9). Tím vlastně definujeme náš výplňový model. Za ní následují jednotlivá slova rozepsaná do fonémů, ze kterých se skládají (příklad je na obrázku 10). Na začátku každého řádku je číslo, to představuje stav. Na konci řádku jsou čísla stavů, do kterých lze z aktuálního stavu přejít. Tato síť musí obsahovat všechna klíčová slova, která budeme hledat (síť je tvořena skriptem KWS.pl). Poslední nutností je *master label file*, který bude obsahovat správné umístění slov v jednotlivých promluvách. Jeho podoba bude následující. Na začátku informace o promluvě je cesta k dané promluvě. Následují jednotlivá slova promluvy, která mají vždy dvě číselné informace. Jedná se o hodnoty ve stovkách nanosekund. První představuje začátek daného slova v promluvě. Druhá informuje o konci slova. Část takového souboru můžeme vidět na obrázku:

```
"/homes/eva/xs/xsadec01/tmp/KWS/BLOCK00/SES003/SA003F08.lab"
0 3800000 sil
3800000 14600000 zapiš
14600000 15800000 mi
15800000 16700000 do
16700000 20400000 diáře
20400000 25200000 schůzku
25200000 28500000 zítra
28500000 29300000 ve
29300000 31400000 tři
31400000 35400000 hodiny
35400000 36400000 u
36400000 52900000 šéfa
.
```

obr. 8: Příklad ze souboru pro vyhodnocení správnosti detekovaných slov

Pokud takový soubor nemáme, je možné ho vygenerovat s pomocí nástroje HTK modulem HVite. K tomu budeme potřebovat natrénované modely fonémů, seznam promluv obsahující umístění parametrizovaných souborů, *master label file* se slovy v jednotlivých promluvách a soubor s výpisem jednotlivých modelů. Také je nutné mít slovník s přepisem slov na symboly SAMPA a všech slov obsažených v souboru *master label file*.

```
N=590
0       W=!NULL          3
1       W=!NULL  f=F
2       W=!NULL          1

3       W=!NULL          2 4 5 6 7 8 9 10 11 12 13 14 15
305 317 328 338 346 356 364 375 389 399 408 417 427 434
4       M=e                      3       l=-20.000000
5       M=a                      3       l=-20.000000
6       M=o                      3       l=-20.000000
7       M=u                      3       l=-20.000000
```

obr. 9: Část s fonémovou smyčkou

```
164  W=!NULL  165
165  W=inkoust_B  167
166  W=inkoust  f=K  1

167  M=i  168
168  M=N  169
169  M=g  170
170  M=o_u  171
171  M=s  172
172  M=t  166

173  W=!NULL  174
174  W=jsem_B  176
175  W=jsem  f=K  1

176  M=j  177
177  M=s  178
178  M=e  179
179  M=m  175
```

obr. 10: Část s klířovými slovy

5.9.2 Detekce

V tuto chvíli máme připraveny natrénované modely i všechny náležitosti potřebné k detekci. Vlastní detekci klíčových slov budeme provádět pomocí SLRatio_new. Tento modul vyžaduje natrénované modely, seznam parametrizovaných testovacích promluv a vytvořenou síť pro detekci (viz krok 14 v readme na přiloženém CD). Ve chvíli, kdy proběhne detekce, vznikne výstupní soubor *master label file* (příklad části souboru je na obrázku 11), s informacemi o každé testované promluvě. V první řadě zde bude cesta k testovanému souboru. Za ní následují klíčová slova. Před každým klíčovým slovem jsou dvě hodnoty udávající počátek a konec výskytu ve stovkách nanosekund. Následuje klíčové slovo a za ním je hodnota udávající skóre klíčového slova. Záznam o každé promluvě je ukončen tečkou. Postupně s pomocí SLRatio_new získáme výsledný *master label file* pro každou sadu modelů.

```
38600000 40400000 všechno -535.059
37400000 39800000 vybledlý -768.161
37300000 40300000 vysvětlit -866.345
38000000 40400000 zajímalo -752.164
36700000 40300000 zarezervovat -1087.61
.
"/homes/eva/xs/xsadec01/tmp/KWS/BLOCK22/SES227/SA227S27.rec"
00000000 01200000 ráda -400.281
00000000 01500000 třeba -441.76
```

obr. 11: Příklad výstupu detekce

5.9.3 Získávání výsledků

K získávání výsledků použijeme modul SResults. Na vstupu je vyžadována dvojice souborů *master label file*. První představuje náš výsledek z SLRatio_new. Ve druhém jsou obsažena správná slova v promluvách s časy, kde dané slovo začíná, a kde končí. Poslední věcí nutnou ke správnému fungování je seznam klíčových slov, která jsme hledali (viz krok 15 v readme na přiloženém CD).

Výstupem je pak seznam klíčových slov, kde jsou vypsány u každého slova úspěšné zásahy slova. Další hodnotou je počet falešných poplachů a za ním následuje hodnota FOM. Na konci jsou uvedeny celkové výsledky. Při používání modulu je také možné nastavit práh, při kterém ještě dané slovo považujeme za klíčové. Kvalitu výstupních hodnot navíc můžeme zlepšit nebo zhoršit penalizací jednotlivých modelů nebo slov.

5.10 Výsledky jednotlivých sad modelů

Každou z připravených sad **head_set**, **close_set**, **far_set, comb_set** a **clofar_set** jsme otestovali na trojici sad **head_test**, **close_test** a **far_test**. Souhrnné výsledky jsou v tabulce č. 4 uvedené níže.

5.10.1 Testovací sada head_test

Na této sadě dosahovala největší hodnoty FOM skupina modelů **head_set** a to 59,33%. Modely **close_set** dosáhly výsledku 47,03%. **Comb_set** pak měla hodnotu FOM 57,45%. Sada **clofar_set** dosáhla 35,46%. Nejhorší výsledek měl **far_set** s 10,78%.

Výsledky skupin **head_set** a **close_set** se liší o více než 12%, což je celkem velký rozdíl. Naopak **comb_set** dosáhla oproti **head_set** výsledku o 1,88% horšího. Takže spojením trénovacích dat z náhlavního mikrofonu a mikrofonu blízko před mluvčím jsme výsledek téměř nezhoršili. Skupina **far_set** byla oproti všem ostatním testovaným modelům velmi výrazně horší, ale **clofar_set** byl o 24,75% lepší než **far_set**. Spojením dat z mikrofonu dále od mluvčího a mikrofonu blízko před mluvčím došlo k výraznému zlepšení výsledku na této sadě.

Modely	Testovací sada	FOM	Hits	False alarm
head_set	**head_test**	59,33%	582	1033604
	close_test	52,82%	567	1041460
	far_test	19,39%	504	1151559
close_set	**head_test**	47,03%	570	1019308
	close_test	56,01%	571	1027029
	far_test	31,98%	521	1060717
far_set	**head_test**	10,71%	484	952037
	close_test	39,07%	536	983078
	far_test	40,96%	541	968282
comb_set	**head_test**	57,45%	587	1028777
	close_test	54,85%	571	1030835
	far_test	22,87%	505	1104415
clofar_set	**head_test**	35,46%	574	1006557
	close_test	53,55%	566	1022705
	far_test	34,88%	527	1016674

Tab. 4 – Výsledky jednotlivých testů

5.10.2 Testovací sada close_test

Zde měla nejvyšší úspěšnost skupina **close_set** s 56,01%. Za ní se umístila **comb_set** skupina s 54,85%. Třetí nejlepší výsledek měla sada **clofar_set** s 53,55%. Čtvrté místo pak obsadila **head_set** s 52,82%. Poslední byla skupina **far_set** s 39,07%.

Rozdíl v tomto případě mezi skupinou **head_set** a **close_set** je 3,19%. **Comb_set** pak byla horší oproti skupině **close_set** pouze o 1,16% a kombinováním tedy nedošlo k výraznému zhoršení. **Clofar_set** také nebyla o tolik slabší, než sada **close_set.** Takže spojením trénovacích promluv obou setů nedošlo k výraznému zhoršení. **Far_set** byla o skoro 17% horší než nejlepší skupina modelů.

5.10.3 Testovací sada far_test

Nejúspěšnější byly modely **far_set**, kde jsme dosáhli 40,96%. Sady **close_set** s 31,98%, **comb_set** s 22,87% a **head_set** s 19,39% měli výrazně horší výsledky. Sada **clofar_set** dosáhla 34,88%.

Výsledky na této sadě byly celkem špatné a rozdíly mezi jednotlivými testovanými sadami byly velmi výrazné. Sada **clofar_set** byla o 6,08% horší než **far_set**. Takže zkombinováním promluv došlo k většímu zhoršení.

5.10.4 Test far_set na sadě far_test

Realizace hlasového ovládání je nejlépe vystižena testem na sadě **far_test**. Nejúspěšnější v tomto případě byly modely **far_set**. Výsledky v tomto testu pro jednotlivá klíčová slova jsou v tabulce:

Klíčové slovo	FOM klíčového slova	Klíčové slovo	FOM klíčového slova	Klíčové slovo	FOM klíčového slova
ahoj	30,00%	nádraží	50,00%	příroda	50,00%
automobilu	50,00%	nahoru	75,00%	ráda	47,37%
byla	54,55%	nahrávání	40,00%	rádio	66,67%
července	50,00%	nakonec	60,00%	skutečně	100,00%
další	69,23%	nejoblíbenější	75,00%	sportovec	60,00%
děkuji	10,00%	několik	20,00%	televize	25,00%
dobrý	30,77%	nervózní	83,33%	televizi	23,81%
doprava	44,44%	nikdo	11,11%	třeba	36,84%
dvacátého	25,00%	oblíbený	72,73%	úplně	25,00%
hlavní	27,27%	odpoledne	20,00%	večer	55,56%
hodin	22,22%	otevřely	75,00%	video	45,45%
chtěla	36,36%	počítač	30,00%	vlastně	44,44%
inkoust	0,00%	podle	33,33%	vodovod	0,00%
jsem	44,71%	podzimní	20,00%	všechno	44,44%
jsme	34,48%	pokoj	42,86%	vybledlý	66,67%
který	36,36%	poslední	33,33%	vysvětlit	0,00%
maminka	27,78%	potřebovala	33,33%	zajímalo	0,00%
město	50,00%	program	66,67%	zarezervovat	85,71%
milión	50,00%	prosím	41,18%		

Tab. 5 – Výsledky detekce far_set na far_test pro jednotlivá slova

Při volbě klíčových slov pro ovládání hlasem podle provedeného testu je lepší vybírat delší slova. Pokud jsou si klíčová slova podobná, pak je výsledek horší, než v případě odlišných slov. Příkladem mohou být třeba slova „*televize*“ a „*televizi*“.

5.10.5 Zhodnocení výsledků

Nejlepších výsledku, podle kriteria Figure of Merit, dosahovaly modely **head_set** na testovacím setu nahraného náhlavním mikrofonem. Nejhorších výsledků pak dosáhla sada **far_set** na témže setu.

Podíváme-li se na výsledné hodnoty, kterých dosahovaly jednotlivé skupiny, dojdeme k zjištění, že skupina **close_set** byla výrazně horší v detekci klíčových slov na promluvách získaných

náhlavním mikrofonem. Zato skupina **head_set** si při testu na datech z mikrofonu umístěného blízko před mluvčím nepohoršila o tolik, jako modely v předchozím případě. Modely **comb_set** sice nedosahovaly nejlepších výsledku na ani jedné z testovacích sad **head_test** a **close_test**, ale na druhou stranu se od výsledků dosažených „specializovanými“ modely **close_set** a **head_set** nelišily velmi zásadně. Výsledky sady **far_set** se na sadě **close_test** a **far_test** výrazně nelišily. Sada **clofar_set** si výrazně polepšila oproti sadě **far_set** v hodnocení na **head_test** a zároveň byla také mnohem lepší na **far_test** než sada **head_set**.

Z výsledků, kterých jsme dosáhli, můžeme usoudit, že s narůstající vzdáleností mezi mluvčím a záznamovým zařízením se snižuje kvalita detekce. Pakliže máme testovací a trénovací promluvy zaznamenány z různých vzdáleností, dochází také ke snížení kvality výsledků. Čím větší je rozdíl vzdáleností mezi mikrofony, kterými byla testovací a trénovací sada zachycena, tím horších výsledků dosahujeme. V případech, kdy kombinujeme trénovací promluvy z mikrofonu v různých vzdálenostech, dostáváme modely horší pro obě vzdálenosti, než jsou „specializované“ modely. Na druhou stranu jsou tyto modely daleko univerzálnější.

6 Závěr

V této práci bylo ukázáno, jakým způsobem se používají skryté Markovovy modely k rozpoznávání řeči a k detekci klíčových slov. Byly zde popsány algoritmy, které se využívají k trénování skrytých Markovovských modelů, Baum-Welchovu re-estimaci. Dále jsme byli seznámeni s Viterbiho algoritmem a s jeho možnými úpravy, aby byl využitelný pro rozpoznávání spojité řeči. Metody založené na skrytých Markovových modelech jsou v poslední době prakticky nejrozšířenější a právě z tohoto důvodu byly zvoleny. Následně jsme se zaměřili na detekci slov a představili si experiment, kterým se tato práce zabývala. Navrhli jsme způsob provedení experimentu, podnikli potřebné kroky a experiment uskutečnili. Vytvořili jsme pětici rozpoznávačů, které byly založeny na fonémech. Každý z nich byl natrénován na specifickém trénovacím setu. Všechny jsme použili k detekci klíčových slov na třech různých sadách testovacích dat a ukázalo se, že nejlepších výsledků dosáhneme v případě, kdy trénovací a testovací sada je zachycena mikrofonem umístěným v podobné vzdálenosti. V našem případě tedy data zachycená pomocí náhlavního mikrofonu dávala nejlepší výsledky při detekci na datech zachycených zase pomocí náhlavního mikrofonu. Stejně tomu tak bylo i pro ostatní případy, kdy platilo, že promluvy ze stejné vzdálenosti byly nejlépe rozpoznávány. Pakliže trénovací data pocházela z náhlavního mikrofonu a detekci jsme prováděli na datech z mikrofonu poblíž mluvčího a naopak, úspěšnost detekce klíčových slov poklesla. To samé platilo i pro promluvy z mikrofonu dále od mluvčího. V případě, že jsme použili k natrénování data ze dvou zdrojů, vznikly modely, které nedosahovaly tak kvalitních výsledků, jako v případě použití dat pouze z jednoho zdroje. Tyto modely však byly univerzálnější. V testovací sadě pocházející z mikrofonu dále od mluvčího měly modely největší problémy. Nejlépe se s tímto testem vypořádala sada trénovaná na mikrofonu v podobné vzdálenosti. Ostatní modely na tom byly výrazně hůře. Čím větší byla vzdálenost mezi mluvčím a mikrofonem, tím horší byly výsledky. Všeobecně se také při narůstajícím rozdílu vzdáleností dat zachycených pro trénování a testování výsledky výrazně zhoršovaly. Pokud tedy víme, že budeme detekovat slova ve větší vzdálenosti, bude pro nás lepší promluvy z bezprostřední blízkosti nepoužívat. V případech, kdy nám je známo, že data budou získávána například pouze náhlavním mikrofonem, použijeme k natrénování modelů nahrávky z náhlavního mikrofonu. Každopádně je při přípravě modelů nejlepší, je-li vzdálenost mluvčího od mikrofonu co nejlépe specifikována. Co se týče volby klíčových slov, je mírně lepší volit slova delší a rozdílná.

Pokud bychom zkoumali problematiku, kterou se tato práce zabývá dále, mohli bychom poskládat další trénovací sady promluv. Například bychom mohli přidat mikrofon v jiné vzdálenosti přehráním a znovu zaznamenáním trénovacích dat. Také by bylo možné vytvořit sadu modelů

z promluv všech mikrofonů. Dále bychom mohli přejít od modelů fonémů k tvorbě modelů kontextově závislých fonémů. Bylo by také možné zvětšit počet parametrů, kterými popisujeme promluvy.

Rozpoznávání řeči a detekce klíčových slov je v dnešní době velmi rychle se rozvíjející obor. Schopnost člověka komunikovat se strojem ve svém rodném jazyce, a navíc přirozeným způsobem, je velice lákavá. Přibylo by nám zde totiž další rozhraní, pro komunikaci se strojem. Rozhraní, které by bylo snadno použitelné i v případě, že ovládající má plné ruce anebo potřebuje svoji pozornost rozdělit mezi několik činností. Až se podaří dostatečně zlepšit kvalitu detekce a rozpoznávání získáme možnost ovládat stroj na větší vzdálenost, než je dosah našich rukou. To je pouze hudba budoucnosti. V dnešní době jsme totiž zatím ještě na začátku, ale první krůčky jsme již učinili

Literatura

[1] S. Young, G. Evermann, D. Kershaw, G. Moore, G. J.Odell, D. Ollason, V. Valtchev, P. Woodland: #The HTK Book#, Cambridge University Engineering Department, 2006 dostupný z WWW: <http://htk.eng.cam.ac.uk/prot-docs/htkbook.pdf>.

[2] J. Černocký: Zpracování řečových signálů — studijní opora, Brno, Ústav počítačové grafiky a multimédií, FIT VUT v Brně, 2006 dostupný z WWW: <http://www.fit.vutbr.cz/study/courses/ZRE/public/>.

[3] ZRE: Hmm s HTK, cvičení, Brno, FIT VUT v Brně, 2009 dostupný z WWW: <http://www.fit.vutbr.cz/study/courses/ZRE/public/>.

[4] SRE: Demo rozpoznávač, Brno, FIT VUT v Brně, 2006 dostupný z WWW: <http://www.fit.vutbr.cz/study/courses/SRE/public/>.

[5] L. Šmídl a J. Zahradil: METODY DETEKCE KLÍČOVÝCH SLOV V PROUDU ŘEČI, Katedra kybernetiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni, 2007 dostupný z WWW: <http://ui.zcu.cz/img/p_248.pdf>.

[6] V. Hubeika: Speaker Independent Estimation of Gender and Age, bakalářská práce, Brno, FIT VUT v Brně, 2006.

[7] Skryté Markovovy modely, Brno, Masarykova Universita v Brně, 2007 dostupný z WWW: <http://nlp.fi.muni.cz/nlp/nlp-prace/referaty/xkrivan/HMM.html>

[8] Petr Pollák, Jan Černocký: Czech Speecon Adult Database. FEL ČVUT v Praze a FIT VUT v Brně. 2004.

[9] SAMPA for Czech [online]. 2003 [cit. 2010-07-01]. Czech SAMPA. Dostupné z WWW: <http://noel.feld.cvut.cz/sampa/>.

[10] Igor Szöke: Concept of Doctoral Thesis, Brno, Faculty of Information technology, 2005

Seznam příloh

Příloha 1. CD/DVD

Obsah přiloženého CD

- CD
  - modely – *obsahuje natrénované modely* ***head_set, close_set, far_set, comb_set*** *a* ***clofar_set***
  - slozka
    - cfg – *obsahuje konfigurační soubory pro parametrizaci a trénovaní*
    - dics – *obsahuje slovník a seznam fonémů*
    - mlf – *obsahuje master label file soubory*
    - net – *obsahuje síť pro detekci klíčových slov*
    - pomocne – *obsahuje pomocné soubory a skripty*
    - proto – *obsahuje prototyp modelu*
    - test – *obsahuje seznamy dat pro jednotlivé testovací sady*
    - training – *obsahuje seznamy dat pro jednotlivé trénovací sady*
  - technicka zprava – *obsahuje technickou zprávu ve formátech pdf a odt*
  - vysledky
    - mlf – *Obsahuje výstupní master label file soubory z SLRatio_new*
    - vystupy – *Obsahuje výstupy z SResults*