ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA RADIOELEKTRONIKY

# BAKALÁŘSKÁ PRÁCE

## Jazykové modely pro rozpoznávání řeči v různých tematických oblastech

## Language modelling for speech recognition in fields of various topic

**Studijní program:** Komunikace, multimédia a elektronika
**Obor:** Multimediální technika
**Autor:** Jiří Valíček
**Vedoucí práce:** Doc. Ing. Petr Pollák, CSc.

**Praha, 2014**

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra radioelektroniky

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jiří Valíček**

Studijní program: Komunikace, multimédia a elektronika
Obor: Multimediální technika

Název tématu: **Jazykové modely pro rozpoznávání řeči v různých tematických oblastech**

Pokyny pro vypracování:

1. Seznamte se s problematikou jazykového modelování v systémech rozpoznávání spojité řeči.
2. Zkompletujte dostupné jazykové modely a textové korpusy pro tvorbu tematicky specifických modelů pro rozpoznávání řeči v různých tematických oblastech. Připravte nástroje i proceduru pro vytvoření modelu pro novou tematickou oblast.
3. Vytvořené modely a jejich kombinace otestujte na testovacích textových korpusech, případně v poskytnutém rozpoznávači řeči.

Seznam odborné literatury:

[1] J. Psutka, L. Miller, J. Matoušek, V. Radová: Mluvíme s počítačem česky. Academia, 2006.
[2] X. Huang, A. Acero, H.-W. Hon: Spoken Language Processing. Prentice-Hall 2001.
[3] S. Yuong et.al.: The HTK Book. (for HTK Version 4.2.1). Cambridge University Engineering Department, 2009.
[4] SRILM - The SRI Language Modeling Toolkit. WEBová stránka http://www.speech.sri.com/projects/srilm

Vedoucí: doc.Ing. Petr Pollák, CSc.

Platnost zadání: do konce letního semestru 2014/2015

Prof. Ing. Miloš Klíma, CSc.
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 2. 2014

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 23. května 2014 .....

Jiří Valíček

## Poděkování

Rád bych poděkoval Doc. Ing. Petrovi Pollákovi, CSc. za vedení této práce a vstřícnost při řešení všech problémů, dále Ing. Petru Mizerovi, za pomoc s jazykovým rozpoznáváním. Velké díky chci věnovat svým rodičům za morální podporu při studiu a nejen za ni.

**Název práce:** Jazykové modely pro rozpoznávání řeči v různých tematických oblastech

**Autor:** Jiří Valíček

**Katedra (ústav):** Katedra radioelektroniky

**Vedoucí bakalářské práce:** Doc. Ing. Petr Pollák, CSc.

**e-mail vedoucího:** pollak@fel.cvut.cz

**Abstrakt:** Tato práce je zaměřena na vytvoření tematických jazykových modelů z dostupných korpusů a pracovního postupu, který bude možné v budoucnu aplikovat na nově získané korpusy. Pro zpracování textu byla použita kombinace nástroje sed a programovacího jazyka Perl. K vytvoření jazykových modelů byl použit balík nástrojů SRILM. Vytvořené modely jsou porovnány primárně pomocí perplexity, pro jednu skupinu modelů bylo provedeno i rozpoznávání řeči. Naměřené hodnoty ukazují, že tematické jazykové modely poskytují lepší odhad pravděpodobností než modely obecné a při použití v rozpoznávači řeči, tak snižují jeho chybovost.

**Klíčová slova:** rozpoznávání řeči, jazykový model, perplexita, trénovací korpus, testovací korpus, SRILM, WER

**Title:** Language Modelling for Speech Recognition in Fields of Various Topic

**Author:** Jiří Valíček

**Department:** Department of Radioelectronics

**Supervisor:** Doc. Ing. Petr Pollák, CSc.

**Supervisor's e-mail address:** pollak@fel.cvut.cz

**Abstract:** This thesis is focused on creation of language models from available corpora and work procedure which can be repeated upon future needs on newly obtained corpora. Unix utility sed and programming language Perl was used for text modifications. Language models were created using SRILM toolkit. To compare language models we computed the perplexity using these models on different test corpora. Speech recognition was used as another way to compare newly created languge models. Results obtained from measured variables shows, that our language specific models are better than general models and if used in speech recognition word error rate is reduced.

**Key words:** speech recognition, language model, perplexity, training corpora, test corpora, SRILM, WER

# Obsah

# Kapitola 1

# Úvod

Mezilidskou komunikaci lze rozdělit na verbální a neverbální. Verbální komunikace je nejdůležitějším komunikačním kanálem člověka. Pomocí řeči dokážeme popisovat svoje vnímání okolí a dorozumívat se s ostatními lidmi. Protože je řeč nejjednodušší způsob komunikace, bylo nevyhnutelné, že se pokusíme dorozumívat pomocí řeči i s počítači. O toto se pokoušejí vědci z celého světa přibližně 60 let a i přes počáteční úspěchy slibující brzký úspěch, se s praktickým využitím v běžném životě setkáváme teprve nyní a v omezené míře. [2, str. 195]

Toto vystřízlivění z prvotních úspěchů měli na svědomí překážky, které se snažíme překonat dodnes. Podle těchto překážek můžeme obtížnost rozpoznávání řeči rozdělit na několik kategorií. Rozpoznávání příkazů z omezeného slovníku, rozpoznávání jednotlivě vyslovených slov a rozpoznávání spojité řeči. Pokud je řeč pronášená spontánně, objevují se v ní mimo slov i tzv. neřečové události. K dalším obtížím spontánní promluvy v českém jazyce patří množství nespisovných nebo hovorových výrazů, změna slovosledu a samotná tvorba slov v českém jazyce pomocí skloňování a časování. Z toho je zřejmé, že námi chtěná přirozená komunikace s počítačem pomocí spontánní promluvy z rozsáhlého slovníku je složitý případ.

Dosavadní pokrok lze přisoudit několika faktorům. Jako hlavní bych uvedl neustálé zvyšování výpočetního výkonu a pokrok směrem k vývoji umělé inteligence. Lze sice namítat, že rozpoznávání řeči není umělá inteligence v pravém slova smyslu, ale jistě vykazuje některé její prvky. Jak zmiňuje [8], inteligence je schopnost předpovědi budoucnosti na základě určitých zkušeností.

Předpověď budoucnosti nebo její odhad je hlavní smysl jazykových modelů. Rozpoznávání souvislé řeči můžeme rozdělit do dvou hlavních částí. Část akustickou, která se zabývá modelováním a rozpoznáváním zvukových událostí, to mohou být například formanty a fonémy. Druhá je část jazyková, která popisuje vztahy mezi slovy a způsob tvorby vět v daném jazyce.

Tato práce se zabývá jazykovým modelováním, a to konkrétně modelováním jazyka pomocí $n$-gramových modelů. $n$-gramové modely jsou soubory pravděpodobností výskytu posloupností slov, získané z trénovacích korpusů. Je tedy nutné je vytvořit před samotným rozpoznáváním řeči. Tématicky specifické jazykové modely se získavají kombinací obecného modelu, který postihuje jazyk a jeho běžnou stavbu s modelem specifickým, ten vytvoříme pro danou tematickou oblast. Tímto způsobem vznikne optimalizovaný model pro dané téma.

Práce je členěna do pěti kapitol. Následující druhá kapitola obsahuje teoretickou část této práce. Je zde popsána problematika a členění rozpoznávání řeči. Co je to jazykový $n$-gramový model a další obecné informace související s problematikou vytváření a testování jazykový modelů. Ve třetí kapitole se nachází popis nástrojů použitých při tvorbě

$n$-gramových modelů a popis ostatních nástrojů. Čtvrtá kapitola obsahuje popis jednotlivých modelů a skupin modelů vytvořených v rámci této práce. Poslední pátá kapitola shrnuje výsledky dosažené v této práci a uvádí náměty pro budoucí zlepšení.

# Kapitola 2

# Jazykové modelování v rozpoznávání řeči

Jak bylo napsáno v úvodu, rozpoznávání řeči je složitá úloha, kterou se zabývají výzkumníci již několik desítek let. Rozpoznávání českého jazyka je z pohledu jazykového modelování znesnadněno, kromě hledisek ovlivňující složitost rozpoznávání řeči, i volnějšími pravidly pro skládání slov do vět, množstvím nespisovných výrazů používaných v běžné mluvě a způsobem tvorby slov. Díky skloňování, časování a sedmi pádům patří čeština mezi silně inflektivní jazyky.

Role jazykového modelu v rozpoznávání řeči je postihnutí slovních spojení a slovosledných pravidel. Rozpoznávání řeči je spolupráce dvou odlišných modelů, akustického a jazykového, za účelem vytvoření nejlepší prohledávací strategie. [2, str. 198]

## 2.1 Rozpoznávání řeči

První úspěchy v rozpoznávání řeči byly učiněny již v polovině minulého století, kdy byla úspěšně rozpoznávána čísla oddělená dlouho mezerou a vyslovená jedním řečníkem [9]. Tato situace se dnes považuje za nejjednodušší případ rozpoznávání řeči.

Rozpoznávání řeči je oproti původním předpokladům obtížnější z následujících důvodů [2, str. 195-196]:

- Hlasy lidí nejsou stejné, to je způsobeno parametry hlasového ústrojí a způsobem artikulace. Podle tohoto hlediska lze systémy rozpoznávání řeči dělit na systémy na řečníku závislé (trénování probíhalo na jednom nebo malé skupině řečníků) a systémy na řečníku nezávislé (trénování probíhalo na hlasech stovek nebo i tisíců řečníků).

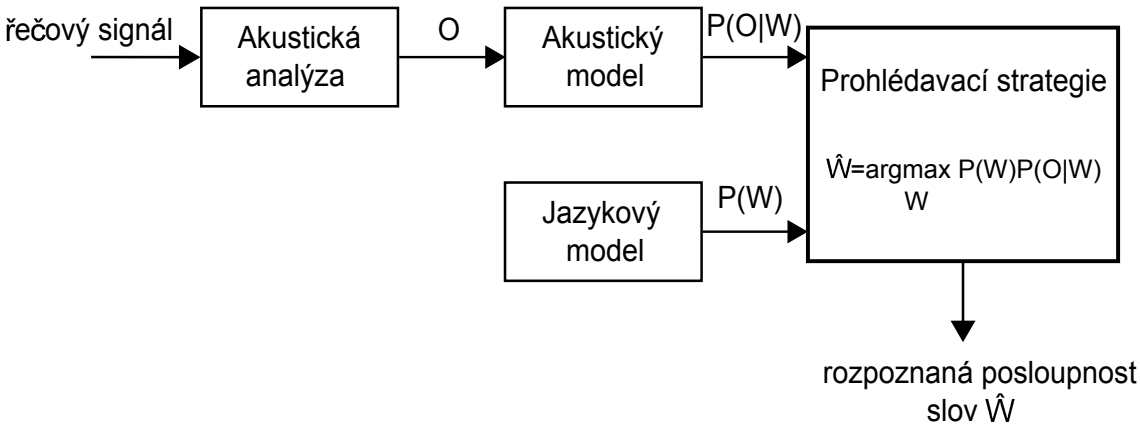
- Nestálost hlasového projevu jedince. Při běžné řeči se mění způsob jakým mluvíme, se situací v níž se nacházíme. Pokud mluvíme šeptem nebo jsme rozčíleni, dochází k výrazné změně způsoby mluvy a dalo by se možná považovat tyto stavy za promluvy jiného řečníka.

- Přítomnost akustického pozadí. Pokud je rozpoznávána promluva, na jejímž pozadí je nezanedbatelný šum, dochází ke ztížení detekce začátků a konců slov. Pokud je takovým šumem další řeč, je nutné takovou odlišit od chtěné promluvy.

- Komplexnost řešené úlohy. První úspěchy v rozpoznávání řeči byly na rozpoznávání izolovaných slov (konkrétně číslovek) z malého slovníku s pauzou. Již při zvětšení tohoto slovníku se úloha stává podstatně obtížnější. Pokud se jedná o souvislou

řeč, tedy bez úmyslných pomlk, navíc například spontánně pronášenou, ve které se vyskytují i neřečové události, je zřejmé, že obtížnost dále roste.

Na obrázku(2.1) je znázorněno blokové schéma systému rozpoznávání řeči.




Obrázek 2.1: Blokové schéma systému rozpoznávání řeči

Úklem tohoto systému je najít takovou posloupnost slov $\hat{W}$, která maximalizuje podmíněnou pravděpodobnost $P(W|O)$, což je nejpravděpodobnější posloupnost $W$ složená z $n$ slov

$$W = \{w_1, w_2..w_n\} \tag{2.1}$$

pro posloupnost příznaků $O$

$$O = \{o_1, o_2...o_T\}. \tag{2.2}$$

charakterizujících řečový signál. Rovnici této posloupnosti slov $\hat{W}$ lze s použitím Bayesova pravidla vyjádřit jako

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|O) = \underset{W}{\operatorname{argmax}}\, \frac{P(W)P(O|W)}{P(O)}. \tag{2.3}$$

Pravděpodobnost $P(O|W)$ označuje šanci vektorů příznaků $O$, při vyslovení posloupnosti $W$. O tuto pravděpodobnost se stará akustický model uvedený v blokovéch schématu. Pravděpodobnost $P(W)$ charakterizuje model jazykový. Protože pravděpodobnost $P(O)$ neovlivňuje hledání maxima posloupnosti $W$, lze ji ignorovat. Posloupnost $\hat{W}$ je možné definovat pomocí maximalizace sdružené pravděpodobnosti $P(W, O)$ jako

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W, O) = \underset{W}{\operatorname{argmax}}\, P(W)P(O|W). \tag{2.4}$$

Výsledek této rovnice je tedy závislý na dvou pravděpodobnostech, $P(W)$ a $P(O|W)$, které odpovídají jazykovému a akustickému modelu. Je tedy zřejmé, že tyto pravděpodobnosti je možné určit nezávisle, tím je dovoleno vytvářet jazykový model odděleně od modelu akustického.

Rozpoznávání řeči je proces stanovení posloupnosti $\hat{W}$, pro kterou posloupnost vektorů příznaků $O$ maximalizuje součin pravděpodobností $P(W)$ a $P(O|W)$. V praktických aplikacích se ovšem používá zjednodušených prohledávacích a rozhodovacích strategií, které snižují náročnost výpočtu a zachovávají přesnost. [2, str. 197-199]

## 2.2 Jazykový model

Smyslem jazykového modelování je definovat zákonitosti daného jazyka, a v případě konkrétních témat, i terminologii a způsob sdělení použitého v dané promluvě. Nejlepších výsledků v systému rozpoznávání řeči dosáhují modely vytvořené dle konkrétní požadavků.

Jazykové modely uvažují způsob tvorby vět v daném jazyce, což zahrnuje použitá slova a pravidla pro jejich řazení. Modelováním těchto zákonitostí se formulují pravděpodobnosti výskytu několika za sebou jdoucích slov. Lze vytvářet modely deterministické, které nedovolují výskyt slov mimo slovník. Používanější jsou však modely, které uvádějí pravděpodobnost i pro slova nevyskytující se v modelu.

Obecně lze říct, že ve dvou různých situacích nejde vyslovit stejnou posloupnost stejně. To by vedlo na vytváření velkého počtu modelů pro každého řečníka. Některé faktory, však ovlivňují jazykový model více, a tak se jim přikládá větší váha. Mezi nejdůležitější kritéria patří jazyk, téma a smysl sdělení.

Při rozpoznávání řeči musí jazykový model poskytovat pravděpodobnosti již v průběhu promluvy, ideálně v reálném čase, aby tak mohl pomoci akustickému modelu v dekódování řeči.

Pokud se budeme zabývat rozpoznáváním spojité řeči, je nutné předpokládat vyslovení libovolné posloupnosti slov bez omezení. Dále by žádná posloupnost neměla nabývat nulové hodnoty, tím by byla vyloučena.

Jazykový model, který určuje apriorní pravděpodobnosti $P(W)$ všech posloupností je nazýván stochastický jazykový model. Pravděpodobnost $P(W)$ obecné posloupnosti $W$, obsahující $K$ slov je určena

$$\begin{aligned} P(W) = P(w_1^K) = P(w_1 w_2 w_3 ... w_K) = \\ = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)...P(w_K|w_1 w_2 ... w_K - 1) = \\ = P(w_1)P(w_2|w_1^1)P(w_3|w_1^2)...P(w_K|w_1^{K-1}) = \prod_{i=1}^{K} P(w_i|w_1^{i-1}) \end{aligned} \quad (2.5)$$

Pokud si vezmeme pouze část této posloupnosti $w_1 w_2 ... w_k (k < K)$, pro její pravděpodobnost platí

$$P(w_1^k) = P(w_1^{k-1})P(w_k|w_1^{k-1}) = P(w_1)P(w_2|w_1^1)P(w_3)(w_1^2)...P(w_k|w_1^{k-1}), \quad k = 2, ..., K. \quad (2.6)$$

Tento rozklad je vhodný pro praktickou implementaci jazykového modelu. Lze pomocí něho rozpoznávát posloupnost již v průběhu promluvy. Výpočet apriorních pravděpodobností $P(w_1^K)$ všech posloupností délky $K$ by bylo velmi složité. V praxi se tedy používá aproximace, která zkracuje historii pouze na posledních $n$ slov. Všechny historie $w_1 ... w_{i-2} w_{i-1}$ shodující se v posledních $n-1$ slovech jsou sloučeny do jedné třídy, což odpovídá aproximaci pravé strany vztahu (2.6) Markovovým modelem $(n-1)$vého řádu. Takové modely nazýváme $n$-gramovými modely.

$n$-gram je posloupnost $n$ slov získaná z trénovacího korpusu. Pokud $n = 0$ označujeme je za zerogamy, $n = 1$ jsou unigramy, $n = 2$ bigramy a $n = 3$ trigramy. Nejčastěji se z praktických důvodů používají bigramy a trigramy. V ideálním případě by mohlo být vhodné použít i jazykové modely s $n >> 3$, ale takové modely jsou velmi rozsáhlé a bylo

by náročné je efektivně používat.

Pro $n$-gramový model platí, že podmíněná pravděpodobnost $P(w_k|w_1^{k-1})$ slova $w_k$ nacházejícího se na pozici $k$ je závislá na $n-1$ slovech, proto lze tuto pravděpodobnost aproximovat jako

$$P(w_k|w_1^{k-1}) \approx P(w_k|w_{k-n+1}^{k-1}) \tag{2.7}$$

a tím pádem platí

$$P(w_1^k) \approx \prod_{i=1}^{k} P(w_i|w_{k-i+1}^{i-1}). \tag{2.8}$$

Hlavní výhoda $n$-gramových modelů je snadný výpočet pravděpodobností, které jsou založeny na zjištění relativních četností výskytu posloupností slov v trénovacím korpusu. Použití $n$-gramových modelů pro český jazyk není úplně ideální. Jak bylo zmíněno dříve, český jazyk nemá příliš pevně dané pořadí slov ve větě, ale snadné vytvoření a úprava $n$-gramových modelů tento nedostatek částečně kompenzují.

Odhad $\bar{P}(w_k|w_{k-2}w_{k-1})$ pravděpodobnosti $P(w_k|w_{k-2}w_{k-1})$ trigramového modelu lze spočítat

$$\bar{P}(w_k|w_{k-2}w_{k-1}) = \frac{N(w_{k-2}, w_{k-1}, w_k)}{N(w_{k-2}, w_{k-1})}, \tag{2.9}$$

kde $N(w_{k-2}, w_{k-1}, w_k)$ je četnost trigramu $w_{k-2}, w_{k-1}, w_k$ a $N(w_{k-2}, w_{k-1})$ je četnost bigramu v trénovacích datech $w_{k-2}, w_{k-1}$. [2, str. 227,228]

## 2.3 Vyhlazování

Při vytváření jazykového modelu počítáme pravděpodobnosti ze slov a jejich posloupností, které se vyskytly v trénovacím korpusu. Ve chvíli, kdy tento model použijeme na testovací korpus, který není s trénovacím korpusem shodný, musíme vzít v potaz slova nevyskytující se v trénovacím korpusu. Pokud bychom tato slova neuvažovali, přiřadili bychom jim nulovou hodnotu. Tím by se při vyhodnocování celkové pravděpodobnosti násobilo nulou. Tento problém se řeší pomocí vyhlazování (smoothing, discounting), tak že se sníží pravděpodobnost všem posloupnostem a tato nevyužitá pravděpodobnost se nechá pro slova a posloupnosti, které se v trénovacím korpusu nevyskytovaly.

### 2.3.1 Good-Turingův odhad

Pro řešení problému s nepozorovanými slovy a posloupnostmi bylo navrženo mnoho různých postupů. Jako velmi úspěšný, a tudíž často používaný se ukázal Good-Turingův odhad. Byl vytvořen Alanem Turingem a jeho asistentem Irvinem J. Goodem, jako způsob odhadu četnosti neznámých živočišných druhů. Je možné ho použít i na jiné jevy.

Jako nepozorované jevy můžeme označit $n$-gramy, které se nevyskytly v trénovacím korpusu. Good-Turingův odhad říká, je-li výskyt daného jevu v celém trénovacím souboru o velikosti $N$ (N je celkový počet trénovacích $n$-gramů) $r$-krát, pak změněná četnost tohoto jevu, kterou označíme r*, je dána předpisem

$$r^* = \frac{(r+1)r_{r+1}}{n_r}, \tag{2.10}$$

kde $n_r$ je počet všech jevů ($n$-gramů) vyskytujících se $r$-krát v trénovacím korpusu. Lze odvodit, že Good-Turingův odhad pravděpodobnosti nepozorovaných jevů je dán četností jevů, které se v trénovacím souboru vyskytují jedenkrát. Takové jevy označujeme za singletony. [2, str. 236,237]

### 2.3.2 Odhad s postupným vynecháváním jednoho jevu

Hlavní myšlenkou tohoto způsobu vyhodnocení je rozdělení trénovacích dat na dvě různé části. První část bude použita na výpočet četností a druhá část bude odložena stranou a použita až pro výpočet parametrů modelu.

Parametry modelu se počítají pro použití například metody maximální věrohodnosti, která při standardním použití [2, str. 232-236] přisuzuje nulovou pravděpodobnost jevům nevyskytujících se v trénovací množině, v našem případě korpusu. Takový odhad je pro rozpoznávání řeči nepoužitelný. Pravděpodobnost, že řečník použije slova mimo trénovací korpus je vysoká, a tak by metoda, která takovým slovům přiřadí nulovou pravděpodobnost, vedla na nulové pravděpodobnosti celku.

Parametry modelu se poté zjistí jako parametry, které nejlépe vystihují odloženou část trénovacích dat. Tím se zvýší správnost odhadu modelu. Nevýhodou této metody je potřeba dalších dat. Tento problém řeší křížová ověřovací technika, která používá obě části rozdělených dat dvakrát. Nejprve se první část použije k výpočtu statistiky a druhá pro odhad parametrů, poté se úlohy prohodí. Speciální křížovou technikou je ověřovací metoda odhadu s postupným vynecháváním jednoho jevu.

Princip metody odhadu s postupným vynecháváním jednoho jevu je shodný s obecnou metodou. Jako jedna část dat slouží jeden $n$-gram a jako druhá část zbytek dat. Tento postup se opakuje pro všechny $n$-gramy. Pokud se bude jako první část dat používat $n$-gram s jedním výskytem, bude simulován výpočet pro $n$-gramy, které se v korpusu nevyskytují. Toto je vhodné, jelikož lze říct, že v obecném korpusu je počet $n$-gramů s právě jedním výskytem vysoký. Další výhoda této metody spočívá v absolutním využití dat. [2, str. 237,238]

## 2.4 Formát ARPA

ARPA (nebo také formát Doug Paul) je formát souboru pro $n$-gramové modely. Tento formát se stal standardem pro většinu nástrojů rozpoznávání řeči, z tohoto důvodu je jeho použití vhodné pro snadnou spolupráci mezi několika různými nástroji.

Na prvním řádku souboru se nachází heslo `\data\`. Na dalších řádcích jsou zapsány počty $n$-gramů daného řádu. $n$-gramy jsou rozděleny podle řádu heslem \N-grams:, kde N je řád $n$-gramu. Řádek s $n$-gramem začíná logaritmickou pravděpodobností (o základu 10) daného $n$-gramu následovaný jednotlivými slovy, která tvoří $n$-gram. Za nimi může být ještě logaritmická hodnota (o základu 10) backoff váhy. Konec souboru je oznámen heslem \end\. Protože $\log(0) = -\infty$, jsou tyto pravděpodobnosti nastaveny na vybranou hodnotu, která je při čtení modelu poté interpretována jako $\log(0)$. Na obrázku je vidět začátek souboru s jazykovým modelem ve formátu ARPA. [7]

## 2.5 Příprava zdrojových textů

Prvním krokem při vytváření jazykového modelu je získání dostatečně rozsáhého korpusu. V ideálním případě bychom používali pouze přepisy promluv, ale protože ty nejsou běžně k dispozici v dostatečném množství, je nutné používat i jiné zdroje. Mezi takové

```
1
2 \data\
3 ngram 1=1137
4 ngram 2=3072
5 ngram 3=168
6
7 \1-grams:
8 -0.9138929      </s>
9 -99     <s>     -0.2795067
10 -1.703538      a       -0.06185211
11 -3.165936      aby     -0.07076145
12 -3.643058      afru    -0.0782582
```

Obrázek 2.2: Jazykový model ve formátu ARPA

patří například knihy různých formátů, články dostupné na internetu nebo internetové diskuze.

Pro potřeby vytvoření jazykového modelu je nutné zajistit značné množství textu. Pokud získáváme text z dopředu známých zdrojů, například celé internetové stránky v HTML nebo zdrojové kódy knih v TEX, lze si připravit nástroje pro zbavení se příkazů těchto programovacích jazyků.

Po odstranění příkazů programovacích jazyků je text téměř připraven. Upravíme soubor, aby obsahoval jednu větu nebo promluvu na řádek. Pokud text obsahuje číslice, je vhodné je přepsat do slovního tvaru nebo celou větu odstranit. Jestliže bychom odstranili pouze číslici, vznikla by chyba ve slovosledu. Při častém výskytu této chyby by byl silně ovlivněn celý model.

Při zpracování přepisů promluv pro modely z korpusu spontánních promluv a korpusu prezentací studentů postgraduálního studia, bylo z důvodu nejasného ukončení vět rozhodnuto o spojení promluv jednotlivých mluvčího do jediné věty. Tím se podstatně zvýšil počet bigramů i trigramů pro některé mluvčí. Zdůvodnění tohoto nárůstu je ve zpracování nahrané řeči, kde při dělší odmlce dojde k ukončení aktuální promluvy a k začátku promluvy další. Pokud mluvčí častěji prokládal svojí promluvu odmlkami, mohly být jeho věty rozděleny i na několik samostaných promluv. Spojením promluv tak dojde k nápravě tohoto problému. Navíc lze předpokládat, že v plynulé promluvě dochází k postupné změně tématu, a tak i při spojení jazykově správně oddělených vět nedojde k velké chybě.

## 2.6 Kombinace modelů

Jedna z nevýhod $n$-gramových modelů je nutnost velkého korpusu pro natrénování modelu. Pokud chceme vytvořit model pro rozpoznávání určitého tématu, je nutné k tomuto tématu získat co nejvíce textů. Sehnat takové množství textu, které by postihlo celé téma včetně obecných jazykových závislostí, je téměř nemožné, proto se jako rozumná cesta ukázalo kombinovat více modelů. K předem připravenému modelu, který postihuje obecné závislosti jazyka, a byl vytvořen z velmi rozsáhleho korpusu, se vytvoří model druhý. Ten bude vytvořen z mnohem menšího tematicky zaměřeného korpusu. Vypočítání pravděpodobností jednotlivých posloupností takového modelu je díky objemu trénovacího korpusu velice rychlé, a tak lze tyto měnší modely vytvářet přesně dle požadavků pro každé rozpoznávání.

Nejjednodušší možností při kombinování jazykových modelů je použití lineární inter-

polace. Lineární interpolaci si lze snadno představit na příkladu hledání hodnoty mezi dvěma body. Tyto dva body spojíme přímkou a hledanou hodnotu odečteme jako průsečík přímky vedené z hledaného místa s přímkou spojující body.

Aplikace lineární interpolace při kombinování modelů lze vyjádřit vztahem [2, str. 253]

$$P_{LI}(w|h) = \sum_{k=1}^{K} \lambda_h(k) \bar{P}(w|h,k), \tag{2.11}$$

kde $P_{LI}(w|h)$ je odhad výsledného interpolovaného modelu, $\lambda_h(k)$ je váha pro $k$-tý model a $\bar{P}(w|h,k)$ je odhad pravděpodobnosti výskytu slova $w$ $k$-tým jazykovým modelem (při dané historii kontextu $h$ - předcházející slova). Přitom musí platit

$$\sum_{w} \bar{P}(w|h,k) = 1, \ pro \ \forall k \tag{2.12}$$

a také

$$\sum_{k=1}^{K} \lambda_h(k) = 1. \tag{2.13}$$

## 2.7 Perplexita

Při vytváření jazykových modelů narazíme na mnoho parametrů, které ovlivňují výsledný model: zvolený korpus, stupeň $n$-gramového modelu, vybraný obecný model pro kombinaci s tematickým modelem, váhování při kombinaci atd. Z toho je zřejmé, že otestovat všechny modely v systému rozpoznávíní řeči by bylo časově velmi náročné. Je proto vhodné využít možnosti otestovat jazykové modely odděleně od systému rozpoznávání řeči. Jazykový model lze ohodnotit na základě jeho předpovědi slov v neznámém textu, a to na základě textu, kterým byl natrénován [2, str. 228]. Nejpoužívanějším způsobem tohoto hodnocení je perplexita.

Perplexita $PP$ je definována vztahem

$$PP = \frac{1}{\sqrt[K]{\overline{P}(w_1 w_2 \dots w_K)}}, \tag{2.14}$$

kde

$$\overline{P}(w_1 w_2 \dots w_K) = \bar{P}(W) \tag{2.15}$$

je odhad apriorní pravděpodobnosti posloupnosti $W$, která obsahuje $K$ slov. Pokud uvážíme pravděpodobnost pro jedno slovo a pravděpodobnost $\bar{P}(W)$, je zřejmé, že v průměru je pravděpodobnost posloupnosti $K$ slov $K$-krát menší než pravděpodobnost jednoho slova. Je tedy vhodné pravděpodobnost posloupnosti normalizovat vzhledem k počtu slov, což je zajištěno

$$\sqrt[K]{\overline{P}(w_1 w_2 \dots w_K)}. \tag{2.16}$$

Tato normalizace nám dovolí porovnávat jazykové modely na různě rozsáhlých korpusech nebo různé modely na jediném korpusu. Pokud spočteme hodnotu výrazu (2.15), značí nižší číslo menší hodnotu pravděpodobnosti normalizovanou na jedno slovo, to znamená vyšší obtížnost rozpoznávání. Příčina nižší hodnoty může být dvojí, nevhodný jazykový

model nebo velká neuspořádanost textu nebo použitého jazyka. Protože má český jazyk volnější pravidla pro tvorbu vět, budou odhady pravděpodobností vyšší než pro jazyky s pevnou stavbou vět.

Někdy se perplexita vyjadřuje logaritmickým zápisem označeným $LP$,

$$LP = \log_2 PP = -\frac{1}{K} \log_2 \overline{P}(W). \tag{2.17}$$

Pro $n$-gramové modely je možné tento vztah upravit do tvaru

$$LP = \log_2 PP = -\frac{1}{K} \sum_{i=1}^{K} \log_2 \bar{P}(w_i|w_1 w_2 ... w_{i-2} w_{i-1}). \tag{2.18}$$

Tuto úpravu je možné provést, protože pravděpodobnosti na pravé straně lze aproximovat pravděpodobnostmi poskytovanými $n$-gramovým jazykovým modelem. Pokud budeme uvažovat logaritmus perplexity $LP$ trigramového jazykového modelu platí

$$LP = \log_2 PP = -\frac{1}{K} \sum_{i=1}^{K} \log_2 \bar{P}(w_i|w_{i-2} w_{i-1}). \tag{2.19}$$

Dále lze dokázat, že pro dostatečně velký korpus platí

$$PP = 2^{LP} = 2^{H(P,\bar{P})}, \tag{2.20}$$

kde $H(P, \bar{P})$ je křížová entropie ergodického zdroje generující posloupnost $W$. Při aplikaci měřeného jazykového modelu na tento zdroj lze vyjádřit

$$H(P, \bar{P}) = -\lim_{K \to \infty} \frac{1}{K} \sum_{W} P(W) \log_2 \bar{P}(W). \tag{2.21}$$

Důležité vlastnosti perplexity:

- Perplexitu lze interpretovat jako průměrný počet slov, mezi kterými se rozhoduje akustický model při použití daného jazykového modelu.
- Perplexitu je možné definovat pro trénovací i testovací korpus.
- Perplexita závisí na použitém korpusu i na použitém jazykovém modelu. Lze proto porovnávat perplexitu dvou korpusů, použitím stejného jazykového modelu na tyto korpusy a porovnat perplexitu dvou modelů při jejich použití na stejný korpus.

I přes dobrou představu o kvalitě modelu, kterou nám perplexita poskytuje, je nutné brát její hodnoty s jistou rezervou a jistě ji nelze stavět na úroveň výsledků chybovosti při rozpoznávání řeči. [2, str. 228-231]

## 2.8 Jazykový model v rozpoznávači řeči

Jak bylo uvedeno výše, nejpodstatnějším testem pro srovnávání jazykových modelů je jejich aplikace v systému rozpoznávání řeči. Výsledkem tohoto testování jsou hodnoty *WER*, neboli word error rate. Tato veličina v sobě shrnuje tři typy chyb vznikající při rozpoznávání řeči [1].

- S: substitutions - náhrada slova

- D: deletions - odstranění slova
- I: insertions - vložení slova

Lze ji vyjádřit jako

$$WER = 100 \times \frac{S + D + I}{N}[\%], \tag{2.22}$$

potom Accuracy ($ACC$) - přesnost spočteme jako

$$ACC = 100 - WER. \tag{2.23}$$

Další významnou veličinou v rozpoznávání řeči je real-time faktor, označovaný jako $RTF$

$$RTF = \frac{P}{I}, \tag{2.24}$$

kde $P$ je doba pro zpracování promluvy o délce $I$.

Abychom mohli jazykový model použít v systému rozpoznávání řeci, musíme vytvořit slovník, ve kterém, pomocí některé z fonetických abeced, popíšeme výslovnost daného slova.

## 2.9 Vytvoření výslovnostního slovníku

Výslovnostní slovník slouží k definici výslovnosti slov obsažených v jazykových modelech. Pro zápis těchto výslovností se používají tzv. fonetické abecedy, kterými jde definovat výslovnost každého slova. Mezi nejpoužívanější fonetické abecedy pro český jazyk patří IPA a SAMPA [2]. IPA je mezinárodní fonetická abeceda, která je schopna popsat výslovnost slov nezávisle na jazyce. Lze díky tomu porovnávat různé jazyky. Fonetická abeceda SAMPA byla vytvořena, z důvodu nesnadného zápisu abecedy IPA v počítači. SAMPA provádí kódování symbolů IPA na 7-bitové tisknutelné znaky ASCII [2, str. 40].

Pro vytvoření výslovnostního slovníku je potřeba získat všechny unigramy, které obsahuje jazykový model. To můžeme udělat několika způsoby. Nejjednodušší způsob je použití balíku nástrojů pro práci s jazykovými modely, většina obsahuje i nástroj pro vytvoření slovníku. Druhá možnost je získání informací přímo z jazykového modelu formátu ARPA pomocí nástrojů pro práci s textem. Poslední možnost je výpočet unigramů z trénovacího korpusu.

Po získání seznamu unigramů je vhodné prvně přiřadit unigramům výslovnost pomocí slovníku, který má správně definované výslovnosti, pro všechna jemu známá slova a pro zbylá slova vygenerovat výslovnosti podle pravidel. Tímto způsobem se částečně omezí chyba, která by vznikla při vytvoření výslovnosti podle českých pravidel pro slovo s cízí výslovností.

# Kapitola 3

# Implementace

V této kapitole jsou popsány jednotlivé nástroje, které byly používány při zpracování zadání a krátký popis některých skriptů mnou vytvořených. Pro tvorbu jazykových modelů existují volně dostupné balíky nástrojů. Většinou se nacházejí na internetových stránkách výzkumných ústavů nebo vysokých škol. Pro moji práci jsem si vybral balík nástrojů SRILM. Další nástroje pro vytvoření jazykových modelů jsou například CMU SLM Toolkit [10], MITLM Toolkit [11] a IRSTLM toolkit [3].

## 3.1 CMU SLM toolkit a MITLM toolkit

Jedná se o méně známé balíky nástrojů, které slouží k vytváření a další práci s $n$-gramovými modely. Tyto balíky nabízejí méně funkcí a jsou používané zejména na svých domovských univerzitách. Za nejrozšířenější nástroj pro práci s $n$-gramovými modely lze považovat SRILM, kterým jsou oba inspirovány.

## 3.2 SRILM

Jedná se o balík nástrojů vyvíjený od roku 1995 centrem SRI International [7], používaný k vytvoření statistických jazykových modelů. Mezi jeho hlavní přednosti patří jednoduchá obsluha, rozsáhlé možnosti při tvorbě $n$-gramů i jazykových modelů, schopnost provádění základních testů na vytvořených modelech a kvalitní dokumentace. Tento balík lze provozovat na operačních systémech Unix i Windows. Ve své práci budu uvažovat pouze použití systému na bázi Linux.[1].

## 3.3 Instalace

V linuxovém prostředí se standartně instalace provádí příkazem

```
$sudo apt-get install nazev_baliku .
```

Tento příkaz lze použít za předpokladu, že se námi požadovaný balík nachází ve standardním repozitáři[2] nebo pokud repozitář přidáme mezi prohledávané repozitáře. V případě balíku SRILM je nutné stáhnout balík ručně ze stránek SRI International [7]. Při stažení můžete vybrat jakou verzi balíku chcete stáhnout, poté vyplníte krátký formulář s několika údaji o Vás. Při stažení potvrzujete, že souhlasíte s licenčními podminkami. Dále je

[1] Konkrétně se jednalo o Ubuntu 13.10 32-bit

[2] Repozitář je server nebo adresář se softwarovými balíčky připravenými k instalaci do systému

uveden stručný postup instalace a problémy, na které jsem narazil.

Po stažení archivu ze stránek a jeho rozbalení získáte složku s několika adresáři a soubory. Soubor s instrukcemi k instalaci se nazývá `INSTALL`. Ke správnému zavedeni je nutné ověřit přítomnost a případně doplnit následující nástroje(neúplný seznam):

- gcc 3.4.3 nebo novější
- GNU make
- Tcl toolkit 7.3 nebo novější

Dále je doporučené mít tyto nástroje (neúplný seznam). Zavedení proběhne i bez nich, ale některé nástroje balíku SRILM je používají.

- GNU awk
- gzip

Všechny potřebné nástroje jsou k dispozici zdarma a nachází se ve standardních repozitářích přítomných po instalaci Ubuntu, na problémy by jsme neměli narazit ani v případě použití jiné distribuce. Jediný nástroj, který se nenacházel v Ubuntu 13.10 32-bit po instalaci byl Tcl toolkit, u ostatních velkých distribucí operačních systémů na bázi Linux lze předpokládat stejné. Poslední krok před samotnou kompilací je úprava proměnných v souboru `/common/Makefile.machine.<platform>`. Platformu volíme podle systému, na kterém chceme SRILM provozovat.

Pokud je výše uvedené splněno, pomocí terminálu v adresáři srilm necháme proběhnout

```
$make World .
```

Během kompilace jsem narazil pouze na jediný problém, tím bylo nesprávné nalezení nástroje Tcl toolkit i přes jeho instalaci pomocí `$sudo apt-get install nazev_baliku`, ostatní byly nalezeny správně. Po upravení cesty k tomuto nástroji proběhla kompilace v pořádku, úprava byla provedena v souboru `/common/Makefile.machine.<platform>`.

Pro snažší použití doporučuji přidat nástroje do adresářů uvedených v PATH a MANPATH.

### 3.3.1 Použité nástroje

Z celého balíku SRILM jsem využíval hlavně tři následující nástroje: *ngram-count*, *ngram-merge* a *ngram*. Níže jsou uvedené parametry s krátkým vysvětlením. Kompletní popis se všemi možnostmi je přítomný v manuálu na internetových stránkách nebo příkazem `$man <jmeno_nastroje>`[3], pokud jste přidali manuálové stránky mezi adresáře v MANPATH. Uvedeny budou parametry, které byly použity pro vytvoření jazykových modelů k této práci.

#### ngram-count

Nástroj pro vytváření a úpravy souborů s $n$-gramy a vytvoření $n$-gramových jazykových modelů. $n$-gramy lze vložit už připravené v textové formě nebo je lze napočítat ze souborů s textu. Výstupem může být buď soubor s $n$-gramy nebo jazykový model ve formátu ARPA.

[3]Standardní příkaz pro vyvolání mauálových stránek v systémech založených na Linux.

- *-order n*
  Slouží k nastavení stupně $n$-gramů, jak pro získání $n$-gramů z textu, tak pro vytvoření jazykového modelu. Standardně nastaveno na $n = 3$.

- *-text soubor*
  Textový soubor, ze kterého chceme napočítat $n$-gramy. Soubor by měl obsahovat jednu větu na řádek. Standardně se přidávají tokeny označující začátek a konec věty.

- *-no-sos -no-eos*
  Vypnutí přidávání tokenů pro začátky a konce vět.

- *-read soubor*
  Načtení souboru s $n$-gramy. Soubor by měl obsahovat jeden $n$-gram na řádek následovaný počtem výskytů.

- *-write soubor*
  Zapsání $n$-gramů do souboru.

- *-write order n*
  Zapsání $n$-gramů do stupně $n$. Standardně $n = 0$, tzn. výpis $n$-gramů všech délek.

- *-sort*
  Abecední seřazení výstupu. Nutné při následném použití nástroje ngram-merge.

- *-lm soubor*
  Určení souboru se vstupním jazykovým modelem.

- *-gtnmin počet*
  $n = 1, 2, ..., 9$ označuje jaký řád $n$-gramu budeme ovlivňovat. $n$-gramy daného řádu s výskytem menším než *počet*, budou počítány s výskytem 0. Pokud je $n$ vynecháno, jsou vybrány $n$-gramy o řádů vyšším než 9. Při vytváření $n$-gramů řádu vyššího než 2, je nutné specifikovat tuto hodnotu, jinak dochází ke snížení počtu trigramů a vyšších, což může být v závislosti na aplikaci nežádoucí.

**Příklad použití**

Výpočet četnosti bigramů z textu v souboru *input*. Zapíše všechny řády napočítaných $n$-gramů ($n$=1 a $n$=2) do souboru *output*.

```
$ngram-count -text input -order 2 -write-order 0 -write output
```

### ngram-merge

Tento nástroj je vhodné použít, pokud chceme vytvořit model z více zdrojových textů a tyto texty nechceme spojovat do jednoho celku. Taková situace nastala při zpracovávání korpusu spontánních promluv, kde byly napočítány unigramy, bigramy a trigramy pro každého mluvčího (celkem jich v korpusu bylo šedesát). Pro testování bylo poté vytvořeno šedesát modelů, pro každého mluvčího jeden, a jeden model, který byl vytvořen spojením všech mluvčích. Toto spojení všech mluvčí proběhlo na bázi souborů s $n$-gramy pomocí nástroje ngram-merge. Je nutné, aby byly $n$-gramy seřazeny podle abecedy.

### Příklad použití

Všechny soubory ve složce ngrams spojí do jediného souboru all_ngrams.

```
$ngram-merge /home/user/ngrams/* > /home/user/all_ngrams
```

**ngram**

Používá se k práci s jazykovými modely založenými na $n$-gramech. Mezi úkony patří výpočet perplexity a kombinace modelů.

Pokud chceme vytvořit kombinaci obecného jazykového modelu s modelem tematicky specifickým, je vhodné specifikovat, jakou váhu přiřadíme jednotlivému modelu. Pokud bychom modely kombinovali spojením souborů s $n$-gramy použitím *ngram-merge*, malý korpus tematického jazykového modelu by nedokázal ovlivnit obecný jazykový model. Proto je nutné kombinovat modely s použitím váhy.

Pro použití jazykového modelu v systému rozpoznávání řeči je nutné vytvořit výslovnostní slovník pro slova vyskytující se v jazykové modelu. Ve výslovnostním sovníku se nacházejí slova a jejich přepis v některé z fonetických abeced. K vytvoření slovníku je potřeba získat všechna slova použitá v modelu, tedy všechny unigramy, to se dá také provést nástrojem *ngram*.

- *-lm soubor*
  Hlavní vstupní jazykový model.
- *-mix-lm soubor*
  Načtení druhého jazykového modelu pro interpolaci.
- *-lambda váha*
  Nastavené váhy pro kombinaci při použití *-mix-lm*. Standardně je váha nastavena na 0,5.
- *-write-lm soubor*
  Vystupní soubor pro nový jazykový model.
- *-write-vocab soubor*
  Zapíše slovník slov použitých v jazykovém modelu.
- *-ppl soubor*
  Spočítá logaritmické pravděpodobnosti a perplexity pro text v souboru. Má pět úrovní detailnosti výstupu, ale uvedu zde pouze první tři, které jsou nejdůležitější a nejčastěji používané. Výstup tohoto příkazu obsahuje tyto veličiny:
  - Počet slov
  - Počet vět
  - Počet OOV (slov mimo slovník)
  - Logaritmická pravěpodobnost
  - ppl Označuje hodnotu perplexity včetně tokenů pro začátek a konec věty.
  - ppl1 Hodnota perplexity počítaná bez tokenů začátku a konce slov.
    * *-debug 0* Statistika pro celý text.
    * *-debug 1* Statistika pro každou větu zvlášť.
    * *-debug 2* Statistika pro každé slovo.

## 3.4 IRSTLM

Jedná se balík nástrojů velmi podobný SRILM toolkitu. Narozdíl od balíků MITLM a CMU SLM obsahuje alternativu ke všem nástrojům použitým v této práci z balíku SRILM.

Pokud jde o vytváření jazykových modelů podobných těm, které byly vytvořeny v rámci této práce, je možné doporučit balík IRSTLM [3] jako vhodnou alternativu k balíku SRILM. Pokud je vyžadována vyšší funkcionalita, je vhodnější zvolit SRILM toolkit.

### 3.4.1 Použité nástroje

Při vytváření modelů pro testování v systému jazykového rozpoznávače bylo nutné zmenšit některé modely z důvodu softwarového omezení rozpoznávače.

#### prune-lm

Tento nástroj slouží ke zmenšení modelu snížením počtu $n$-gramů pro $n > 2$. Při vypuštění $n$-gramů s nízkou pravděpodobností můžeme podstatně zmenšit jazykový model za cenu nízkého nárůstu WER při rozpoznávání řeči. IRSTLM implementuje toto změnšení podobně k *Weighted Difference Method* popsané v práci [5].

V této práci byl navržen *weighted difference factor* $n$-gramu, který byl defnován jako

$$wd_{factor} = K \times (\log(P_P) - \log(P_{BO}), \tag{3.1}$$

kde $K$ je počet $n$-gramů po Good-Turingově odhadu, $P_P$ je původní pravděpodobnost a $P_{BO}$ je pravděpodobnost spočtená pomocí Backing-off metody.

Závěrem této práce bylo zjištení, že *Weighted Difference Method* je efektivnější než původně používaná metoda *cut-off* při srovnání na perplexitě a WER.

#### Příklad použití

```
prune-lm -threshold=1e-7 model_puvodni model_oriznuty
```

Hodnota *1e-7* byla určena experimentálně pro jazykový model s 340 000 unigramů z Českého národního korpusu. Pro tuto hodnotu byl použitý jazykový rozpoznávač schopen vykonat testování s daným modelem.

## 3.5 Čištění korpusu

Jak bylo uvedeno dříve, číštění textu je nedílnou součástí vytváření jazykových modelů. V této kapitole jsou uvedeny nástroje, které byly použity k čištění a úpravě textu a ukázky jejich použití. Jedná se primárně o programovací jazyk Perl a nástroj sed.

### 3.5.1 Perl

Tento programovací jazyk vytvořil Larry Wall okolo roku 1987. Původním záměrem bylo zjednodušit psaní unixových skriptů a zpracování textu. Mezi výhody Perlu patří přítomnost ve většíně Linuxových distribucí, snadná syntaxe a práce s regulárními výrazy.

Perl byl v této práci použit především pro vytváření výslovnostních slovníků.

### 3.5.2 sed

Tento nástroj vytvořil Lee E. McMahon okolo roku 1974. Dnes je standardně přítomný ve většíně Linuxových distribucí, ale lze jej používat i pod jinými operačními systémy. Zkratka *sed* označuje **s**tream **ed**itor, toto slovní spojení je možné přeložit jako editor proudu. Protože pracuje po řádcích, lze s ním editovat velmi efektivné i velké soubory.

Pomocí sed byly vytvořeny skripty pro čištění a úpravu textu. Tyto skripty se skládají

z mnoha jednotlivých spuštění sed s různými substitucemi a úpravami na daný soubor textu.

**Příklady použití**

```
sed -i 's/([[:alpha:] ,]*)//g' soubor
```

- přepínač ***-i*** říká sed-u provádět změny do stejného souboru
- konstrukce ***'s/aaa/bbb/g'*** znamená najít v textu posloupnost "aaa"a nahradit ji "bbb", *g* značí provést tuto substituci na všechny výskyty v souboru
- ***[]*** do hranatých závorek se píší výčty znaků k vyhledání například *[a-z]* nebo *[aeiou]*, *[[:alpha:]]* postihne všechna písmena
- ******* značí jakékoliv množství opakování

```
sed -i 's/[a-ž]*()//g' soubor
```

- Pokud v mluvě bylo některé slovo přerušeno, například z důvodu přeřeku, bylo označeno pomocí kulatých závorek. Nedokončené slovo například *autobus* jako *autob()* a podobně. Protože taková slova nejsou platná pro trénování, bylo nutné je z textu odstranit. K tomu slouží tento příkaz.

# Kapitola 4

# Vytvořené modely a testování

Cílem této práce bylo vytvoření tematicky specifických jazykových modelů. Pro vytvoření těchto modelů bylo použito několik zdrojů textu a dříve vytvořené modely z Českého národního korpusu. Tematicky specifické jazykové modely byly vytvářeny kombinováním dvou modelů. Jako základní model jsem použil modely vytvořené z Českého národního korpusu. Jednalo se o bigramové a trigramové modely.

## 4.1 Modely CNK

Tyto modely byly vytvořeny v rámci práce [4], ve které byly porovnávány výsledky modelů vytvořených z veřejně dostupných zdrojů. K vytvoření modelů posloužily dva různé korpusy, korpus *Czech Web 1T 5-gram*, ze kterého vznikly modely *WEB1T* a korpus *SYN2006PUB 5-gram*. Z druhého zmiňovaného korpusu byly vytvořeny modely *CNC*, v této práci označované jako *CNK*. Jednalo se o modely rozsahu 60 tisíc, 120 tisíc, 180 tisíc, 240 tisíc a 340 tisíc unigramů. Konkrétní parametry jednotlivých modelů jsou zobrazeny v tabulce (4.1). Jako druhý model při kombinování sloužily modely získané z korpusů uvedených v této práci.

| **model** | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|
| cnk60k | 60 002 | 28 273 982 | 17 976 344 |
| cnk120k | 120 002 | 36 403 752 | 19 501 433 |
| cnk180k | 180 002 | 40 319 972 | 20 028 748 |
| cnk240k | 240 002 | 42 557 801 | 20 268 094 |
| cnk340k | 340 002 | 44 602 819 | 20 443 214 |

Tabulka 4.1: Tabulka modelů CNK

## 4.2 Použité korpusy

Jako korpusy pro vytváření tematicky specifických jazykových modelů byly použity následující zdroje:

- Kniha Technologie hlasových komunikací - dále označeno jako korpus *THKBK*.

  Modely vytvořené z tohoto korpusu by mohly být použity pro přepisy přednášek nebo při přepisech korpusu PONDELKY.

- Přepis spontánních promluv - dále označeno jako korpus *NCCCz*.

  Tento korpus nebyl tematicky jednotný jako zbylé dva. Jeho tematičnost nebo jedinečnost spočívá v zachycení slangu a obecné češtiny. Při kombinaci s obecnými modely by mohl být vhodný pro rozpoznávání běžné mluvy.

- Přepisy prezentací studentů postgraduálního studia - dále označeno jako korpus *PONDELKY*.

  Prezentace studentů byly na tři různá témata. Podle nich byl korpus rozdělen a vznikly tak menší korpusy. Nejobsáhleší z nich je korpus Zpracování řeči. Naopak korpus Teorie signálů je nejmenší, vznikl pouze ze třech prezentací, a tudíž je vhodný spíše k testování, než výraznému doplnění jiného modelu.

  - Biologické signály
  - Zpracování řeči
  - Teorie signálů

### 4.2.1 Konkrétní úpravy korpusů

V této části budou popsány úpravy, které byly použity přo přípravu jednotlivých korpusů k vytvoření jazykových modelů.

#### Korpus THKBK

Tento korpus byl složen ze zdrojových kódů sázecího softwaru LaTeX. Kromě odstranění tabulek, vzorců a grafiky bylo nutné vybrat pouze text a odstranit veškeré formátování. K tomu byl vytvořen skript, který postihoval všechny požadované problémy.

```
sed -i 's/\\item//g' $output;
sed -i 's/,//g' $output;
sed -i 's/\s\{2,\}\([[:upper:]]\)/\n\1/g' $output;

sed -i 's/\\myindex\[\([[:alpha:] \,\!]*\)\]{[[:alpha:] \,\!]*}/\1/g' $output;
sed -i 's/} \([[:upper:]]\)/}\n\1/g' $output;
sed -i 's/^[[:upper:]]*\.//g' $output;
```

Obrázek 4.1: Část scriptu pro zpracování korpusu knihy Technologie h.k.

#### Korpus PONDELKY

Tento korpus byl rozdělen na tři různé tematické celky podle témat, jimiž se prezentující zabývali. Po tomto rozdělení byly odstraněny synchronizační časové značky.

```
<Sync time="0"/>        <Sync time="8.738"/>    czdsp002000      [other-]
<Sync time="8.738"/>    <Sync time="10.181"/>   czdsp002001      tak můžem začít
```

Obrázek 4.2: Ukázka synchronizačních časových značek pro jeden z přepisů

Po odstranění těchto značek byly smazány neřečové události označené v textu [*udalost*] a nedokončená slova označená *autob*(). Další úpravou bylo odstranění $*pismeno*, které označuje jednotlivě vyslovené písmeno (například $é ve zkratce EEG). Posledním krokem bylo spojení jednotlivých promluv, které byly na řádcích po jedné, do jediné promluvy. To bylo učiněno z důvodu nejasného konce vět v případě spontánních promluv.

**Korpus NCCCz**

Postup při úpravě toho korpusu se v mnohém shoduje s úpravami provedenými na korpusu *PONDELKY*. Jednalo se také o promluvy uložené na řádcích po jedné a také se zde vyskytovala nedokončená slova označená *autob*(), takže čištění probíhalo podobně. Nakonec byly promluvy spojeny jako v případě korpusu *PONDELKY*.

Posledním krokem, společným pro všechny korpusy, bylo jejich rozdělení na trénovací a testovací část v poměru 70 : 30. Větší množství dat, oproti běžným zvyklostem, bylo k trénovací množině přidáno z důvodu velikosti korpusů. U větších korpusů by bylo možné použít dělení 90 : 10 a zlepšit tím vytvořené modely.

V tabulce (4.2) je vidět srovnání velikostí jednotlivých jazykových modelů vytvořených z dostupných korpusů. Velikosti modelů *PONDELKY* jsou znatelně různé, je to z důvodu množství prezentací vyskytujících se v korpusu. Jak lze odvodit, nejvíce zaznamenaných prezentací bylo na téma Zpracování řeči. Naopak pro téma Teorie signálu bylo zaznamenáno pouze několik prezentací. Model *NCCCz_ALLSPK* ukazuje hodnoty pro model vytvořený ze všech mluvčí obsažených v *NCCCz*. Hodnoty pro jednotlivé mluvčí tohoto korpusu byly přibližně 10× nižší v každé kategorii.

Pokud se podíváme na nárůst bigramů a trigramů pro jednotlivé modely, jde vidět, že počet bigramů a trigramů u modelů *PONDELKY_BIO, PONDELKY_REC, PONDELKY_SIG* a *NCCCz_ALLSPK* není konstatní, ale zvyšuje se s počtem unigramů. To bylo způsobeno spojením promluv. Model *THKBK* je svým nárůstem nejpřirozenější.

| model | unigramy | bigramy | trigramy |
|---|---|---|---|
| THKBK | 5 786 | 16 529 | 18 081 |
| PONDELKY_BIO | 5 364 | 22 013 | 29 075 |
| PONDELKY_REC | 9 269 | 43 897 | 60 062 |
| PONDELKY_SIG | 1 509 | 4 797 | 5 851 |
| NCCCz_ALLSPK | 21 816 | 113 329 | 197 765 |

Tabulka 4.2: Srovnání velikosti modelů vytvořených z dostupných korpusů

## 4.3 Vytvoření modelu

Po vyčištění zdrojového textu je možné, buď napočítat $n$-gramy, nebo přímo vytvořit jazykový model. Z důvodu častých úprav jsem ve většině případů počítal nejdříve soubor $n$-gramů a z nich poté vytvářel jazykové modely. Tento postup je také doporučován vývojáři balíku SRILM při vytváření rozsáhlých jazykových modelů. Pro napočítání $n$-gramů je nutné specifikovat několik parametrů. Hlavním je určení stupně $n$-gramu, a pokud plánujeme spojovat více souborů s $n$-gramy do jediného, tak i řazení. Seřadit soubor lze i dodatečně. Jestliže máme připravený soubor s $n$-gramy, můžeme vytvořit jazykový model. Ten lze následně otestovat nebo kombinovat s jiným modelem a upravit tak například obecný jazykový model přidáním tematicky specifického jazykového modelu.

Rychlost kombinování modelů se odvíjí od jejich velikosti. Většina[4] výpočetních operací spojené s touto prací byla prováděna ve virtualizovaném prostředí Ubuntu 13.10 32-bit pomocí programu VMware [12]. V počítači byl instalovaný procesor Intel i5-3210M a 12GB operační paměti. Při použití nevirtualizovaného systému by bylo dosaženo znatelně lepších výsledků. V tabulce(4.3) vidíme rychlost kombinování dvou modelů. Kombinovaly se modely vytvořené z Českého národního korpusu [6] s modelem získaným z prezentací studentů na téma zpracování biologických signálů.

| **model** | **unigramy** | **bigramy** | **trigramy** | **čas [s]** |
|---|---|---|---|---|
| PONDELKY_BIO | 5 364 | 22 013 | 29 075 | - |
| cnk60k | 60 002 | 28 273 982 | 17 976 344 | 156 |
| cnk120k | 120 002 | 36 403 752 | 19 501 433 | 207 |
| cnk180k | 180 002 | 40 319 972 | 20 028 748 | 216 |
| cnk240k | 240 002 | 42 557 801 | 20 268 094 | 227 |
| cnk340k | 340 002 | 44 602 819 | 20 443 214 | 252 |

Tabulka 4.3: Srovnání velikosti modelů a čas kombinací

## 4.4 Testování modelů

Jak již bylo několikrát uvedeno v této práci, pro testování jazykových modelů bylo použito dvou metod. První metoda spočívá ve vypočtení perplexity při použití několika modelů na stejný testovací korpus. Tímto způsobem lze například určit přínos většího (s vyšším počtem unigramů) modelu, a určit tak optimální velikost modelu. Druhý způsob je použití jazykového modelu při rozpoznávání, a tím zjistit reálný přínos tohoto modelu. K dalším možnostem srovnávání jazykových modelů patří porovnávání podle *OOV*. *OOV* je zkratka z aglického *Out Of Vocabulary*, do češtiny přeloženo jako *mimo slovník*. Tedy porovnat, jak přesně vystihuje daný model téma testovacího korpusu a při vytvoření několika velikostí stejného modelu přínos jeho zvětšování ke snížení *OOV*. Kombinací těchto testů lze získat model přesně vyhovující našim požadavkům.

Protože je testování v rozpoznávači časově náročnější, primární testování probíhá na úrovní perplexit. K testování na rozpoznávači je potřeba vytvořit výslovnostní slovník pro daný model. K tomu byl vytvořen skript v Perlu, který spojuje dříve vytvořený nástroj `transc` pro převod textu do IPA CTU a knihovnu `phones_conv.pm` pro další konverzi mezi

[4]Práce na modelech NCCCz byly jako jediné prováděny na školním clusteru.

několika abecedami výslovnosti. Použitý systém rozpoznávání řeči pracoval se slovníky ve formátu HTK.

## 4.5 Výsledky testování

Před uplatněním vytvořených modelů zbývá poslední krok, a tím je jejich otestování. Ve druhé kapitole byla vysvětlena veličina perplexita, kterou lze použít ke srovnání jazykových modelů odděleně od systému rozpoznávání řeči. Touto veličinou budeme porovnávat většinu modelů. Jako druhé hlavní hledisko pro porovnávání bude sloužit veličina OOV. Pro obě uvedené veličiny budeme sledovat jejich změny při kombinaci s modely obecnými z Českého národního korpusu. Pro kombinování modelů byly použity pevně nastavené váhy o hodnotách $0.25, 0.5$ *a* $0.75$. Tyto hodnoty označují váhu pro obecný model, tím byly vždy modely z Českého národního korpusu.

Modely vytvořené z korpusu spontánních promluv budou otestovány také v systému rozpoznávání řeči, kde se ukáže přínos tematického jazykového modelu oproti modelu obecnému.

### 4.5.1 Modely THKBK

Z tohoto korpusu THKBK vznikl jeden model, jehož velikost je uvedena v tabulce(4.4).

| **model** | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|
| THKBK | 5 786 | 16 529 | 18 081 |

Tabulka 4.4: Velikost modelu pro korpus THKBK

Kombinací tohoto modelu s modely z CNK (uvedené v tabulce 4.1) vzniklo dalších 15 modelů, které odpovídají dříve zmíněnému postupu pro kombinování.

| **váha 0.25** | **váha 0.5** | **váha 0.75** |
|---|---|---|
| cnk60_THKBK_25 | cnk60_THKBK_50 | cnk60_THKBK_75 |
| cnk120_THKBK_25 | cnk120_THKBK_50 | cnk120_THKBK_75 |
| cnk180_THKBK_25 | cnk180_THKBK_50 | cnk180_THKBK_75 |
| cnk240_THKBK_25 | cnk240_THKBK_50 | cnk240_THKBK_75 |
| cnk340_THKBK_25 | cnk340_THKBK_50 | cnk340_THKBK_75 |

Tabulka 4.5: Modely vzniklé kombinací CNK a THKBK

Tyto modely byly otestovány na testovacím korpusu knihy. V tabulce(4.6) jsou zachyceny perplexity pro jednotlivé modely.

Jak vidíme nejnižší perplexity dosáhl samotný model *THKBK*, což je očekávaný výsledek. Tento model by vytvořen z trénavacího korpusu knihy, proto jejímu stylu odpovídá nejvíce. Tento model má ovšem vysoké OOV, proto by při použití v rozpoznávači nedosáhl tak dobrých výsledků. Co se OOV týká, zajímavé srovnání je například u modelů *cnk340k* a *cnk60k+THKBK*. Tyto modely mají téměř totožné množství slov mimo slovník. Model vytvořený námi je ale mnohem menší, takže dosahuje $5\times$ nižší perplexity. Další modely, které stojí za povšimnutí je skupina modelů *cnk180k+THKBK*. Je vidět,

že při testování větších modelů již OOV klesá pomalu, proto je pro výraznější snížení nutné použít mnohem většího modelu. Očekávaný byl i poslední výsledek. Skupina modelů *cnk340k+THKBK* má nejnižší OOV a nejvyšší perplexitu. Jelikož perplexita určuje počet slov, mezi kterými by se musel akustický model dále rozhodovat, je vhodné zvážit přínost maximálního snížení OOV za cenu zvýšené perplexity. Pokud by byla slova akusticky velmi odlišná, tak by nám 2× zvýšená perplexita by nám oproti modelům skupiny *cnk60k+THKBK* nemusela vadit, protože by slova byla vyloučena akustickým modelem.

Pokud se podíváme na vliv váhování modelů na perplexitu, vidíme nárůst od 1.5% do 3% pro váhu 0.5 a nárůst okolo 20% při váze 0.75. Pokud bychom měli rozhodnout o váhování pouze z perplexity a OOV je zřejmé, že nejlepšího výsledku by vždy dosáhl model s nejnižší vahou. Kdybychom vzali tuto úvahu do extrému, a obecnému modelu bychom tak přiřadili téměř nulovou váhu, model by měl perplexitu nejnižší při zachování OOV. Je nutné uvědomit si, že v reálné aplikaci nebude taková shoda, které zde byla vytvořena rozdělením jednoho korpusu na korpus trénovací a testovací. Pro zvolení ideálního váhování je nutné tedy uvážit konkrétní aplikaci.

| **Model** | **Testovací korpus(počet slov)** | **Perplexita** | **OOV[%]** |
|---|---|---|---|
| THKBK | kniha_test(10 840) | 869.29 | 16.04 |
| cnk60k | kniha_test(10 840) | 2 147.06 | 19.03 |
| cnk120k | kniha_test(10 840) | 3 044.22 | 12.73 |
| cnk180k | kniha_test(10 840) | 3 838.23 | 9.82 |
| cnk240k | kniha_test(10 840) | 4 370.33 | 8.22 |
| cnk340k | kniha_test(10 840) | 5 220.92 | 6.75 |
| cnk60k+THKBK_25 | kniha_test(10 840) | 1 020.77 | 6.86 |
| cnk60k+THKBK_50 | kniha_test(10 840) | 1 050.05 | 6.86 |
| cnk60k+THKBK_75 | kniha_test(10 840) | 1 296.05 | 6.86 |
| cnk120k+THKBK_25 | kniha_test(10 840) | 1 182.34 | 4.83 |
| cnk120k+THKBK_50 | kniha_test(10 840) | 1 199.06 | 4.83 |
| cnk120k+THKBK_75 | kniha_test(10 840) | 1 456.18 | 4.83 |
| cnk180k+THKBK_25 | kniha_test(10 840) | 1 311.47 | 3.82 |
| cnk180k+THKBK_50 | kniha_test(10 840) | 1 331.93 | 3.82 |
| cnk180k+THKBK_75 | kniha_test(10 840) | 1 614.07 | 3.82 |
| cnk240k+THKBK_25 | kniha_test(10 840) | 1 387.85 | 3.27 |
| cnk240k+THKBK_50 | kniha_test(10 840) | 1 413.96 | 3.27 |
| cnk240k+THKBK_75 | kniha_test(10 840) | 1 713.76 | 3.27 |
| cnk340k+THKBK_25 | kniha_test(10 840) | 1 502.88 | 2.65 |
| cnk340k+THKBK_50 | kniha_test(10 840) | 1 551.85 | 2.65 |
| cnk340k+THKBK_75 | kniha_test(10 840) | 1 900.90 | 2.65 |

Tabulka 4.6: Srovnávací tabulka pro modely testované na testovacím korpusu kniha

### 4.5.2 Modely NCCCz

Z NCCCz byly vytvořeny jazykové modely pro jednotlivé mluvčí. Dále jeden model pro všechny mluvčí a kombinace modelu pro všechny mluvčí s modely z Českého národního korpusu. Tyto modely jsou srovnány nejen z pohledu perplexit a *OOV*, ale také v systému rozpoznávání řeči, kde měřená veličina byla *WER*.

Tabulka(4.7) shrnuje velikosti modelů pro jednotlivé mluvčím z NCCCz.

| **model** | **unigramy** | **bigramy** | **trigramy** | **model** | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|---|---|---|---|
| 1 | 1 089 | 2 928 | 3 610 | 31 | 1 427 | 4 222 | 5 192 |
| 2 | 1 120 | 3 039 | 3 841 | 32 | 1 243 | 3 147 | 3 860 |
| 3 | 956 | 2 412 | 2 889 | 33 | 843 | 2 090 | 2 585 |
| 4 | 1 332 | 3 821 | 4 763 | 34 | 982 | 3 029 | 3 923 |
| 5 | 1 444 | 4 411 | 5 604 | 35 | 1 022 | 3 954 | 5 582 |
| 6 | 745 | 1 746 | 2 046 | 36 | 940 | 2 939 | 3 908 |
| 7 | 626 | 1 639 | 1 948 | 37 | 1 107 | 3 095 | 3 807 |
| 8 | 2 165 | 6 443 | 7 903 | 38 | 1 437 | 4 448 | 5 627 |
| 9 | 661 | 1 532 | 1 794 | 39 | 555 | 1 199 | 1 293 |
| 10 | 1 679 | 5 078 | 6 447 | 40 | 1 530 | 4 517 | 5 754 |
| 11 | 890 | 2 329 | 2 855 | 41 | 907 | 2 435 | 3 059 |
| 12 | 1 103 | 3 090 | 3 899 | 42 | 804 | 1 801 | 2 009 |
| 13 | 1 450 | 4 868 | 6 461 | 43 | 1 323 | 3 556 | 4 358 |
| 14 | 1 014 | 3 174 | 4 107 | 44 | 1 216 | 4 002 | 5 685 |
| 15 | 614 | 1 634 | 1 922 | 45 | 867 | 2 130 | 2 472 |
| 16 | 977 | 2 827 | 3 543 | 46 | 1 255 | 3 064 | 3 571 |
| 17 | 1 657 | 4 703 | 5 788 | 47 | 745 | 1 889 | 2 187 |
| 18 | 657 | 1 526 | 1 762 | 48 | 1 114 | 3 150 | 3 835 |
| 19 | 1 152 | 3 131 | 3 913 | 49 | 1 279 | 3 672 | 4 605 |
| 20 | 1 472 | 4 025 | 5 092 | 50 | 954 | 2 594 | 3 193 |
| 21 | 858 | 2 111 | 2 526 | 51 | 800 | 2 132 | 2 613 |
| 22 | 1 188 | 3 547 | 4 455 | 52 | 1 260 | 3 730 | 4 681 |
| 23 | 1 417 | 4 156 | 5 038 | 53 | 957 | 2 393 | 2 895 |
| 24 | 551 | 1 222 | 1 394 | 54 | 895 | 2 403 | 2 972 |
| 25 | 1 442 | 4 584 | 6 032 | 55 | 949 | 2 440 | 2 996 |
| 26 | 1 549 | 4 418 | 5 584 | 56 | 1 444 | 3 412 | 3 855 |
| 27 | 588 | 1 442 | 1 761 | 57 | 867 | 2 207 | 2 627 |
| 28 | 1 332 | 4 073 | 5 223 | 58 | 1 048 | 2 495 | 2 838 |
| 29 | 969 | 2 626 | 3 235 | 59 | 1 745 | 4 909 | 6 000 |
| 30 | 802 | 2 395 | 3 158 | 60 | 908 | 2 355 | 2 832 |

Tabulka 4.7: Velikost modelů jednotlivých mluvčí z NCCCz

Při pohledu na tabulku velikostí modelů si můžeme všimnout, že někteří mluvčí mají výrazně nižší počet unigramů, to může mít dvě různé příčiny: mluvčí mluvil méně nebo používal menší slovní zásobu. Kombinace obou těchto jevů je s největší pravděpodobností příčinou nejnižších hodnot v okolí 600 unigramů.

Pro lepší představu o modelech jako celku byla vytvořena tabulka(4.8), která zobrazuje průměrnou hodnotu, směrodatnou odchylku, minimální a maximální hodnotu pro jednotlivé $n$-gramy.

| - | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|
| **AVG** | 1 099 | 3 072 | 3 823 |
| **STD** | 334 | 1 125 | 1 489 |
| **MIN** | 551 | 1 199 | 1 293 |
| **MAX** | 2 165 | 6 443 | 7 903 |

Tabulka 4.8: Statistické hodnoty modelů jednotlivých mluvčí

K dalšímu testování byl vytvořen model *NCCCz_ALLSPK* spojením všech mluvčí. Ten byl dále kombinován s modely vytvořenými z Českého národního korpusu. Z důvodu omezení u rozpoznáváče řeči musely být modely CNK zmenšeny. Konkrétní zmenšení vyjařují čísla v názvu modelu ve tvaru 1e-6 nebo 1e-7. Číslo před modelem vyjadřuje stupeň modelu, 2 je bigramový a 3 trigramový. Jak vidíme v tabulce(4.9), modely vytvořené pro rozpoznávání jsou podstaně menší než původní modely CNK.

| **model** | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|
| 2_NCCCz_ALLSPK | 21 816 | 113 329 | - |
| 3_NCCCz_ALLSPK | 21 816 | 113 329 | 197 765 |
| 2_cnk180k+NCCCz_ALLSPK | 186 201 | 352 835 | - |
| 2_cnk340k_1e6 | 340 002 | 230 801 | - |
| 2_cnk340k_1e6+NCCCz_ALLSPK | 344 559 | 331 214 | - |
| 2_cnk340k_1e7 | 340 002 | 1 789 273 | - |
| 2_cnk340k_1e7+NCCCz_ALLSPK | 344 559 | 1 872 707 | - |
| 3_cnk340k_1e6 | 340 002 | 167 598 | 20 423 |
| 3_cnk340k_1e7+NCCCz_ALLSPK | 344 559 | 271 140 | 217 301 |
| 3_cnk340k_1e7 | 344 559 | 1 684 619 | 405 803 |

Tabulka 4.9: Velikost modelů CNK spojených s modelem NCCCz_ALLSPK

Hodnoty v tabulce(4.10) byly naměřeny pro promluvy vícero mluvčích. Tyto první výsledky v systému rozpoznávání řeči ukazují zlepšení přibližně 10% oproti obecným modelům. Protože je rozpoznávač stále ve vývoji, jsou hodnoty *WER* velmi vysoké. Lze ale předpokládat, že při zlepšení akustické části rozpoznávání se bude rozdíl mezi obecnými modely a modely vytvořenými v této práci dále zvětšovat. Pokud se podíváme na to, jak perplexita ukazuje kvalitu modelu, můžeme říct, že 20% nárůst perplexity vedl na přibližně 1%-ní zhoršení *WER*.

V této tabulce(4.10) můžeme také porovnat přínos trigramového modelu oproti modelu bigramovému(skupina modelů *2_cnk340k_1e-6+NCCCz* a *3_cnk340k_1e-6+NCCCz*). Pokud se podíváme na perplexity, můžeme říct, že modely jsou prakticky totožné. Rozpoznávání řeči přineslo použití trigramových modelů snížení *WER* o téměř 3%.

Otestování jazykového modelu v rozpoznávání řeči je časově mnohem náročnější než výpočet perplexit OOV. Námi použitý rozpoznávač prováděl testování s real-time faktorem RTF$\doteq$ 0.48, tzn. že jedna hodina promluv se testovala přibližně 30minut.

| Model | Test. korpus(slov) | PPL | OOV[%] | WER[%] |
|---|---|---|---|---|
| 2_cnk180k+NCCCz_25 | ncccz_test(122 224) | 364.57 | 2.39 | 75.07 |
| 2_cnk180k+NCCCz_50 | ncccz_test(122 224) | 414.76 | 2.39 | 76.12 |
| 2_cnk180k+NCCCz_75 | ncccz_test(122 224) | 545.15 | 2.39 | 78.24 |
| 2_cnk340k_1e-6 | ncccz_test(122 224) | 2 796.61 | 3.36 | 85.79 |
| 2_cnk340k_1e-6+NCCCz_25 | ncccz_test(122 224) | 429.62 | 1.83 | 75.48 |
| 2_cnk340k_1e-6+NCCCz_50 | ncccz_test(122 224) | 502.81 | 1.83 | 76.77 |
| 2_cnk340k_1e-6+NCCCz_75 | ncccz_test(122 224) | 686.57 | 1.83 | 78.63 |
| 2_cnk340k_1e-7 | ncccz_test(122 224) | 2 069.39 | 3.36 | 84.98 |
| 2_cnk340k_1e-7+NCCCz_25 | ncccz_test(122 224) | 407.53 | 1.83 | 75.06 |
| 2_cnk340k_1e-7+NCCCz_50 | ncccz_test(122 224) | 470.26 | 1.83 | 76.23 |
| 2_cnk340k_1e-7+NCCCz_75 | ncccz_test(122 224) | 627.73 | 1.83 | 77.95 |
| 3_cnk340k_1e-6 | ncccz_test(122 224) | 3 052.66 | 3.36 | 86.18 |
| 3_cnk340k_1e-6+NCCCz_25 | ncccz_test(122 224) | 432.09 | 1.83 | 72.21 |
| 3_cnk340k_1e-6+NCCCz_50 | ncccz_test(122 224) | 506.81 | 1.83 | 73.83 |
| 3_cnk340k_1e-6+NCCCz_75 | ncccz_test(122 224) | 702.02 | 1.83 | 76.33 |
| 3_cnk340k_1e-7 | ncccz_test(122 224) | 2 066.61 | 3.36 | 85.26 |

Tabulka 4.10: Naměřené hodnoty pro kombinace modelů CNK s NCCCz_ALLSPK(označen pouze NCCCz)

Další testování v rozpoznávači probíhalo na promluvách jednotlivých mluvčí s použitím třech skupin modelů. Toto testování nebylo provedeno pro všechny mluvčí, někteří byly vynecháni z důvodu nižší kvality záznamu jejich promluv. První skupinou byly modely pro jednotlivé mluvčí (tabulka 4.7) označené jako *Modely A*, dále model *3_cnk340k_1e-7*(tabulka 4.10), označený jako *Model B* a poslední byly modely vytvořené kombinací modelů jednotlivých mluvčí a modelu *3_cnk340k_1e-7* s vahou 0.75 pro obecný model, ty jsou označeny *Modely C*.

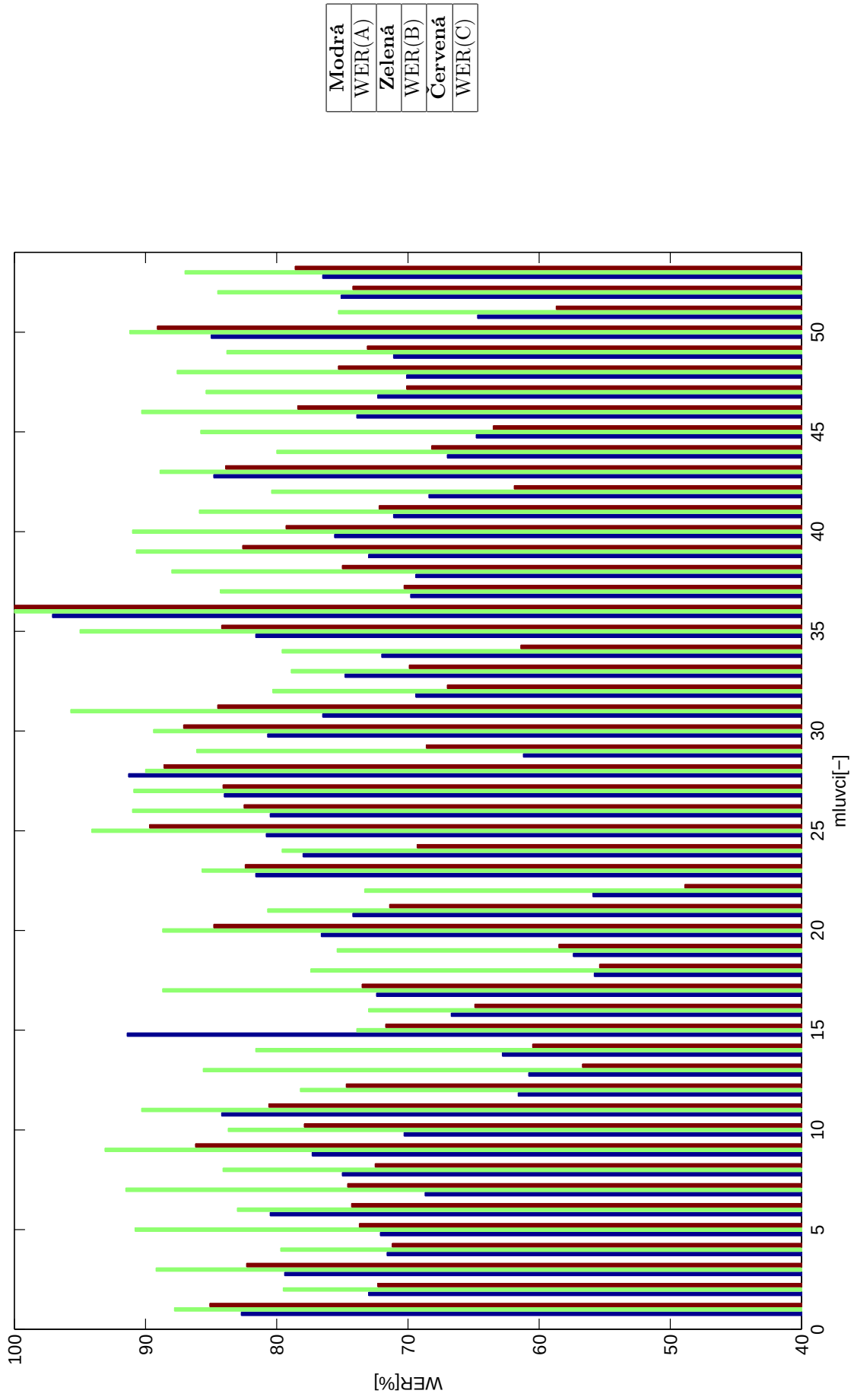
Jak vidíme v tabulce(4.11), tak jedno rozpoznávání nebylo úspěšné(mluvčí 42), *WER* vyšlo 100%. V jednom případě byl model vytvořený pro mluvčího podstatně horší než obecný model(mluvčí 19). U mluvčí 17, 24, 23, 27 a 58 byla hodnota *WER* pro kombinovaný model nižší než 60%. Ze statistických hodnot lze odvodit, že modely vytvořené kombinací modelů mluvčí a obecného modelu jsou nepatrně horší(vyšší průměrná hodnota a vyšší směrodatná odchylka *WER*) než modely jednotlivých mluvčí. Toto zjištění odpovídá předpokladu, že jazykový model vytvořený na míru jednomu mluvčímu, pro daný typ promluvy, musí být lepší než obecnější model.

| | | **Modely A** | | | **Model B** | | | **Modely C** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| mluvčí | korpus | OOV | PPL | WER | OOV | PPL | WER | OOV | PPL | WER |
| [-] | [-] | [%] | [-] | [%] | [%] | [-] | [%] | [%] | [-] | [%] |
| 1 | 2 271 | 23.0 | 143.6 | 82.7 | 3.7 | 1877.3 | 87.8 | 3.0 | 695.2 | 85.1 |
| 4 | 2 984 | 21.3 | 172.8 | 73.0 | 3.2 | 1866.9 | 79.5 | 2.5 | 766.1 | 72.3 |
| 5 | 3 046 | 20.4 | 154.6 | 79.4 | 3.3 | 1706.3 | 89.2 | 2.5 | 715.6 | 82.3 |
| 6 | 1 198 | 27.4 | 120.9 | 71.6 | 3.4 | 2308.4 | 79.7 | 3.1 | 852.8 | 71.2 |
| 7 | 766 | 28.2 | 127.0 | 72.1 | 4.4 | 2878.3 | 90.8 | 3.8 | 878.4 | 73.7 |
| 8 | 4 610 | 21.5 | 212.6 | 80.5 | 3.0 | 2520.7 | 83.0 | 2.7 | 980.5 | 74.3 |
| 10 | 3 961 | 19.9 | 174.4 | 68.7 | 2.6 | 1640.3 | 91.5 | 2.3 | 685.8 | 74.6 |
| 11 | 1 885 | 24.8 | 134.4 | 75.0 | 4.2 | 1863.6 | 84.1 | 3.8 | 699.3 | 72.5 |
| 12 | 1 871 | 23.9 | 124.3 | 77.3 | 3.5 | 2962.5 | 93.1 | 3.2 | 778.6 | 86.2 |
| 13 | 3 782 | 18.4 | 152.9 | 70.3 | 3.5 | 2469.8 | 83.7 | 3.1 | 757.1 | 77.9 |
| 15 | 833 | 29.1 | 142.3 | 84.2 | 3.0 | 2669.1 | 90.3 | 2.9 | 1197.6 | 80.6 |
| 16 | 1 742 | 21.6 | 106.5 | 61.6 | 3.2 | 2059.0 | 78.2 | 2.7 | 519.7 | 74.7 |
| 17 | 2 918 | 23.7 | 164.4 | 60.8 | 4.8 | 2760.8 | 85.6 | 3.7 | 872.8 | 56.7 |
| 18 | 752 | 32.8 | 136.5 | 62.8 | 6.1 | 2605.1 | 81.6 | 4.8 | 1081.3 | 60.5 |
| 19 | 2 405 | 27.2 | 164.3 | 91.4 | 3.3 | 1900.8 | 73.9 | 3.0 | 868.0 | 71.7 |
| 20 | 3 027 | 24.9 | 159.4 | 66.7 | 4.1 | 1851.2 | 73.0 | 3.9 | 816.3 | 64.9 |
| 22 | 2 467 | 19.1 | 141.9 | 72.4 | 2.6 | 1883.4 | 88.7 | 2.3 | 648.2 | 73.5 |
| 23 | 2 346 | 23.9 | 174.5 | 55.8 | 2.8 | 1972.9 | 77.4 | 2.2 | 858.3 | 55.4 |
| 24 | 670 | 29.3 | 138.9 | 57.4 | 3.3 | 2468.9 | 75.4 | 2.8 | 792.7 | 58.5 |
| 25 | 3 444 | 20.2 | 153.7 | 76.6 | 3.5 | 1788.0 | 88.7 | 2.6 | 662.8 | 84.8 |
| 26 | 2 856 | 25.4 | 151.7 | 74.2 | 4.4 | 2683.5 | 80.7 | 3.5 | 868.1 | 71.4 |
| 27 | 735 | 26.5 | 90.3 | 55.9 | 2.6 | 2225.5 | 73.3 | 2.3 | 591.6 | 48.9 |
| 28 | 2 865 | 20.3 | 141.2 | 81.6 | 3.2 | 1990.3 | 85.7 | 2.8 | 686.3 | 82.4 |
| 29 | 1 715 | 27.3 | 148.1 | 78.0 | 2.7 | 2053.7 | 79.6 | 2.6 | 840.2 | 69.3 |
| 30 | 1 747 | 20.3 | 105.1 | 80.8 | 2.3 | 2074.5 | 94.1 | 2.2 | 561.2 | 89.7 |
| 31 | 2 052 | 25.3 | 176.4 | 80.5 | 3.5 | 2479.1 | 91.0 | 3.1 | 940.4 | 82.5 |
| 32 | 2 491 | 27.7 | 137.0 | 84.0 | 5.1 | 2111.8 | 90.9 | 4.2 | 835.1 | 84.1 |
| 33 | 1 533 | 29.0 | 129.2 | 91.3 | 3.9 | 2311.6 | 90.0 | 3.6 | 843.6 | 88.6 |
| 34 | 1 551 | 22.5 | 143.0 | 61.2 | 3.7 | 2476.2 | 86.1 | 3.4 | 752.4 | 68.6 |
| 35 | 2 852 | 15.8 | 105.6 | 80.7 | 2.0 | 1653.6 | 89.4 | 1.2 | 425.0 | 87.1 |
| 36 | 1 967 | 17.8 | 107.8 | 76.5 | 3.7 | 1947.8 | 95.7 | 3.2 | 488.2 | 84.5 |
| 37 | 2 011 | 22.2 | 138.5 | 69.4 | 3.6 | 1655.1 | 80.3 | 3.1 | 639.2 | 67.0 |
| 38 | 3 078 | 20.6 | 159.0 | 74.8 | 3.1 | 1622.5 | 78.9 | 2.9 | 665.2 | 69.9 |
| 39 | 603 | 34.3 | 180.4 | 72.0 | 1.8 | 2525.9 | 79.6 | 1.8 | 1297.1 | 61.4 |
| 41 | 1 459 | 23.0 | 129.6 | 81.6 | 3.2 | 2054.8 | 95.0 | 2.6 | 648.0 | 84.2 |
| 42 | 947 | 31.0 | 172.0 | 97.1 | 4.9 | 3281.9 | 100.0 | 4.0 | 1151.1 | 100.0 |
| 43 | 2 427 | 25.8 | 146.9 | 69.8 | 3.5 | 2478.6 | 84.3 | 2.7 | 924.0 | 70.3 |
| 44 | 3 336 | 17.3 | 91.6 | 69.4 | 2.4 | 2187.2 | 88.0 | 1.7 | 453.4 | 75.0 |
| 45 | 1 300 | 28.2 | 129.1 | 73.0 | 2.8 | 2348.7 | 90.7 | 2.5 | 947.1 | 82.6 |
| 47 | 1 506 | 23.8 | 160.7 | 75.6 | 3.7 | 2058.7 | 91.0 | 3.5 | 800.1 | 79.3 |
| 48 | 1 546 | 24.3 | 158.5 | 71.1 | 4.5 | 2514.0 | 85.9 | 3.8 | 861.8 | 72.2 |
| 49 | 2 488 | 23.9 | 187.3 | 68.4 | 2.9 | 1862.7 | 80.4 | 2.3 | 826.6 | 61.9 |
| 50 | 1 895 | 23.6 | 143.7 | 84.8 | 1.4 | 1700.6 | 88.9 | 1.2 | 670.5 | 83.9 |
| 51 | 1 274 | 24.9 | 121.3 | 67.0 | 2.3 | 2196.5 | 80.0 | 2.1 | 727.5 | 68.2 |
| 52 | 3 038 | 22.1 | 154.6 | 64.8 | 3.0 | 1406.8 | 85.8 | 2.2 | 642.2 | 63.5 |

| | | **Modely A** | | | **Model B** | | | **Modely C** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| mluvčí | korpus | OOV | PPL | WER | OOV | PPL | WER | OOV | PPL | WER |
| [-] | [-] | [%] | [-] | [%] | [%] | [-] | [%] | [%] | [-] | [%] |
| 53 | 1 510 | 26.4 | 139.5 | 73.9 | 3.8 | 2124.1 | 90.3 | 3.4 | 746.0 | 78.4 |
| 54 | 1 125 | 25.6 | 133.9 | 72.3 | 2.0 | 3119.5 | 85.4 | 1.5 | 915.3 | 70.1 |
| 55 | 1 416 | 22.5 | 117.2 | 70.1 | 4.2 | 1829.0 | 87.6 | 3.2 | 607.6 | 75.3 |
| 56 | 1 955 | 27.3 | 203.0 | 71.1 | 3.8 | 2317.3 | 83.8 | 3.2 | 1122.4 | 73.1 |
| 57 | 1 338 | 23.3 | 128.8 | 85.0 | 2.9 | 1805.3 | 91.2 | 2.5 | 627.0 | 89.1 |
| 58 | 1 515 | 24.8 | 172.8 | 64.7 | 3.4 | 2216.8 | 75.3 | 2.8 | 868.8 | 58.7 |
| 59 | 2 497 | 21.4 | 186.7 | 75.1 | 3.3 | 1819.9 | 84.5 | 3.0 | 749.3 | 74.2 |
| 60 | 1 483 | 23.5 | 130.6 | 76.5 | 2.7 | 2260.2 | 87.0 | 2.7 | 653.1 | 78.6 |
| AVG | 2058.3 | 24.1 | 145.7 | 73.8 | 3.4 | 2178.2 | 85.4 | 2.9 | 783.1 | 74.4 |
| STD | 907.6 | 3.8 | 26.0 | 8.7 | 0.9 | 405.4 | 6.1 | 0.7 | 178.4 | 10.1 |
| MIN | 603.0 | 15.8 | 90.3 | 55.8 | 1.4 | 1406.8 | 73.0 | 1.2 | 425.0 | 48.9 |
| MAX | 4610.0 | 34.3 | 212.6 | 97.1 | 6.1 | 3281.9 | 100.0 | 4.8 | 1297.1 | 100.0 |

Tabulka 4.11: Výsledky rozpoznávání pro mluvčí NCCCz

Z důvodu množství hodnot v tabulce(4.11) byl pro snažší porovnání *WER* vytvořen graf(4.3). Každá trojice sloupců reprezentuje jednoho mluvčího. Modrá barva jsou hodnoty *WER* pro model mluvčího, zelená barva zastupuje hodnoty *WER* obecného modelu a červená barva ukazuje hodnotu *WER* modelu kombinovaného. Hodnoty *WER* jsou vynášeny od *40-100%* na osu $y$, na ose $x$ jsou vyneseni mluvčí v pořadí dle tabulky(4.11).






Obrázek 4.3: Grafické znázornění WER pro NCCCz

### 4.5.3 Modely PONDELKY

Korpus PONDELKY byl rozdělen na tři části podle témat a z každé části byl vytvořen jeden model. Jako první si opět ukážeme velikost těchto modelů.

| **model** | **unigramy** | **bigramy** | **trigramy** |
|---|---|---|---|
| PONDELKY_BIO | 5 364 | 22 013 | 29 075 |
| PONDELKY_REC | 9 269 | 43 897 | 60 062 |
| PONDELKY_SIG | 1 509 | 4 797 | 5 851 |

Tabulka 4.12: Velikost modelů korpusu PONDELKY

Jak lze odečíst z tabulky(4.12), velikost jednotlivých korpusů byla rozdílná, čemuž odpovídá i množství jednotlivých $n$-gramů. Konkrétní velikosti těchto modelů jsou uvedené v tabulce(5.1) v příloze.

| **model** | $\delta_{n=1}[-]$ | $\delta_{n=1}[\%]$ | $\delta_{n=2}[-]$ | $\delta_{n=2}[\%]$ | $\delta_{n=3}[-]$ | $\delta_{n=3}[\%]$ |
|---|---|---|---|---|---|---|
| cnk60k+PONDELKY_BIO | 1 556 | 2.53 | 8 153 | 0.03 | 23 102 | 0.13 |
| cnk120k+PONDELKY_BIO | 1 143 | 0.94 | 7 609 | 0.02 | 23 051 | 0.12 |
| cnk180k+PONDELKY_BIO | 930 | 0.51 | 7 402 | 0.02 | 23 041 | 0.11 |
| cnk240k+PONDELKY_BIO | 807 | 0.34 | 7 313 | 0.02 | 23 039 | 0.11 |
| cnk340k+PONDELKY_BIO | 650 | 0.19 | 7 233 | 0.02 | 23 035 | 0.11 |
| cnk60k+PONDELKY_REC | 3 341 | 5.27 | 18 400 | 0.07 | 48 556 | 0.27 |
| cnk120k+PONDELKY_REC | 2 484 | 2.03 | 17 236 | 0.05 | 48 420 | 0.25 |
| cnk180k+PONDELKY_REC | 2 028 | 1.11 | 16 758 | 0.04 | 48 399 | 0.24 |
| cnk240k+PONDELKY_REC | 1 760 | 0.73 | 16 554 | 0.04 | 48 387 | 0.24 |
| cnk340k+PONDELKY_REC | 1 493 | 0.44 | 16 400 | 0.04 | 48 383 | 0.24 |
| cnk60k+PONDELKY_SIG | 388 | 0.64 | 1 969 | 0.01 | 4 830 | 0.03 |
| cnk120k+PONDELKY_SIG | 302 | 0.25 | 1 872 | 0.01 | 4 823 | 0.02 |
| cnk180k+PONDELKY_SIG | 248 | 0.14 | 1 828 | 0.00 | 4 820 | 0.02 |
| cnk240k+PONDELKY_SIG | 221 | 0.09 | 1 810 | 0.00 | 4 820 | 0.02 |
| cnk340k+PONDELKY_SIG | 194 | 0.06 | 1 800 | 0.00 | 4 820 | 0.02 |

Tabulka 4.13: Změna velikosti modelů oproti CNK při kombinaci s modely korpusu PONDELKY

Pokud porovnáme množství přidaných $n$-gramů s jejich počtem v námi vytvořených modelech, můžeme si všimout, že nejvíce bylo přidáno bigramů a trigramů. Tento jev lze vysvětlit ořízntutím, které bylo aplikováno při vytváření modelů *CNK*. Bigramy a trigramy vyskytující se v našich trénovacích korpusech se s nějvětší pravděpodobností nevyskytly v Českém národním korpusu a pokud ano, tak ne s dostatečnou četností, aby prošly přes ořízntutí. Přírůstek unigramů je zhruba 1/4 až 1/3 při kombinaci s modelem *cnk60k*.

Výsledky spočtené na testovacím korpusu PONDELKY_BIO se podobají výsledkům naměřeným na testovacím korpusu *THKBK*. Nejnižší perplexity dosáhl model *PONDELKY_BIO*, ale díky své malé velikosti má vysoké *OOV*. Zajímavý je ovšem pokles *OOV* při zvětšování obecného modelu. Je vidět, že od modelu *cnk120k* již *OOV* klesá jen pomalu a téměř trojnásobně velký model *cnk340k* tak sníží počet neznámých slov pouze o dalších $1,5\%$. Z toho je možno usoudit, že jako obecný model by mohl být dostačující *cnk120k*. Další zlepšení by bylo vhodnější dosahovat rozšířením tematického korpusu, tak aby pokryl i slova mimo slovník.

Když se podíváme podrobněji na perplexity, všimneme si, že nízká váha obecného modelu v kombinaci s tematickým modelem dosahuje nejlepších výsledků. Pro další testování v případném rozpoznávači bych vybral kombinaci modelů *cnk60k+PONDELKY_BIO*, *cnk120k+*
*PONDELKY_BIO* a *cnk340k+PONDELKY_BIO*. Tak by bylo možné s jistotou určit přínos největšího modelu.

| **Model** | **Testovací korpus(počet slov)** | **Perplexita** | **OOV[%]** |
|---|---|---|---|
| PONDELKY_BIO | bio_test(14 038) | 455.08 | 14.21 |
| cnk60k | bio_test(14 038) | 1 397.91 | 11.87 |
| cnk120k | bio_test(14 038) | 1 739.35 | 8.55 |
| cnk180k | bio_test(14 038) | 2 004.28 | 6.69 |
| cnk240k | bio_test(14 038) | 2 144.21 | 5.90 |
| cnk340k | bio_test(14 038) | 2 350.29 | 4.89 |
| cnk60k+PONDELKY_BIO_25 | bio_test(14 038) | 572.71 | 6.20 |
| cnk60k+PONDELKY_BIO_50 | bio_test(14 038) | 604.08 | 6.20 |
| cnk60k+PONDELKY_BIO_75 | bio_test(14 038) | 751.06 | 6.20 |
| cnk120k+PONDELKY_BIO_25 | bio_test(14 038) | 641.74 | 4.77 |
| cnk120k+PONDELKY_BIO_50 | bio_test(14 038) | 669.31 | 4.77 |
| cnk120k+PONDELKY_BIO_75 | bio_test(14 038) | 823.50 | 4.77 |
| cnk180k+PONDELKY_BIO_25 | bio_test(14 038) | 687.18 | 4.02 |
| cnk180k+PONDELKY_BIO_50 | bio_test(14 038) | 713.03 | 4.02 |
| cnk180k+PONDELKY_BIO_75 | bio_test(14 038) | 873.27 | 4.02 |
| cnk240k+PONDELKY_BIO_25 | bio_test(14 038) | 713.76 | 3.64 |
| cnk240k+PONDELKY_BIO_50 | bio_test(14 038) | 739.09 | 3.64 |
| cnk240k+PONDELKY_BIO_75 | bio_test(14 038) | 903.61 | 3.64 |
| cnk340k+PONDELKY_BIO_25 | bio_test(14 038) | 763.95 | 3.04 |
| cnk340k+PONDELKY_BIO_50 | bio_test(14 038) | 788.77 | 3.04 |
| cnk340k+PONDELKY_BIO_75 | bio_test(14 038) | 962.15 | 3.04 |

Tabulka 4.14: Srovnání pro modely testované na testovacím korpusu PONDELKY_BIO

U tohoto korpusu byly použity modely, které jako tematický model používaly *PONDELKY_REC*. Tento model je díky největšímu množství prezentací poměrně obsáhlý, lze tak předpokládat, že jeho aplikace přinese dobré výsledky. Nyní se tedy podívejme na výsledky změřené na testovacím korpusu pro téma Zpracování řeči. Jako první se zaměříme na hodnoty *OOV*. Hned první dva řádky jsou zajímavé. Námi vytvořený tematický model má nižší hodnotu *OOV* než nejmenší obecný model a jen o $1,3\%$ více než model *cnk120k*, který má 120 000 unigramů. Při kombinaci tematického modelu s obecným modelem *cnk60k*, byla hodnota *OOV* snížena na polovinu. Další pokles je spíše pozvolný a za přínosné bych považoval použití obecného modelu do rozsahu 180 000 unigramů, poté je pokles již velmi malý. Z pohledu perplexit je opět nejvhodnější model *PONDELKY_REC*. Při pohledu na oba faktory, kterými srovnáváme jazykové modely v této práci, bych pro aplikaci bez konkrétních požadavků na velikost modelu a robustnost doporučil modely *cnk60+PONDELKY_REC* a *cnk120+PONDELKY*, které poskytují dobrý kompromis mezi velikostí a předpokládanou kvalitou.

| **Model** | **Testovací korpus(počet slov)** | **Perplexita** | **OOV[%]** |
|---|---|---|---|
| PONDELKY_REC | rec_test(33 279) | 526.998 | 10.09 |
| cnk60k | rec_test(33 279) | 1 260.78 | 11.87 |
| cnk120k | rec_test(33 279) | 1 584.01 | 8.77 |
| cnk180k | rec_test(33 279) | 1 835.39 | 6.93 |
| cnk240k | rec_test(33 279) | 1 980.67 | 6.03 |
| cnk340k | rec_test(33 279) | 2 162.58 | 5.11 |
| cnk60+PONDELKY_REC_25 | rec_test(33 279) | 567.978 | 5.74 |
| cnk60+PONDELKY_REC_50 | rec_test(33 279) | 615.354 | 5.74 |
| cnk60+PONDELKY_REC_75 | rec_test(33 279) | 775.87 | 5.74 |
| cnk120+PONDELKY_REC_25 | rec_test(33 279) | 626.856 | 4.57 |
| cnk120+PONDELKY_REC_50 | rec_test(33 279) | 674.138 | 4.57 |
| cnk120+PONDELKY_REC_75 | rec_test(33 279) | 843.947 | 4.57 |
| cnk180+PONDELKY_REC_25 | rec_test(33 279) | 673.203 | 3.83 |
| cnk180+PONDELKY_REC_50 | rec_test(33 279) | 720.646 | 3.83 |
| cnk180+PONDELKY_REC_75 | rec_test(33 279) | 898.518 | 3.83 |
| cnk240+PONDELKY_REC_25 | rec_test(33 279) | 708.142 | 3.34 |
| cnk240+PONDELKY_REC_50 | rec_test(33 279) | 755.75 | 3.34 |
| cnk240+PONDELKY_REC_75 | rec_test(33 279) | 939.889 | 3.34 |
| cnk340+PONDELKY_REC_25 | rec_test(33 279) | 741.48 | 2.91 |
| cnk340+PONDELKY_REC_50 | rec_test(33 279) | 789.401 | 2.91 |
| cnk340+PONDELKY_REC_75 | rec_test(33 279) | 979.91 | 2.91 |

Tabulka 4.15: Srovnání pro modely testované na testovacím korpusu PONDELKY_REC

Poslední srovnání modelů tohoto korpusu i celé práce je pro model *PONDELKY_SIG* a jeho kombinace s modely CNK. Výsledky naměřené na tomto testovacím korpusu jsou zkreslené jeho velikostí. Jak vidíme, tak *OOV* pro model *PONDELKY_SIG* vyšlo 22%, to je velmi vysoká hodnota. Pokud by byl model s takto vysokou hodnotou *OOV* použit v systému rozpoznávání řeči, celé rozpoznávání by pravděpodobně dosáhlo velmi nízké úspěšnosti. Pokud se podíváme na pokles *OOV* při zvětšování obecného modelu, lze za poslední model s relativně velkým přínosem označit model *cnk240k*. U minulých dat jsme došli k závěru, že velký přinos má model *cnk180k* nebo menší. Tato změna oproti minulým modelům je způsobena malým trénovacím korpusem, který nedostečně zachycuje toto téma. Perplexity pro kombinované modely vyšly přibližně v rozmezí $500 - 1000$, tyto hodnoty tudíž korespondují s minulými měřeními. Jako optimální modely, při uvážení obou veličin, bych zvolil modely skupiny *cnk240k+PONDELKY_SIG*, a to hlavně kvůli nízkému *OOV*.

| **Model** | **Testovací korpus(počet slov)** | **Perplexita** | **OOV[%]** |
|---|---|---|---|
| PONDELKY_SIG | sig_test(3 292) | 241.20 | 22.11 |
| cnk60k | sig_test(3 292) | 1 681.41 | 13.94 |
| cnk120k | sig_test(3 292) | 2 319.77 | 9.81 |
| cnk180k | sig_test(3 292) | 2 918.91 | 6.90 |
| cnk240k | sig_test(3 292) | 3 321.77 | 5.38 |
| cnk340k | sig_test(3 292) | 3 564.09 | 4.65 |
| cnk60k+PONDELKY_SIG_25 | sig_test(3 292) | 516.07 | 7.53 |
| cnk60k+PONDELKY_SIG_50 | sig_test(3 292) | 547.62 | 7.53 |
| cnk60k+PONDELKY_SIG_75 | sig_test(3 292) | 700.45 | 7.53 |
| cnk120k+PONDELKY_SIG_25 | sig_test(3 292) | 617.84 | 5.56 |
| cnk120k+PONDELKY_SIG_50 | sig_test(3 292) | 648.35 | 5.56 |
| cnk120k+PONDELKY_SIG_75 | sig_test(3 292) | 821.69 | 5.56 |
| cnk180k+PONDELKY_SIG_25 | sig_test(3 292) | 705.96 | 4.22 |
| cnk180k+PONDELKY_SIG_50 | sig_test(3 292) | 735.45 | 4.22 |
| cnk180k+PONDELKY_SIG_75 | sig_test(3 292) | 925.86 | 4.22 |
| cnk240k+PONDELKY_SIG_25 | sig_test(3 292) | 754.69 | 3.55 |
| cnk240k+PONDELKY_SIG_50 | sig_test(3 292) | 784.63 | 3.55 |
| cnk240k+PONDELKY_SIG_75 | sig_test(3 292) | 986.10 | 3.55 |
| cnk340k+PONDELKY_SIG_25 | sig_test(3 292) | 807.93 | 2.95 |
| cnk340k+PONDELKY_SIG_50 | sig_test(3 292) | 837.61 | 2.95 |
| cnk340k+PONDELKY_SIG_75 | sig_test(3 292) | 1 050.52 | 2.95 |

Tabulka 4.16: Srovnání pro modely testované na testovacím korpusu PONDELKY_SIG

# Kapitola 5

# Závěr

Zadáním této práce bylo vytvořit jazykové modely pro tematické oblasti dostupných korpusů a příprava nástrojů pro vytvoření jazykových modelů při získání korpusů pro novou tematickou oblast.

Pro splnění této části byl vytvořen soubor skriptů, které pracují s balíkem nástrojů SRILM, programovacím jazykem Perl a dalšími nástroji dostupnými v Linuxové distribuci Ubuntu. Při použití této práce jako návodu, je možné pro jakýkoliv korpus textů vytvořit jazykové modely a ty poté otestovat.

Součástí této práce byla i příprava jazykového modelu pro použití v rozpoznávači řeči. Pro splnění tohoto úkolu byl vytvořen skript, který vytváří z jazykového modelu výslovnostní slovník.

Vytvořené jazykové modely jsme testovali a porovnávali použitím OOV a perplexity. Pro modely z NCCCz (The Nijmegen Corpus of Casual Czech) byly kromě OOV a perplexit naměřeny i hodnoty WER v rozpoznávači řeči. Závěry s použitím WER se shodují se závěry učiněnými na základě OOV a perplexit. Díky tomu víme, že výsledky naměřené na ostatních korpusech a závěry z nich utvořené jsou správné.

V budoucí práci by bylo dobré se zaměřit na zmenšení modelů, konkrétně porovnat modely s různým vyhlazováním a oříznutím a vytvořit tak maximálně efektivní model z pohledu jeho velikosti. Další oblast, která by mohla zlepšit výsledky jazykových modelů a nebyla v této práci příliš prozkoumána, je kombinování modelů. Kromě lineární kombinace, použité v této práci, existují i jiné metody, které by mohly také přispět ke zlepšení.

Při zpracování této práce jsem si uvědomil některé skutečnosti z tématu zpracování řeči, které mi dříve unikaly nebo nebyly úplně jasné. Dále jsem také získal nové schopnosti související s prací s textem v systémech na bázi Linux. Práce pro mě byla velmi přínosná a doufám, že i modely a postup, který jsem navrhl bude dále využíván.

# Literatura


[1] Steve Renals a Hiroshi Shimodaira. Automatic speech recognition. `http://www.inf.ed.ac.uk/teaching/courses/asr/`.

[2] Josef Psutka a Luděk Müller a Jindřich Matoušek a Vlasta Radová. *Mluvíme s počítačem česky*. Academia, 2006. 80-200-1309-1.

[3] Marcello Federico a Nicola Bertoldi. Irst language modeling toolkit. `http://sourceforge.net/projects/irstlm/`.

[4] Václav Procházka a Petr Pollák a Jindřich Žďánský a Jan Nouza. Performance of czech speech recognition with language models created from public resources.

[5] Kristie Seymore a Ronald Rosenfeld. Scalable backoff language models. `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=607084`.

[6] Filozofická fakulta Univerzity Karlovy v Praze. Český národní korpus. `http://korpus.cz/`.

[7] SRI International R&D for Government and Business. The sri language modeling toolkit. `http://www.speech.sri.com/projects/srilm/`.

[8] Tomáš Mikolov. Modelování jazyka v rozpoznávání češtiny. Master's thesis, Vysoké učení technické v Brně, 2007.

[9] Dana Nejedlová. *Creation of Lexicons and Language Models for Automatic Broadcast News Transcription*. PhD thesis, Technická univerzita v Liberci, 2006.

[10] University of Cambridge. Carnegie mellon university statistical language modeling toolkit. `http://svr-www.eng.cam.ac.uk/~prc14/toolkit_documentation.html`.

[11] Massachusetts Institute of Technology. Mit language modeling toolkit. `https://code.google.com/p/mitlm/`.

[12] VMware(NYSE:VMW). Vmware player. `http://www.vmware.com/cz/products/player`.

# Seznam tabulek

# Seznam obrázků

# Seznam zkratek

CMU SLM - Carnegie Mellon Statistical Language Modeling
MITLM - Massachusetts Institute of Technology Language Modeling
SRILM - Stanford Research Institute Language Modeling
TCL - Tool Command Language
SOS - Start Of Sentence
EOS - End Of Sentence
OOV - Out Of Vocabulary
PP/PPL - Perplexita
SED - Strem Editor
CNK - Český Národní Korpus
IPA - International Phonetic Alphabet
CTU - Czech Technical University
WER - Word Error Rate
ACC - Accuracy
RTF - Real Time Factor

# Přílohy

| model | unigramy | bigramy | trigramy |
|---|---|---|---|
| cnk60k+PONDELKY_REC | 63 343 | 28 292 382 | 18 024 900 |
| cnk120k+PONDELKY_REC | 122 486 | 36 420 988 | 19 549 853 |
| cnk180k+PONDELKY_REC | 182 030 | 40 336 730 | 20 077 147 |
| cnk240k+PONDELKY_REC | 241 762 | 42 574 355 | 20 316 481 |
| cnk340k+PONDELKY_REC | 341 495 | 44 619 219 | 20 491 597 |
| cnk60k+PONDELKY_BIO | 61 558 | 28 282 135 | 17 999 446 |
| cnk120k+PONDELKY_BIO | 121 145 | 36 411 361 | 19 524 484 |
| cnk180k+PONDELKY_BIO | 180 932 | 40 327 374 | 20 051 789 |
| cnk240k+PONDELKY_BIO | 240 809 | 42 565 114 | 20 291 133 |
| cnk340k+PONDELKY_BIO | 340 652 | 44 610 052 | 20 466 249 |
| cnk60k+PONDELKY_SIG | 60 390 | 28 275 951 | 17 981 174 |
| cnk120k+PONDELKY_SIG | 120 304 | 36 405 624 | 19 506 256 |
| cnk180k+PONDELKY_SIG | 180 250 | 40 321 800 | 20 033 568 |
| cnk240k+PONDELKY_SIG | 240 223 | 42 559 611 | 20 272 914 |
| cnk340k+PONDELKY_SIG | 340 196 | 44 604 619 | 20 448 034 |

Tabulka 5.1: Velikost modelů kombinace CNK a PONDELKY