Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň 2012

Petr Řezáček

Prohlášení

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

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

Plzeň 18. května 2012 .....

podpis

Anotace

Tématem této bakalářské práce je podrobné vyhodnocování kvality systémů rozpoznávání řeči. Práce obsahuje nejprve stručný přehled vývoje metod rozpoznávání řeči. V další části jsou popsány použité postupy při vyhodnocování (Levenshteinova metoda, intervaly spolehlivosti, konfúzní tabulka) a zkoumané jevy (kvalita rozpoznávání slov, písmen, různých typů souhlásek apod.). V poslední části práce jsou popsané a okomentované dosažené výsledky pro testovací sady referenčních a rozpoznaných promluv. Vytvořený skript je přiložen na CD a jeho uživatelská i programátorská dokumentace je v dodatku.

Klíčová slova: rozpoznávání souvislé řeči, vyhodnocování rozpoznávání řeči, HTK, Levenshteinova metoda, intervaly spolehlivosti, konfúzní tabulka

Abstract

The subject of this bachelor thesis is detailed evaluation of speech recognition systems quality. The thesis contains a brief overview of methods of speech recognition. Levenshtein method, the confusion table algorithm and the confidence intervals algorithm are explained and used for evaluation of chars, vowels, consonants etc. The results are presented and analysed at the conclusion of the thesis. Scripts for evaluation are included and documented on the CD, user manual is in appendix.

Key words: speech recognition, speech recognition evaluation, HTK, Levenshtein method, confidence intervals, confusion table

Obsah

1 Úvod - komunikace člověk-stroj **1**

2 Cíle práce **3**

3 Rozpoznávání řeči **4**

3.1 Historie rozpoznávání řeči 4

3.2 Hlavní problémy rozpoznávání řeči 5

3.3 Metody rozpoznávání řeči 6

3.3.1 Porovnávání obrazů 6

3.3.2 Statistický přístup 7

3.3.3 Znalostní přístup 8

4 HTK **10**

5 Vyhodnocování kvality rozpoznávání řeči **11**

5.1 Porovnávání rozpoznaných a referenčních promluv 12

5.2 Levenshteinova metoda 14

5.2.1 Výpočet levenshteinovy vzdálenosti 14

5.3 Intervaly spolehlivosti 15

5.3.1 Interval spolehlivosti ve statistice 15

5.3.2 Použití intervalu spolehlivosti pro vyhodnocování rozpoznávání řeči . . 15

5.3.3 Postup pro výpočet intervalu spolehlivosti 18

5.3.4 Vlastní implementace algoritmu 20

5.3.5 Výsledky 20

5.4 Konfúzní tabulka 20

5.4.1 Vytváření konfúzní tabulky . . . 21
5.4.2 Využití konfúzní tabulky . . . 21
5.5 Vyhodnocení pro clustery . . . 22
5.5.1 Postup vyhodnocení pro clustery . . . 23
5.6 Hromadné vyhodnocování . . . 23
5.7 Chybovost na začátku a konci slov a promluv . . . 23
5.8 Záměna sekvencí slov . . . 24

6 Výsledky **25**
6.1 Konzolový výstup . . . 25
6.2 HTML soubor . . . 26
6.2.1 Rozbor výsledků . . . 26
6.3 Graf porovnání délek slov . . . 29
6.4 Záměna sekvencí slov . . . 30
6.5 Konfúzní tabulky . . . 30
6.6 Časté záměny písmen . . . 31
6.7 Časté záměny slov . . . 31

7 Závěr **32**

A Programová dokumentace **1**
A.1 Spouštění skriptu . . . 1
A.1.1 Vypsání nápovědy . . . 1
A.1.2 Příklad spouštění skriptu pro vyhodnocení všech vět . . . 2
A.1.3 Příklad spouštění skriptu pro vyhodnocení vět odpovídajících požadovaným clusterům . . . 2
A.1.4 Výstupní soubory . . . 2

Seznam obrázků

1.1 Komunikace člověk-stroj 1

5.1 Postupné porovnávání řetězců 13
5.2 Postupné porovnávání řetězců 13
5.3 95% interval spolehlivosti, normální rozdělení (převzato z [9]) 16
5.4 Ukázka souboru s clustery 22

6.1 Ukázka výstupu - HTML soubor (1. část) 27
6.2 Ukázka výstupu - HTML soubor (2. část) 28
6.3 Porovnání délek slov 29

Seznam tabulek

5.1 Výpočet levenshteinovy vzdálenosti, počáteční stav, převzato z [4] 14
5.2 Vypočtená levenshteinova vzdálenost, konečný stav, převzato z [4] 15
5.3 Některé kvantily normovaného normálního rozdělení používané pro intervaly spolehlivosti 19
5.4 Konfúzní tabulka 21

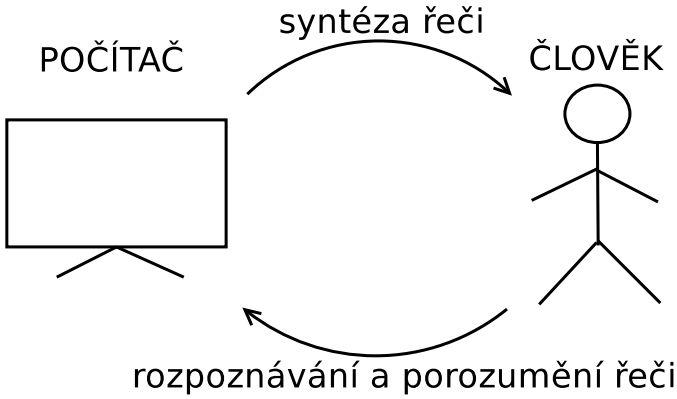
6.1 Ukázka vygenerované konfúzní tabulky 30

Kapitola 1

Úvod - komunikace člověk-stroj

Komunikaci můžeme rozdělit na *komunikaci mezi lidmi*, *komunikaci mezi stroji* a *komunikaci mezi člověkem a strojem*. Komunikace mezi lidmi je už dávno prostudované a popsané téma. Komunikace mezi stroji už delší dobu funguje na vysoké úrovni, ačkoliv se stále snažíme ji zdokonalovat. V současné době však v popředí zájmu stojí komunikace člověk-stroj (většinou počítač). Tato oblast by zasloužila vylepšení, která by komunikaci zjednodušila a zrychlila. Navíc by pak tato vylepšení mohla přispět k tomu, že by se možnost komunikace s počítačem zpřístupnila i lidem, kteří prozatím nemají možnost ji používat. Jedná se např. o slabozraké. Nejlepší možností pro člověka se ukázala komunikace mluvenou řečí.

Úlohu komunikace člověk-počítač lze rozdělit na dílčí úlohy - syntézu řeči, rozpoznávání řeči a porozumění řeči (viz. obr. 1.1).



Obr. 1.1: Komunikace člověk-stroj

Syntéza řeči znamená převod textu do mluvené podoby. Obecná syntéza řeší úlohu, jak

převézt libovolnou textovou předlohu do mluveného slova tak, aby byla vzniklá promluva srozumitelná a přirozená, jako lidská řeč. Ačkoliv v dnešní době poskytuje uspokojivé výsledky, stále není dokonalá. Přesto však stačí na vytváření zvukových knih pro nevidomé, čtení SMS zpráv a předčítání emailů. Větší přirozenosti lze dosáhnout u syntézy řeči z limitované (omezené) oblasti. Takovým systémem může být např. telefonní automat nebo mluvící hodinky.

Rozpoznávání řeči (angl. *speech recognition*), kterému je věnována tato práce, je protikladem syntézy - jedná se o převod mluvené řeči do textové podoby. Nachází uplatnění při slovní komunikaci s počítačem, ale také např. při zapisování diktátu počítači nebo při automatickém titulkování.

Porozumění řeči zajišťuje pochopení rozpoznané posloupnosti slov. To je důležité pro to, aby dokázal stroj splnit zadaný příkaz, vykonat požadovanou akci.

Rozhraní mezi počítačovou aplikací a uživatelem komunikujícím hlasem, které obsahuje všechny tyto tři moduly a modul pro generování odezvy, se nazývá *hlasový dialogový systém*. Pro správnou činnost všech takových systémů je především velmi důležité, aby již výše zmíněné rozpoznávání řeči nebylo zatíženo velkou chybou. Proto se tato práce věnuje výsledkům rozpoznávání řeči a způsobům *vyhodnocování* této úlohy.

Kapitola 2

Cíle práce

Hlavním tématem této práce je počítačové rozpoznávání řeči. V následující kapitole (3) je tato problematika, spadající do umělé inteligence, popsána.

Práce je zaměřená na vyhodnocování výsledků rozpoznávání, čemuž je věnována kapitola 5.

Mým úkolem bylo rozšířit dodaný skript a zajistit jeho kompatibilitu s referenčním systémem HTK [6]. Kromě vyhodnocování kvality rozpoznávání na úrovni vět, slov a znaků jsem měl naprogramovat i vyhodnocování krátkých a dlouhých slov, samohlásek a souhlásek a jejich typů, zjistit chybovost na začátku a na konci slov/promluv a vypsat časté chyby rozpoznávání.

Kromě percentuálního vyjádření jsem měl jednotlivé statistiky popsati *intervaly spolehlivosti*, které lépe vyjadřují kvalitu rozpoznávacího systému a lze je lépe využít pro porovnávání různých rozpoznávacích systémů či různých nastavení jednoho systému.

Statistické výstupy tohoto skriptu by měli sloužit ke zkoumání častých chyb a jejich příčin a k přenastavení parametrů systému rozpoznávání pro získání lepších výsledků.

Mělo by být také umožněno dávkové spouštění skriptu pro hromadné vyhodnocení výsledků, skript by tedy neměl vyžadovat žádnou interakci uživatele.

Ověření správné funkčnosti vytvořeného programu a analýza výsledků je pak v kapitole 6.

Kapitola 3

Rozpoznávání řeči

Termínem rozpoznávání řeči rozumíme převod mluvené řeči do textu. Jak už bylo napsáno v úvodu, uplatní se zejména při automatickém zapisování diktátu (např. pořizování záznamu ze soudního řízení), ovládání počítače či přístrojů pro nevidomé hlasem (v těchto případech musí být systém rozpoznávání řeči rozšířen i o porozumění řeči, aby počítač rozuměl, co po něm chceme). Výhodné je i již zmíněné automatické titulkování, což umožňuje jednak neslyšícím sledování pořadů v televizi, tak pořizování záznamu pořadu v textové podobě. V současné době lze např. sledovat záznamy ze zasedání Poslanecké sněmovny, které jsou automaticky titulkované (projekt Západočeské univerzity v Plzni).

3.1 Historie rozpoznávání řeči

Rozvoj rozpoznávání, stejně jako analýzy a syntézy řeči, úzce souvisí s rozvojem výpočetní techniky. První pokusy o rozpoznávání řeči se objevují na začátku 20. století. Příkladem je třeba hračka Radio Rex, využívající poznatků z oblasti zpracování řečového signálu, která se objevila na trhu v roce 1920. Hračka byla založena na využití frekvence 500Hz, která je obsažena ve slově Rex. Při vyslovení tohoto slova byl přerušen elektrický obvod a díky pružině pes vyskočil z boudy. Je zřejmé, že hračka reagovala i na jiná slova, jež měla podobnou charakteristiku.

Ve 30. letech byl v Bellových laboratořích[1] vyvinut přístroj, který dokázal klasifikovat

[1] vývojová organizace provádějící výzkum a vývoj zařízení po celém světě, zal. 1925, ústředí v New Jersey v USA

jednotlivé číslovky. Po natrénování systému na určitého řečníka dokázal systém na základě vysloveného slova určit číslo, o které se jedná, s chybou 2%.

V 50. letech vytvořil svůj klasifikátor i Homer W. Dudley, americký vědec zabívající se syntézou i rozpoznáváním řeči. Na konci 50. let byla při klasifikaci kromě akustické informace použita i informace a pravděpodobnosti gramatiky, tj. pravděpodobnost výskytu lingvistické jednotky závisí také na předešlé lingvistické jednotce.

Během šedesátých a sedmdesátých let došlo k rozvoji Fourierovy transformace a začala se také využívat technika *dynamického programování* (viz. 3.3.1). Klasifikace s využitím dynamického programování byla v sedmdesátých letech nejčastějším přístupem k rozpoznávání izolovaných slov.

V osmdesátých letech se do popředí zájmu dostala technika klasifikace řeči na základě statistického přístupu ke zpracování řečového signálu [8]. Využívá se zde způsob modelování řeči použitím tzv. Markovových modelů, které modelují krátší řečové jednotky, jako např. fonémy, trifóny. Tato metoda je vhodná pro rozpoznávání souvislé řeči a je používaná až do dnešní doby, protože poskytuje nejlepší výsledky.

Ačkoliv se zdá, že při takto rychlém vývoji je sestavení systému na rozpoznávání mluvené řeči libovolného člověka na libovolné téma otázkou několika let, není tomu tak. Existuje totiž několik problémů, které rozpoznávání řeči ztěžují.

3.2 Hlavní problémy rozpoznávání řeči

Prvním problémem rezpoznávání řeči je, že každý člověk má jinou barvu hlasu a přízvuk, hovoří odlišným tempem. Je to způsobeno odlišnými parametry hlasového ústrojí a odlišným způsobem artikulace. Podle závislosti na řečníkovi rozdělujeme systémy rozpoznávání řeči na řečníku závislé a nezávislé. Systémy na řečníku závislé jsou natrénované na hlas jednoho konkrétního řečníka, jehož mluva je po natrénování rozpoznávána s malou chybou. Mnohem větší chyba však nastane při použití tohoto klasifikátoru na jiné osoby. Trénování systémů na řečníku nezávislých je obtížnější, neboť jsou při něm použity hlasy mnoha (stovek až tisíců) řečníků.

Dalším faktorem, který rozpoznávání ztěžuje je odlišnost hlasu jednoho konkrétního člověka v různých situacích. Řečový signál závisí na hlasitosti promluvy (šeptání, hlasitá mluva), na náladě řečníka (radost, smutek, rozčilení) i na případné hlasové indispozici. Je téměř ne-

možné, aby člověk vyslovil určité slovo v různých situacích naprosto stejně. Řečové signály se liší nejen svojí délkou, ale také poměrnou délkou jednotlivých fonémů (hlásek).

U rozpoznávání souvislé řeči je problémem i *koartikulace* (z lat.) [2], což je vzájemné ovlivňování artikulačních pohybů, ke kterému dochází při výslovnosti sousedních hlásek. Dochází tedy k pozměnění fonetických vlastností začátků a konců slov v závislosti na kontextu slov okolních.

U klasifikátorů, které budou pracovat v rušném prostředí, je důležité, aby systém dokázal odlišit samotnou mluvu od měnícího se akustického pozadí (např. při telefonování z jedoucího automobilu). Hlasitější okolní šum způsobuje problémy při rozpoznávání začátků a konců slov a některých souhlásek (hlavně sykavek, např. s, š, z, ž).

Velký vliv na správnou funkci systému má způsob, jakým je řeč promlouvána - zda se jedná o řeč čtenou nebo o spontánní. U spontánně pronášené promluvy je řeč často mnohem rychlejší a více hlásek je vysloveno špatně. Navíc taková promluva často obsahuje i více „neřečových událostí“ (nádechy, váhání atd.), nebo se řečník v půlce slova zarazí a slovo buď zopakuje, nebo pokračuje slovem úplně jiným. Problémem českého jazyka jsou také hovorové výrazy a nespisovné vazby.

Funkce systému pro rozpoznávání závisí také na složitosti úlohy. Rozlišujeme rozpoznávání izolovaných slov (např. malý slovník povelů pro ovládání přístroje pro nevidomé), rozpoznávání diskrétního diktátu (mezi slovy je krátká pauza, aby nedocházelo ke vzájemné koartikulaci) a rozpoznávání souvislé řeči.

3.3 Metody rozpoznávání řeči

K rozpoznávání řeči můžeme přistupovat třemi různými způsoby. První způsob, porovnávání se vzory, je vhodný k rozpoznávání izolovaných slov. K rozpoznávání souvislé řeči se dnes používá statistický nebo znalostní přístup.

3.3.1 Porovnávání obrazů

Metoda porovnávání obrazů (angl. *template matching*) zpracovává slovo jako celek a zařazuje ho do třídy, jejíž vzorový obraz je nejblíže rozpoznávanému slovu. Problémem porovnávání je ale nejstejná délka. Stejné slovo namluvené několikrát (dokonce i stejným řečníkem)

má vždy jinou délku.

První možností, jak se s tímto problémem vyrovnat, je lineární časová normalizace, tj. zkrácení nebo natažení promluv na stejnou konstantní délku I. Když ale porovnáme 2 nahrávky stejného slova, zjistíme, že kromě odlišné délky se liší i poměrem délek vyslovování jednotlivých hlásek. Časová normalizace na jednotnou délku nemůže toto nelineární časové kolísání uvnitř slova postihnout.

Z důvodů uvedených výše se místo časové normalizace využívá výhod *dynamického programování* (*DTW algoritmus*), které respektuje kolísání v časové oblasti. Časové rozdíly se modelují časově nelineární „bortivou“ funkcí. Algoritmus *dynamického programování* hledá posloupnost bodů, která minimalizuje celkovou vzdálenost porovnávaných promluv. Takto vypočtená vzdálenost se ještě normuje.

Metoda rozpoznávání porovnáváním obrazů se používá (používala) pro klasifikaci izolovaných slov z malých slovníků. Trénováním klasifikátoru, používajícím tuto metodu, rozumíme nahrání dostatečného množství promluv, se kterými budeme rozpoznávané slovo porovnávat. Při klasifikaci neznámého slova pak využijeme *DTW algoritmus* a hledáme takové slovo ze slovníku, které má od rozpoznávaného slova minimální vzdálenost.

3.3.2 Statistický přístup

Statistický přístup byl vyvíjen od roku 1975 a slouží k rozpoznávání spojité řeči. Jedním z autorů této metody je i Bedřich Jelínek, český emigrant do USA.

Na systém rozpoznávání se díváme jako na akustický procesor spojený s lingvistickým dekodérem. *Akustický procesor* převádí řečový signál na posloupnost značek - vektorů příznaků (většinou 1 vektor na každých 10 ms) a *lingvistický dekodér* převádí tuto posloupnost na řetězec slov. Rozpoznávání chápeme jako dekódování s maximální pravděpodobností.

Označme $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w_N}\}$ jako posloupnost slov a $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots \mathbf{o}_N\}$ jako posloupnost značek (vektorů příznaků). Cílem metody je nalézt posloupnost slov $\hat{\mathbf{W}}$, která maximalizuje pravděpodobnost $P(\mathbf{W}|\mathbf{O})$ (tj. pravděpodobnost, že byla vyřčena posloupnost slov $\mathbf{W}$, je-li řečový signál převeden na posloupnost značek $\mathbf{O}$).

Při odvození základního vztahu pro podmíněnou pravděpodobnost využijeme *Bayesův vztah* [7]:

$$P(\hat{\mathbf{W}}|\mathbf{O}) = \max_{\mathbf{W}} P(\mathbf{W}|\mathbf{O}) = \max_{\mathbf{W}} \frac{P(\mathbf{W}) \cdot P(\mathbf{O}|\mathbf{W})}{P(\mathbf{O})} = \max_{\mathbf{W}} (P(\mathbf{W}) \cdot P(\mathbf{O}|\mathbf{W}))$$

kde $P(\mathbf{W})$ je pravděpodobnost posloupnosti **W** slov ze slovníku a $P(\mathbf{O}|\mathbf{W})$ je pravděpodobnost, že bude řečový signál promluvy převeden na posloupnost značek **O**, známe-li posloupnost slov **W**. Pravděpodobnost dané posloupnosti značek $P(\mathbf{O})$ jsme mohli z maxima vypustit, jelikož nezávisí na **W**.

Jazykový model

Jazykový (statistický) model slouží k určení pravděpodobnosti $P(\mathbf{W})$. Jedná se vlastně o pravděpodobnost vyřčení určité posloupnosti slov v daném jazyce a získává se zpracováním rozsáhlých textů. Pravděpodobnost každého slova oceníme na základě jeho levého kontextu. Protože by však bylo velmi složité určit pravděpodobnost slova na základě všech jeho předchůdců, využívá se *n-gramový model* většinou *trigramový*. Řešíme tedy jen pravděpodobnost výskytu trojic slov, kterou aproximujeme relativní četností příslušné trojice v rozsáhlých textových korpusech. Při určování pravděpodobností u všech různých *trigramů* se ale musí určitá malá nenulová pravděpodobnost přiřadit i posloupnostem, které se v trénovacích textech nevyskytují. Jinak by totiž takováto trojice slov nebyla nikdy rozpoznána.

Akustický model

Akustický model, nebo též *model řečníka*, slouží k určení $P(\mathbf{O}|\mathbf{W})$, resp. $P(\mathbf{O}|\lambda_W)$.

Používá se zde modelování řeči *skrytými Markovskými modely*. Předpokládá se, že v určitém časovém okamžiku se hlasové ústrojí nachází v jednom z konečného počtu stavů artikulačních konfigurací a vzniká při tom určitý akustický signál. Vhodné se ukázalo modelovat např. trifóny 3-stavovým Markovovým modelem.

Pravděpodobnost $P(\mathbf{O}|\lambda_W)$ tedy chápeme jako pravděpodobnost, že výstupní posloupnost značek **O** bylo vygenerována Markovovým modelem λ_W, který odpovídá posloupnosti slov **W**.

3.3.3 Znalostní přístup

Znalostní systémy využívají dekompozice úlohy na dílčí podúlohy, na jejichž řešení se může podílet více nezávislých odborníků (*expertů*). *Báze znalostí* je vytvářena dlouhodobým experimentováním a klasifikací mluvené řeči. V systémech rozpoznávání se využívají následující typy znalostí:

Akusticko-fonetické znalosti

Slouží k převodu nasegmentovaného řečového signálu na fonetický přepis promluvy. Tento přepis však nikdy není jednoznačný, pro každý úsek promluvy existuje více alternativ fonetických jednotek s různou pravděpodobností (tzv. *fonetická mříž*).

Lexikální a fonologické znalosti

Zde se využívá pravidel pro fonetickou transkripci k transformaci fonetické mříže do přijatelné posloupnosti slov.

Syntaktické znalosti

Tyto znalosti bývají reprezentovány vhodnou gramatikou a vyjadřují přípustnou konstrukci klasifikovaných vět.

Sémantické znalosti

Slouží k ocenění smyslového obsahu posloupnosti slov. Pro reprezentaci těchto vlastností se používají *rámce* a *sémantické sítě*.

Pragmatické znalosti

Analyzují a oceňují kontext promluvy, pro reprezentaci se využívají *scénáře*.

Kapitola 4

HTK

Balík HTK [6], který je vyvíjen na univerzitě v Cambridge slouží k rozpoznávání řeči. Princip rozpoznávání je založen reprezentaci *skrytým Markovovým modelem* [11] (angl. *Hidden Markov Model*, HMM).

Součástí balíku je i program `HResult.exe`, který slouží k vyhodnocování kvality rozpoznávání, tj. porovnává rozpoznané promluvy s promluvami referenčními. Program vyhodnocuje úspěšnost rozpoznání na úrovni slov a promluv.

Ukázka výstupu vyhodnocování tří vět je uveden níže:

```
------------------------ Sentence Scores ----------------------------
====================== HTK Results Analysis =======================
  Date: Fri May 18 11:50:22 2012
  Ref : words.mlf
  Rec : vysledek_all_1.txt
------------------------- File Results ------------------------------
text89_53.rec:  100.00(100.00)  [H=   1, D=  0, S=  0, I=  0, N=  1]
text92_46.rec:  100.00(100.00)  [H=   1, D=  0, S=  0, I=  0, N=  1]
text92_47.rec:    0.00(  0.00)  [H=   0, D=  0, S=  1, I=  0, N=  1]
------------------------ Overall Results ---------------------------
SENT: %Correct=66.67 [H=2, S=1, N=3]
WORD: %Corr=66.67, Acc=66.67 [H=2, D=0, S=1, I=0, N=3]
====================================================================
```

Kapitola 5

Vyhodnocování kvality rozpoznávání řeči

Vyhodnocování kvality systémů rozpoznávání řeči spočívá v porovnávání množiny rozpoznaných posloupností slov s množinou referenčních posloupností slov (tj. tím, co bylo ve skutečnosti řečeno). Zajímá nás především úspěšnost rozpoznávání na úrovni promluv, slov a hlásek. Pro podrobnější zkoumání je ale také vhodné vyhodnotit podrobnější statistiky. Celkový výčet zkoumaných rysů je uveden v následujícím přehledu.

Zkoumané rysy

1. úspěšnost rozpoznávání celé promluvy, tj. jaké procento promluv bylo rozpoznáno bezchybně
2. úspěšnost rozpoznávání slov
   - všechna slova
   - dlouhá slova
   - krátká slova (délky max 3) - zde předpokládám větší chybovost, než u dlouhých slov
3. úspěšnost rozpoznávání hlásek
   - všechny hlásky

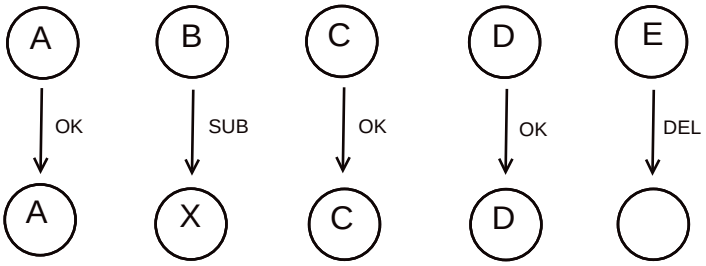
- samohlásky
- souhlásky - pro zjištění problematických skupin dále rozdělujeme na
  - okluzivy: `p, b, t, d, ť, ď, k, g`
  - nazály: `m, n, ň`
  - frikativy: `f, v, s, z, š, ž, ch, h`
  - semiokluzivy: `c, č`
  - vibranty: `r, ř` aproximanty: `l, j`

4. Chybovost na začátku a na konci
   - vět
   - slov
5. vypsání sekvencí slov, u kterých došlo k záměně, např. `okolo` → `o kolo`, `že na` → `žena`
6. vygenerovat konfúzní tabulku
   - pro slova
   - pro hlásky
7. vypsání záměn
   - slov
   - hlásek
8. vyhodnocování úspěšnosti rozpoznávání pro určité *clustery*
9. porovnání délek slov v referenčních a rozpoznaných větách

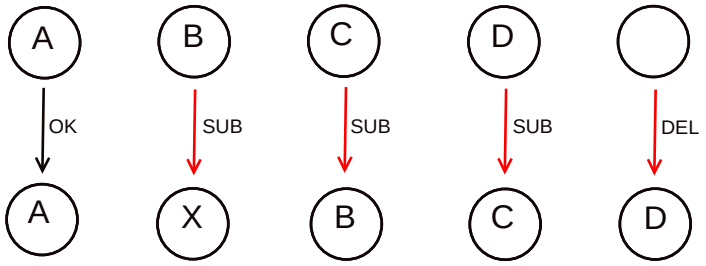
5.1 Porovnávání rozpoznaných a referenčních promluv

K při určování úspěšnosti rozpoznávání je třeba porovnávat rozpoznaný řetězec s referenčním řetězcem.

Definujme 3 základní operace, které lze se znaky provést:



Obr. 5.1: Postupné porovnávání řetězců



Obr. 5.2: Postupné porovnávání řetězců

- *deletion* (D) - smazání jednoho znaku
- *insertion* (I) - vložení jednoho znaku
- *substitution* (S) - náhrada jednoho znaku za jiný znak

Ukázka operací je na obr. 5.1, kde je znázorněné postupné porovnávání 2 slov. Pokud bychom S, I, i D penalizovali hodnotou 1, byla by vzdálenost slov (tj. počet chyb) v tomto případě rovna 2.

Pokud bychom porovnávání prováděli postupně (např. ve `for` cyklu), jedna jediná chyba uprostřed řetězce (rozpoznání hlásky navíc nebo naopak vynechání jedné hlásky) by nám ale mohla nakumulovala vysokou chybovost, jak je znázorněno na obr. 5.1. V tomto případě je počet chyb roven 4, přestože nás snadno napadne, že se slova liší pouze jedním vloženým písmenem `X`. Z tohoto důvodu se porovnání řetězců provádí jiným způsobem - *levenshteinovou metodou*.

5.2 Levenshteinova metoda

Levenshteinova[1] vzdálenost (angl. *Levenshtein distance*) dvou řetězců je definovaná jako minimální počet operací mazání (D), vkládání (I) a substituce (S) takových, aby po jejich provedení byly zadané řetězce totožné.

Levenshteinova metoda je založena na procházení a vyplňování matice, kde sloupce odpovídají znakům řetězce **A** a řádky znakům řetězce **B** (viz obr.5.1).

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1								
u	2								
n	3								
d	4								
a	5								
y	6								

Tab. 5.1: Výpočet levenshteinovy vzdálenosti, počáteční stav, převzato z [4]

5.2.1 Výpočet levenshteinovy vzdálenosti

Procházíme postupně matici a pro každou pozici *i,j*, kde *i* je číslo řádku a *j* je číslo sloupce, provedeme následující:

$$\begin{aligned} matrix[i][j] = min((matrix[i-1][j] &+ cost\_DEL), \\ (matrix[i][j-1] &+ cost\_INS), \\ (matrix[i-1][j-1] &+ cost\_SUB)) \end{aligned}$$

kde $cost\_SUB = 0$, pokud $\mathbf{A}[i] = \mathbf{B}[j]$.

Tento výpočet je znázorněn v tabulce 5.2. Výsledná levenshteinova vzdálenost řetězců je rovna hodnotě $matrix[-1][-1]$.

[1] Vladimir Levenshtein, 1965

Kromě vypočtené minimální vzdálenosti lze levenshteinovou metodou určit i řetězec operací, které musí být aplikovány na řetězec **A**, aby byl tento převeden na řetězec **B**.

Pro uvedený příklad je hodnota levenshteinovy vzdálenosti rovna 3 a řetězec operací `ops = oiiosooo`.

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	**3**

Tab. 5.2: Vypočtená levenshteinova vzdálenost, konečný stav, převzato z [4]

5.3 Intervaly spolehlivosti

5.3.1 Interval spolehlivosti ve statistice

Interval spolehlivosti (a, b) je odhadem parametru θ rozdělení náhodné veličiny X takovým, že platí vztah

$$P(a < \theta < b) = 1 - \alpha \ ,$$

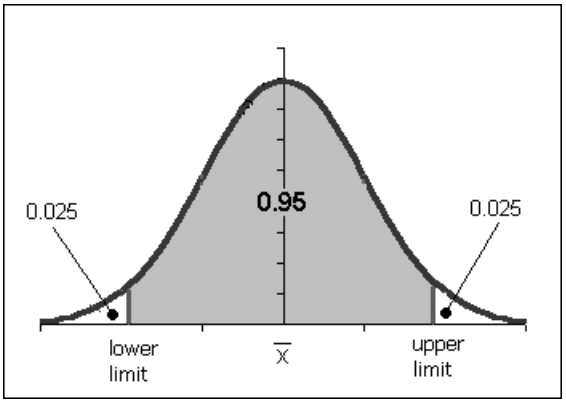
kde $\alpha \in (0, 1)$ ()často $\alpha = 0.01$, 0.05 nebo 0.10), a - dolní mez spolehlivosti, b - horní mez spolehlivosti.

Interval (a, b) se nazývá $100 \cdot (1 - \alpha)\%$-ní oboustranný interval spolehlivosti.

Na obr. 5.3.1 je zobrazen 95% interval spolehlivosti pro střední hodnotu veličiny s normálním rozdělením.

5.3.2 Použití intervalu spolehlivosti pro vyhodnocování rozpoznávání řeči

Vyhodnocování rozpoznávání řeči je založeno na porovnávání rozpoznaných vět s referenčními větami. Počet chyb v každé větě se určuje jako normovaná Levenshteinova vzdálenost



Obr. 5.3: 95% interval spolehlivosti, normální rozdělení (převzato z [9])

vět (vydělená délkou věty) a je označován Word Error Rate (WER).

Budeme tedy určovat interval spolehlivosti pro *WER*.

Předpokládejme, že vyhodnocujeme rozpoznávání s vět. Délku věty i označíme n_i, počet chyb označíme e_i.

Hodnotu *WER* tedy můžeme vyjádřit jako

$$w = \frac{\sum_{i=1}^{s} e_i}{\sum_{i=1}^{s} n_i}$$

Přejdeme k náhodným veličinám a definujeme náhodnou veličinu W^*

$$W^* = \frac{\sum_i E_i}{\sum_i N_i},$$

kde N_i a E_i jsou také náhodné veličiny. Platí, že různé dvojice (N_i, E_i) jsou nezávislé a rovnoměrně rozložené (*iid*[2]), ale pro dané i nemusí být N_i a E_i *iid*. Musíme najít výraz s nezávislými proměnnými.

Definujeme distribuční funkci náhodné veličiny W^*

$$P(W^* < x) = P\left(\frac{\sum_i E_i}{\sum_i N_i} < x\right)$$

a upravíme do tvaru

$$P(W^* < x) = P\left(\sum_i (E_i - x \cdot N_i) < 0\right)$$

[2]Independent and identically distributed

Definujeme náhodnou veličinu Z_i^x

$$Z_i^x = E_i - x \cdot N_i \ ,$$

pro kterou platí, že Z_i^x jsou *iid*, protože dvojice (N_i, E_i) jsou také nezávislé.

Centrální limitní teorém (Central Limit Theorem)

Centrální limitní teorém je přechodem od rozdělení nějaké náhodné veličiny k normálnímu rozdělení. Podmínkou je, aby zkoumaná náhodná veličina byla sumou nebo průměrem n nezávislých náhodných veličin.

Existuje několik definic centrálního limitního teorému, uvádím zde pouze jednu verzi (převzato z [10])

Definice - Centrální limitní teorém

Mějme n nezávislých, stejně distribuovaných náhodných veličin $X_1, X_2, \ldots X_n$ se střední hodnotou $mean(X)$ a nenulovou variancí $var(X)$. Nechť $S_n = \sum_{i=1}^{n} X_i$.

Pak platí vztah

$$\lim_{n\to\infty} P\left(\frac{S_n - n \cdot mean(X)}{\sqrt{(s)} \cdot var(X)} < x\right) = \Phi(x) \ ,$$

kde $\Phi(x)$ je distribuční funkce normálního normovaného rozdělení, tj. pravděpodobnost, že náhodná veličina s normálním normovaným rozdělením je menší než x.

Aplikace centrálního limitního teorému

Centrální limitní teorém chceme aplikovat na vztah

$$P(W^* < x) = P\left(\sum_i Z_i^x < 0\right)$$

Výraz v závorce tedy musíme upravit:

$$\begin{aligned} P(W^* < x) &= P\left(\sum_i Z_i^x - s \cdot mean(Z^x) < -s \cdot mean(Z^x)\right) = \\ &= P\left(\frac{\sum_i Z_i^x - s \cdot mean(Z^x)}{\sqrt{s} \cdot sd(Z^x)} < \frac{-s \cdot mean(Z^x)}{\sqrt{s} \cdot sd(Z^x)}\right) = \\ &= P\left(\frac{\sum_i Z_i^x - s \cdot mean(Z^x)}{\sqrt{s} \cdot sd(Z^x)} < \frac{-\sqrt{s} \cdot mean(Z^x)}{sd(Z^x)}\right) \approx \\ &\approx \Phi\left(\frac{-\sqrt{s} \cdot mean(Z^x)}{sd(Z^x)}\right) \end{aligned}$$

$sd(Z^x)$ označuje odchylku náhodné veličiny Z^x

$$sd(Z^x) = \sqrt{var(E) + x^2 \cdot var(N) - 2 \cdot x \cdot cov(N, E)}$$

$mean(Z^x)$ je střední hodnota náhodné veličiny Z^x

$$mean(Z^x) = mean(E) - x \cdot mean(N)$$

Výpočet mezí intervalu spolehlivosti

Pro určení intervalu spolehlivosti zvolíme hodnotu distribuční funkce Φ, určíme odpovídající kvantil l a vyřešíme rovnici s neznámou x

$$\frac{-\sqrt{s} \cdot mean(Z^x)}{sd(Z^x)} = l$$

Po dosazení

$$\frac{-\sqrt{s} \cdot (mean(E) - x \cdot mean(N))}{\sqrt{var(E) + x^2 \cdot var(N) - 2 \cdot x \cdot cov(N, E)}} = l$$

Po umocnění, vynásobení a uspořádání dostaneme kvadratickou rovnici pro neznámou x

$$(l^2 \cdot var[N] - s \cdot (mean[N])^2) \cdot x^2 + (2 \cdot s \cdot mean[N] \cdot mean[E] - 2 \cdot l^2 \cdot cov[N, E]) \cdot x + \\ + (l^2 \cdot var[E] - s \cdot (mean[E])^2) = 0$$

Řešení x_1, x_2 této rovnice jsou meze hledaného intervalu spolehlivosti.

Tento postup navržený v [5] je výhodný proto, že ho lze aplikovat pouze na jedna testovaná data. Jiná metoda určování intervalu spolehlivosti spočívá v provádění Monte Carlo simulací.

5.3.3 Postup pro výpočet intervalu spolehlivosti

Mějme sadu s vět.

Označme n_i délku i-té věty a e_i počet chyb v i-té větě, tj. počet rozdílů mezi rozpoznanou větou a větou referenční.

Dále označme N diskrétní náhodnou veličinu udávající délku vět, která nabývá hodnot n_i, a E diskrétní náhodnou veličinu udávající počet chyb, která nabývá hodnot e_i.

1. Spočítáme střední hodnotu náhodných veličin N, E, N^2, E^2 a $E \cdot N$, tj. spočteme

   - $mean[N] = \frac{1}{n} \cdot \sum_{i=1}^{s} n_i$

- $mean[E] = \frac{1}{n} \cdot \sum_{i=1}^{s} e_i$
- $mean[N^2] = \frac{1}{n} \cdot \sum_{i=1}^{s} n_i^2$
- $mean[E^2] = \frac{1}{n} \cdot \sum_{i=1}^{s} e_i^2$
- $mean[N \cdot E] = \frac{1}{n} \cdot \sum_{i=1}^{s} (n_i \cdot e_i)$

2. Vypočítáme variance náhodných veličin N a E:
   - $var[N] = mean[N^2] - (mean[N])^2$
   - $var[N] = mean[E^2] - (mean[E])^2$
3. Vypočítáme vzájemnou kovarianci náhodných veličin N a E:
   - $cov[N, E] = mean[N \cdot E] - mean[N] \cdot mean[E]$
4. Zvolíme $100 \cdot (1 - \alpha)$%-ní interval spolehlivosti a určíme hodnotu l jako $(1 - \frac{\alpha}{2})$-tý kvantil normálního normovaného rozdělení (často používané hodnoty jsou uvedeny v tabulce 5.3)
5. Vyřešíme kvadratickou rovnici

$$(l^2 \cdot var[N] - s \cdot (mean[N])^2) \cdot x^2 + (2 \cdot s \cdot mean[N] \cdot mean[E] - 2 \cdot l^2 \cdot cov[N, E]) \cdot x + \\ + (l^2 \cdot var[E] - s \cdot (mean[E])^2) = 0$$

6. Určíme interval spolehlivosti (x_1, x_2)

$100 \cdot (1 - \alpha)$%-ní interval	hodnota kvantilu $u_{1-\frac{\alpha}{2}}$
90 %	1.64
95 %	1.96
99 %	2.58

Tab. 5.3: Některé kvantily normovaného normálního rozdělení používané pro intervaly spolehlivosti

5.3.4 Vlastní implementace algoritmu

Pomocí vektoru počtu chyb ve větách a vektoru délek vět jsem spočítal podle výše zmíněných vzorečků střední hodnoty a variance počtu chyb a délek vět a vzájemnou kovarianci.

Hodnotu parametru l jsem zvolil jako 90% kvantil, což odpovídá hodnotě $l = 1.64$.

Kvadratickou rovnici pro výpočet hraničních bodů intervalu spolehlivosti x_1, x_2 jsem vyřešil známým vzorečkem

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

kde

- $a = l^2 \cdot var[N] - s \cdot (mean[N])^2$
- $b = 2 \cdot s \cdot mean[N] \cdot mean[E] - 2 \cdot l^2 \cdot cov[N, E]$
- $c = l^2 \cdot var[E] - s \cdot (mean[E])^2$

Intervaly spolehlivosti se standardně počítají pro *WER*, tj. udávají chybovost systému. Protože v mých statistikách se vyskytuje přesnost (*Accuracy*), přepočítal jsem následujícím způsobem vypočtené meze, tak aby interval odpovídal přesnosti systému, nikoli chybovosti, a interval byl vyjádřen v procentech:

- $x_1^* = (1 - x_2) \cdot 100\%$
- $x_2^* = (1 - x_1) \cdot 100\%$

5.3.5 Výsledky

Výpočet intervalů spolehlivosti jsem testoval na datech s téměř 2000 promluv. Rozsahy intervalů tak vyšly poměrně malé, viz kapitola 6.

5.4 Konfúzní tabulka

Konfúzní tabulka obsahuje informace o úspěšnosti detekce hlásek či jejich záměn za hlásky jiné. Je v ní zaznamenáno, jak často byla příslušná hláska rozpoznána správně, kolikrát byla zaměněna za jinou, kolikrát byla vložena navíc či vynechána.

Ukázka části konfúzní tabulky je na obr. 5.4. Čísla na hlavní diagonále udávají absolutní četnost správného rozpoznání, čísla mimo diagonálu znamenají počet chybné segmentace slova/písmene za slovo/písmeno jiné. V tabulce jsou zahrnuty i informace o počtu vložení dané položky a počtu smazání (proto slupec a řádek DEL, INS).

	DEL	a	b	c	d	e	f	g	...
INS	0	0	0	0	0	1	0	0	
a	4	45	0	0	0	0	0	1	
b	0	2	18	0	0	0	3	0	
c	2	0	0	32	0	0	0	4	
d	0	0	0	0	65	0	0	0	
e	0	1	0	0	0	14	0	2	
f	0	0	0	0	0	0	20	0	
g	3	0	0	0	1	3	0	29	
...									

Tab. 5.4: Konfúzní tabulka

5.4.1 Vytváření konfúzní tabulky

Vycházel jsem z řetězce operací, které jsou jedním z výstupů *levenshteinova algoritmu*. V případě operace `d`, `i` nebo `s` inkrementuji příslušné políčko tabulky. Pokud se písmeno/slovo v tabulce zatím nevyskytuje, zvětším tabulku o jeden řádek a jeden sloupec.

5.4.2 Využití konfúzní tabulky

Hodnoty v konfúzní tabulky využívám pro výpisy častých záměn slov či písmen. Jelikož tabulka obsahuje absolutní četnosti výskytů, musím hodnoty normalizovat, abych získal relativní četnost konkrétních záměn.

Vypsání záměn slov a písmen

Při vypsání záměn slov a písmen během rozpoznávání jsem postupně procházel po řádcích konfúzní a hledal nenulová čísla mimo hlavní diagonálu. Chybné rozpoznávání jsem zazna-

menával do souboru (viz. kapitola 6) včetně absolutní a relativní četnosti, tj. počtu výskytu daného problému děleného celkovým počtem všech položek v porovnávaných promluvách.

5.5 Vyhodnocení pro clustery

Pojem *cluster* značí určitou třídu, do které promluva spadá. Můžeme např. rozlišovat, zda byla namluvena mužem nebo ženou, mladou či starší osobou, zda je zašuměná či bez šumu nebo zda se jedná o větu tázací ši oznamovací.

Ve svém skriptu předpokládám, že mám k dispozici soubor s těmito informacemi o jednotlivých promluvách. Uživatel tedy při spouštění skriptu pouze zadá, pro který *cluster* chce vyhodnocení provézt, např.

```
python MLF_evaluate.py ref.txt rec.txt clusters.txt men
```

Program vybere z množiny vět pouze ty, které vyhovují zadaným požadavkům a provede pro ně kompletní vyhodnocení.

Soubor obsahující informace o clusterech, výše označen jako `clusters.txt`, má následující strukturu:

```
id_vety cluster1 cluster2 ...
```

Konkrétní příklad souboru s clustery je na obr. 5.5.

a0001	noised	men
a0002	women	
a0003	men	noised
a0004	men	nonoise
a0005	nonoise	
...		

Obr. 5.4: Ukázka souboru s clustery

Počítal jsem také s možností, že by bylo potřeba vyhodnotit průnik několika různých *clusterů*, např. když chceme zjistit přesnost systému rozpoznávání u mladých žen či u starších mužů v zašuměném prostředí.

5.5.1 Postup vyhodnocení pro clustery

1. načtení souboru s clustery a požadovaných clusterů (vstupní parametry skriptu)
2. vyfiltrování promluv podle zadaných vstupních parametrů
3. kompletní vyhodnocení vyfiltrovaných promluv jako při standardním vyhodnocení
4. výpis výsledků a statistik

5.6 Hromadné vyhodnocování

Skript je uzpůsoben k dávkovému zpracování, tj. nevyžaduje žádnou interakci uživatele. Vstupními parametry jsou pouze

- referenční promluvy
- rozpoznané promluvy
- soubor s clustery
- požadované clustery

Poslední dva parametry jsou nepovinné.

Výstupní soubory obsahují v názvu datum a čas spuštění skriptu a název souboru s rozpoznanými promluvami, pro snadnou orientaci uživatele.

Spouštění může vypadat např. takto:

```
python MLF_evaluate.py ref1.txt rec1.txt clusters.txt men,noised
python MLF_evaluate.py ref2.txt rec2.txt
python MLF_evaluate.py ref3.txt rec3.txt clusters.txt women
```

5.7 Chybovost na začátku a konci slov a promluv

Chybovost na začátku a konci promluv jsem určoval z řetězce operací získaného levenshteinovou metodou použitou pro slova. V případě, že první, resp. poslední operací bylo `d`, `i` nebo `s`, inkrementoval jsem čítač chyb na začátku, resp. konci promluv.

K určení chybovosti na začátku a konci slov jsem použil kompletně vytvořenou konfúzní tabulku slov. Nenulová čísla mimo hlavní diagonálu zde určují chybu v rozpoznání slova. Pro tato slova jsem zjistil, zda se shodují první, resp. poslední znaky. Pokud v nich byla chyba, inkrementoval jsem čítač chyb na začátku, resp. konci slova.

5.8 Záměna sekvencí slov

Záměnou sekvencí slov rozumíme rozpoznání nx slov za ny slov, typicky 2 za jedno nebo naopak. Příkladem takovýchto frází je

- `okolo` → `o kolo`
- `nebyl` → `ne byl`
- `že na` → `žena`
- `a si` → `asi`

Vyšel jsem z předpokladu, že tyto případy vznikají vložením nebo odebráním mezery při rozpoznávání, jinak jsou tyto řetězce totožné. Pro nalezení tohoto jevu bylo vhodné použít řetězec operací získaný levenshteinovou metodou. Pokud operace `d`, resp. `i` v řetězci operací odpovídala mezeře v rozpoznané, resp. referenční promluvě, otestoval jsem okolní prvky, zdali jsou správně rozpoznané. Pokud ano, jednalo se záměnu sekvencí slov přidáním/odebráním mezery.

Kapitola 6

Výsledky

Všechny výstupní soubory obsahují v názvu datum a čas spuštění skriptu, aby bylo snadné porovnávat jednotlivé výstupy při opakovaném nebo hromadném spouštění skriptu:

`nazev_rok-mesic-den_hodiny-minuty-vteriny.pripona`

Výstupy skriptu provádějícím kompletní vyhodnocení kvality rozpoznávání řeči jsou následující:

6.1 Konzolový výstup

Konzolový výstup obsahuje pouze informace o načtení vstupních souborů, počtu promluv a statistiky úspěšnosti rozpoznání vět.

```
python MLF_evaluate.py SD-E_test_partA_words.txt e01_256-64_recog.txt
Reading  SD-E_test_partA_words.txt
Readed total:  1922 records
Reading  e01_256-64_recog.txt
Readed total:  1922 records
Detected "*" wildchar: */a30115a4.lab
======================
Rejection:
       OK: 1597 (83.09 %)             OK_rej:    0 ( 0.00 %)
 FAIL_rej:    0 ( 0.00 %)      FAIL_accepted:  325 (16.91 %)
--------------------------
Sentences:  (total 1922)
    OK  : 1597 (83.09 %)       FAIL :  325 (16.91 %)
--------------------------
```

6.2 HTML soubor

Výstupní soubor ve formátu html (`statistics_rok-mesic-den_hodina-minuta-vterina.txt`) obsahuje kompletní vygenerované statistiky úspěšnosti. Ukázka tohoto výstupu je na následujících dvou stránkách na obr. 6.1, 6.2.

Soubor názvy vstupních souborů, v případě vyhodnocování pro clustery i názvy požadovaných clusterů. Dále jsou v tabulce přehledně vypsány statistiky pro jednotlivé sledované skupiny.

Výsledky jsou charakterizovány dvěma parametry. Prvním parametrem je procento úspěšnosti *Corr* a druhým o něco přísnějším parametrem popisujícím rozpoznávací systém je procento přesnosti *Acc*:

$$\begin{aligned} Corr &= \frac{N - S - D}{N} \cdot 100\% \\ Acc &= \frac{N - S - D - I}{N} \cdot 100\% \,, \end{aligned}$$

kde N je celkový počet elementů (slov, písmen), hodnota S udává počet substitucí, D je mazání a I je počet vkládání.

Všechny procentuální statistické výsledky jsou též reprezentovány intervalem spolehlivosti (výpočet viz 5.3). Intervaly jsou poměrně úzké, protože vzorků bylo velké množství (téměř 2000).

6.2.1 Rozbor výsledků

Chyby na začátku a na konci promluv

Z mých testů vyplývá, že na začátku promluv je více chyb v podobě mazání nebo vkládání, na konci promluv zase převažují chyby substituční.

Krátká vs. dlouhá slova

Dle očekávání je rozpoznávání delších slov podstatně přesnější. Důvodem bude podle mého názoru velké množství krátkých slov, která si nejsou zcela nepodobná a systém je tedy při rozpoznávání může snadno zaměnit.

MLF evaluate statistics

2012-05-20_11-28-21

Reference file: SD-E_test_partA_words.txt

Recognized file: e01_256-64_recog.txt

Test type	Results	
Rejection	OK:	1597 (83.09%)
	OK_rej:	0 (0.00 %)
	FAIL_rej:	0 (0.00 %)
	FAIL_accepted:	325 (16.91 %)
Sentence	OK:	1597 (83.09%)
	FAIL:	325 (16.91 %)
	sum=	1922
Words	Corr:	8032 (93.45%) - (92.68%,94.22%)
	Acc:	92.07% - (91.18%,92.95%)
	H=	8032
	D=	181
	S=	382
	I=	119
	N=	682
	sum count=	8595
	sum OK+FAIL=	8714
Short words	Corr:	2237 (86.47%) - (84.61%,88.40%)
	Acc:	81.87% - (79.56%,84.25%)
	H=	2237
	D=	145
	S=	205
	I=	119
	N=	469
	sum count=	2587
	sum OK+FAIL=	2706
Long words	Corr:	5797 (97.56%) - (97.18%,97.94%)
	Acc:	97.02% - (96.58%,97.46%)
	H=	5797
	D=	37
	S=	108
	I=	32
	N=	177
	sum count=	5942
	sum OK+FAIL=	5974
Chars	Corr:	49926 (97.59%) - (97.30%,97.87%)
	Acc:	96.73% - (96.38%,97.07%)
	H=	49926
	D=	700
	S=	533
	I=	441
	N=	1674
	sum count=	51159
	sum OK+FAIL=	51600
Vowels	Corr:	16860 (97.62%) - (97.30%,97.93%)
	Acc:	97.00% - (96.64%,97.36%)
	H=	16860
	D=	231
	S=	180
	I=	107
	N=	518
	sum count=	17271
	sum OK+FAIL=	17378
Consonants	Corr:	25600 (97.77%) - (97.49%,98.04%)
	Acc:	96.88% - (96.53%,97.22%)
	H=	25600
	D=	305
	S=	280
	I=	233
	N=	818
	sum count=	26185
	sum OK+FAIL=	26418

Obr. 6.1: Ukázka výstupu - HTML soubor (1. část)

Okluzivy	Corr:	8945 (97.82%) - (97.48%,98.16%)
	Acc:	97.06% - (96.66%,97.45%)
	H=	8945
	D=	129
	S=	70
	I=	70
	N=	269
	sum count=	9144
	sum OK+FAIL=	9214
Nazaly	Corr:	4087 (98.53%) - (98.17%,98.88%)
	Acc:	97.61% - (97.15%,98.08%)
	H=	4087
	D=	44
	S=	17
	I=	38
	N=	99
	sum count=	4148
	sum OK+FAIL=	4186
Frikativy	Corr:	6400 (98.08%) - (97.73%,98.44%)
	Acc:	96.52% - (96.03%,97.01%)
	H=	6400
	D=	83
	S=	42
	I=	102
	N=	227
	sum count=	6525
	sum OK+FAIL=	6627
Semiokluzivy	Corr:	1787 (99.28%) - (98.93%,99.63%)
	Acc:	98.33% - (97.79%,98.87%)
	H=	1787
	D=	8
	S=	5
	I=	17
	N=	30
	sum count=	1800
	sum OK+FAIL=	1817
Vibranty	Corr:	2155 (99.08%) - (98.70%,99.46%)
	Acc:	98.21% - (97.71%,98.70%)
	H=	2155
	D=	17
	S=	3
	I=	19
	N=	39
	sum count=	2175
	sum OK+FAIL=	2194
Aproximanty	Corr:	2240 (98.12%) - (97.58%,98.66%)
	Acc:	97.07% - (96.41%,97.72%)
	H=	2240
	D=	34
	S=	9
	I=	24
	N=	67
	sum count=	2283
	sum OK+FAIL=	2307
Errors in words	Beginning:	301 (3.50%)
	End:	323 (3.76 %)
Errors in speeches	Beginning:	D= 43, I= 45, S= 93
	End:	D= 29, I= 8, S= 98

Obr. 6.2: Ukázka výstupu - HTML soubor (2. část)

Chyby na začátku a konci slov

Počet chyb na začátku a konci slov se nijak výrazně neliší.

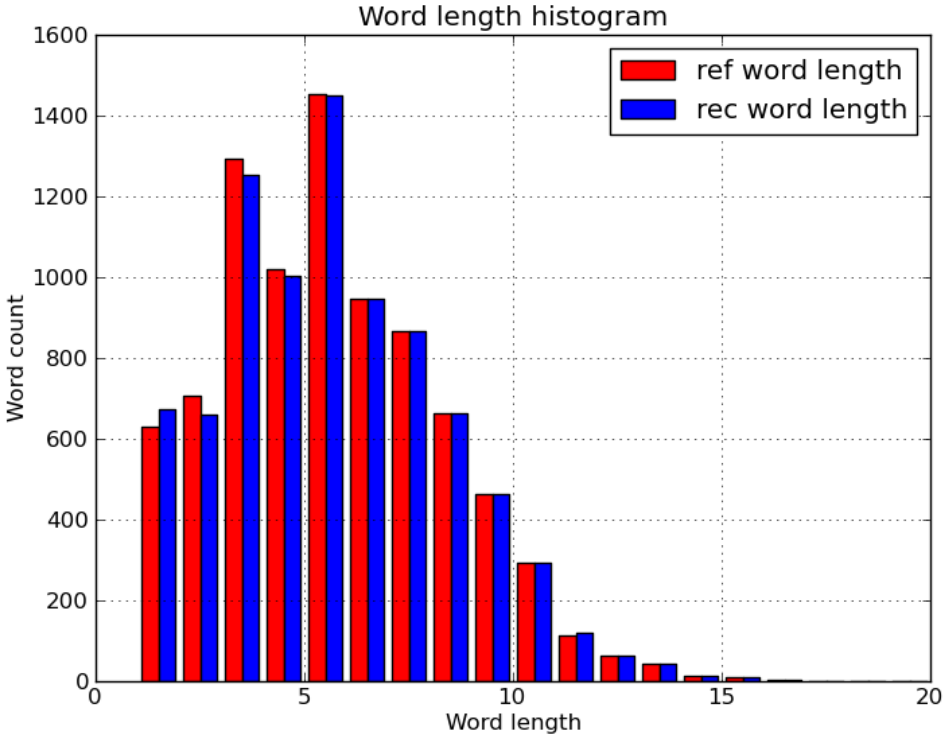
Samohlásky vs. souhlásky

V přesnosti rozpoznávání souhlásek a samohlásek není podstatný rozdíl. Z typů souhlásek jsou nejvíce problematické okluzivy, což pravděpodobně souvisí s průběhem jejich akustického signálu.

6.3 Graf porovnání délek slov

Graf je uložen jako obrázek s názvem
`word_length_hist_rok-mesic-den_hodina-minuta-vterina.txt`.

Ukázka tohoto výstupu je na obr. 6.3.



Obr. 6.3: Porovnání délek slov

Z histogramu je patrné, že se délky slov v referenčních a rozpoznaných promluvách až na malé odchylky v kratších slovech (do 5 písmen) neliší.

6.4 Záměna sekvencí slov

Tento jev je zachycen v souboru `phrases_rok-mesic-den_hodina-minuta-vterina.txt`. Ukázka výstupu pro jedna z dat, které jsem měl k dispozici:

```
dvanáct é -> dvanácté
a si -> asi
že na -> žena
že na -> žena
```

6.5 Konfúzní tabulky

Konfúzní tabulka pro písmena je uložena v souboru typu `csv`:
(`table_chars_rok-mesic-den_hodina-minuta-vterina.csv`)

Ukázka části vygenerované tabulky viz tab. 6.5. Největší hodnoty v tabulce vyšly skutečně na hlavní diagonále.

	DEL	p	o	s	u	n
INS	0	29	33	26	40	35
p	163	1181		1	0	0 2
o	384	3	2712	3	28	1
s	227	1	4	2232	5	3
u	140	2	7	0	1262	3
n	253	2	5	4	1	2574

Tab. 6.1: Ukázka vygenerované konfúzní tabulky

V souboru `table_words_rok-mesic-den_hodina-minuta-vterina.csv` je uložena konfúzní tabulka pro slova.

6.6 Časté záměny písmen

Časté záměny písmen jsou zapsány do souboru

`CharErrors_rok-mesic-den_hodina-minuta-vterina.txt`.

Ukázka části vygenerovaného souboru viz. níže.

```
b -> v 19x (0.0343 %)
v -> n 6x (0.0108 %) z 6x (0.0108 %)
i -> t 7x (0.0126 %) space 6x (0.0108 %) e 10x (0.0180 %) í 13x (0.0235 %) ...
á -> a 9x (0.0162 %)
é -> e 12x (0.0217 %)
j -> l 9x (0.0162 %)
y -> i 8x (0.0144 %)
```

6.7 Časté záměny slov

Časté záměny písmen jsou zapsány do souboru

`WordErrors_rok-mesic-den_hodina-minuta-vterina.txt`, obsah souboru je ekvivalentní souboru v 6.6.

Kapitola 7

Závěr

Tématem bakalářské práce je podrobné vyhodnocování kvality systémů rozpoznávání řeči. Mým úkolem bylo zajistit kompatibilitu dodaného skriptu s referenčním `Hresult.exe` z balíku HTK a skript rozšířit o vyhodnocování krátkých a dlouhých slov, souhlásek, souhlásek a různých typů souhlásek. Měl jsem také vypsat časté záměny písmen, slov a sekvencí slov, ze kterých by bylo možné zkoumat příčiny špatného rozpoznávání a případně upravovat parametry rozpoznávacího systému. K přesnějšímu porovnání rozpoznávacích systémů a přesnějšímu vyhodnocení pomáhají ještě další údaje jako například intervaly spolehlivosti. Rozšířený skript jsem testoval na dodaných datech, výsledky (Acc/Corr) se shodují s `Hresult.exe` a jsou podrobně popsány v kapitole 6. Skript by měl sloužit nejen k určení úspěšnosti systému rozpoznávání, ale i k detekci častých chyb a problémů. Zkoumání podrobných vyhodnocení by mělo vést ke zvýšení přesnosti rozpoznávání a tak k celkovému zlepšení výsledků rozpoznávání řeči.

Literatura

[1] *Prof. Ing. Mařík V., SCs., Prof. RNDr. Štěpánková O., CSs., Ing. Lažanský J., Ph.D. a kol.:*

Umělá inteligence (5)

Academia, Praha, 2007

[2] *Prof. Ing. Psutka J., CSc.:*

Komunikace s počítačem mluvenou řečí

Academia, Praha, 1995

[3] *Prof. Ing. Psutka J., CSc., Doc. Ing. Matoušek J., Ph.D., Doc. Ing. Müller L., Ph.D., Doc. Dr. Ing. Radová V.:*

Mluvíme s počítačem česky

Academia, Praha, 2006

[4] *Wikipedia, the free encyclopedia*

Levenshtein distance

http://en.wikipedia.org/wiki/Levenshtein_distance

[5] *Vilar Jean Miguel, Lenzo K.A.:*

Efficient Computation of Confidence Intervals for Word Error Rate.

Departmento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Castellon, 2008

[6] *Young S., Kershaw D., Odell J., Ollason D., Valtchev V., Woodland P.*

The HTK Book

Cambridge University Engineering Department

http://htk.eng.cam.ac.uk

[7] *RNDr. Kobeda Z.*

Přednášky k předmětu KMA/Pravděpodobnost a statistika

KMA, ZČU Plzeň, 2009

[8] *Brown P.; Cocke J., Della Pietra S., Della Pietra V., Jelinek F., Mercer R., Roossin P.*

A statistical approach to language translation.

Thomas J.Watson Research Center, Yorktown Heights, NY, 1988

http://www.aclweb.org/anthology-new/C/C88/C88-1016.pdf

[9] *Dana Lee Ling*

Confidence Intervals

Collage of Micronesia

http://www.comfsm.fm/~dleeling/statistics/

[10] *Prof.Stanton Ch.*

The Central Limit Theorem

Department od Mathematics, California State University, San Bernardino

http://www.math.csusb.edu/faculty/stanton/probstat/clt.html

[11] *Blunsom P.*

Hidden Markov Model

2004

http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf

Dodatek A

Programová dokumentace

Tato část slouží jako návod na používání vytvořeného skriptu `MLF_evaluate.py`, který podrobně vyhodnocuje kvalitu rozpoznávání řeči.

A.1 Spouštění skriptu

Skript se spouští příkazem

`python MLF_evaluate.py soubor_ref soubor_rec [soubor_s_clustery clustery]`, kde vstupními parametry jsou

- `soubor_ref` - soubor obsahující referenční promluvy, povinný parametr
- `soubor_rec` - soubor obsahující rozpoznané promluvy, povinný parametr
- `soubor_s_clustery` - soubor obsahující informace o příslušnosti promluv ke clusterům
- `clustery` - názvy požadovaných clusterů oddělené čárkami (bez mezery)

Poslední dva parametry jsou nepovinné, ale v případě, že chceme vyhodnocovat věty příslušící jen k některým clusterům, musí být uvedeny oba.

A.1.1 Vypsání nápovědy

Pro vypsání nápovědy lze použít parametr `-h`:

```
python MLF_evaluate.py -h
Script for speech recognition evaluation
```

```
Usage: python MLF_evaluate.py reference_file recognized_file
python MLF_evaluate.py reference_file recognized_file clusters_file comma,selected,clusters
for clusters evaluation

-h   print this help message end exit
```

A.1.2 Příklad spouštění skriptu pro vyhodnocení všech vět

```
python MLF_evaluate.py ref.txt rec.txt
```

A.1.3 Příklad spouštění skriptu pro vyhodnocení vět odpovídajících požadovaným clusterům

```
python MLF_evaluate.py ref.txt rec.txt clusters.txt man,noise
```

Ukázka vzhledu souboru s clustery viz. níže:

```
a0001 noised men
a0002 women
a0003 men noised
a0004 men nonoise
a0005 nonoise
```

A.1.4 Výstupní soubory

Všechny výstupní soubory obsahují v názvu datum a čas spuštění skriptu.

- `statistics_rok-mesic-den_hodina-minuta-vterina.txt` - obsahuje kompletní vyhodnocení
- `word_length_hist_rok-mesic-den_hodina-minuta-vterina.txt` - histogram porovnání délek jednotlivých slov v referenčních a rozpoznaných promluvách
- `phrases_rok-mesic-den_hodina-minuta-vterina.txt` - obsahuje detekované záměny sekvencí slov
- `table_chars_rok-mesic-den_hodina-minuta-vterina.csv` - konfúzní tabulka pro písmena, vhodné otevírat v programech balíku Office

- `table_words_rok-mesic-den_hodina-minuta-vterina.csv` - konfúzní tabulka pro slova, vhodné otevírat v programech balíku Office
- `CharErrors_rok-mesic-den_hodina-minuta-vterina.txt` - obsahuje výpis všech záměn písmen včetně jejich absolutního a relativního počtu
- `WordErrors_rok-mesic-den_hodina-minuta-vterina.txt` - obsahuje výpis všech záměn slov včetně jejich absolutního a relativního počtu