VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

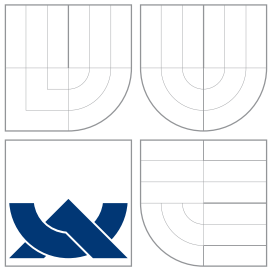
FILTROVÁNÍ TEXTŮ EXTRAHOVANÝCH Z PDF, OCR NEBO WEBU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

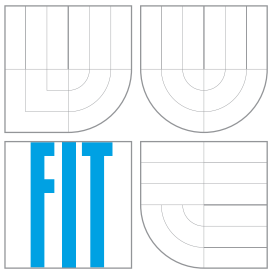
AUTOR PRÁCE
AUTHOR

TOMÁŠ ŽIGÁRDI

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

FILTROVÁNÍ TEXTŮ EXTRAHOVANÝCH Z PDF, OCR NEBO WEBU

FILTERING OF TEXTS EXTRACTED FROM PDF, OCR OR WEB

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — TOMÁŠ ŽIGÁRDI
AUTHOR

VEDOUCÍ PRÁCE — Ing. IGOR SZŐKE, Ph.D.
SUPERVISOR

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá normalizací textů vzniklých převedením z různých formátů a vytvořením výslovnostních slovníků. Jednou z jejich možností využití je například při strojovém zpracování řeči. Analyzovány jsou chyby, které vznikají při převodu a původní řešení tohoto problému. Dále je uveden návrh a implementace normalizačních kroků a výslovnostních slovníků. Výsledky implementovaného řešení jsou vyhodnoceny a porovnány s existujícím řešením.

Abstract

This bachelor thesis describes normalization of texts created by conversion of other formats and creation of pronunciation dictionaries. They are important in speech processing process. Mistakes caused by conversion and original solution of this problem are analyzed. Design and implementation of normalization steps and pronunciation dictionaries is shown. Results are compared with results of original solution of this problem.

Klíčová slova

Normalizace textu, OCR, PDF, výslovnostní slovníky, regulární výraz, Bash, Perl, Awk.

Keywords

Text normalization, OCR, PDF, pronunciation dictionaries, regular expression, Bash, Perl, Awk.

Citace

Tomáš Žigárdi: Filtrování textů extrahovaných z PDF, OCR nebo webu, bakalářská práce, Brno, FIT VUT v Brně, 2013

Filtrování textů extrahovaných z PDF, OCR nebo webu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Igora Szőkeho. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Žigárdi
15. května 2013

Poděkování

Chcel by som poďakovať vedúcemu práce, Ing. Igorovi Szőkemu, Ph.D, za jeho odbornú pomoc, poskytnuté rady, konzultácie, ochotu a čas, ktorý mi pri tvorbe práce venoval.

Obsah

Kapitola 1

Úvod

Táto práca sa zaoberá automatizovanou úpravou jednoduchého anglického textu a vytvorením výslovnostných slovníkov pre výsledný text. Upravovaný text vznikol prevedením z rôznych formátov.

S textom uloženým v niektorých formátoch často nie je jednoduché pracovať (časť z neho vymazať, prepísať a podobne). Preto je vhodné texty z vybraných formátov extrahovať a upraviť v editore. Získaný text obsahuje mnoho nedokonalostí. Tie je potrebné vhodne opraviť. Pre výsledný text sú vytvorené výslovnostné slovníky, ktoré sú dôležité napríklad pri strojovom spracovaní reči. Text upravený do výslednej podoby, môže byť použitím výslovnostných slovníkov a vhodnej aplikácie automaticky prečítaný počítačom, alebo iným zariadením.

Pôvodné formáty a metódy, ktorými sú texty získané, sú opísané v úvodnej časti práce. Nasleduje popis chýb, ktoré vznikli prevodom (je nutné ich opraviť) a ďalších úprav textu do výslednej podoby. V ďalšej časti je opísané pôvodné riešenie tohoto problému a jeho kladné a záporné vlastnosti. Nasleduje popis jednotlivých krokov, ktorými boli texty upravené a popis pravidiel na úpravu, ktoré boli vytvorené a aplikované. Vysvetlený je aj postup pri vytváraní a úprave výslovnostných slovníkov. Ďalej sú porovnané výsledky pôvodného riešenia problému s navrhnutým a implementovaným riešením. V záverečnej časti sú navrhnuté smery ďalšieho vývoja vytvorenej práce.

K práci bol vytvorený plagát popisujúci postup pri úprave textu a vytváraní slovníkov, doplnený o praktické ukážky.

Kapitola 2

Popis technológií a metód použitých v práci

Texty určené na spracovanie majú tri hlavné zdroje. Prvým z nich sú textové súbory získané ľubovolným spôsobom. Napríklad z webových stránok. V každom z najpoužívanejších prehliadačov je možnosť uložiť stránku ako jednoduchý text. Takto získané texty sa svojou kvalitou najviac približujú originálnemu formátu.

Ďalším zdrojom sú PDF (Adobe Portable Document Format) dokumenty. Na prevod dokumentov tohoto formátu slúži mnoho nástrojov. V práci bol použitý linuxový nástroj pdftotext a metóda OCR (optical character recognition). Výber použitej technológie závisí od rozhodnutia používateľa.

Posledným zdrojom textov sú obrázky rôznych formátov, ktoré obsahujú textové informácie, napríklad oscanovaný text. V tomto prípade bola tiež použitá technológia OCR. Výber spracovaných obrázkových formátov taktiež ovplivňuje používateľ.

2.1 Popis formátu PDF

PDF je multiplatformový súborový formát, ktorý sa používa na reprezentovanie dokumentov. Je nezávislý od aplikácie, hardwaru, alebo operačného systému na ktorom bol vytvorený, rovnako ako na výstupných zariadeniach, na ktorých sa dokument používa.

PDF dokument je zložený z objektov, ktoré charakterizujú jeho štruktúru a vzhľad. V súbore sú všetky objekty a štrukturované informácie reprezentované ako jedna postupnosť bytov. Dokument môže okrem textu obsahovať grafy, tabuľky, obrázky, informácie vyššej úrovne a interaktívne prvky (použiteľné iba v elektronickej verzii dokumentu). Patria sem napríklad textové poznámky, hypertextové odkazy, prílohy, zvuky a videá. Môžu byť definované aj akcie, ktoré sa majú vykonať pri vstupe z klávesnice, alebo myši. Informácie vyššej úrovne umožujú zmenu obsahu a vzhľadu dokumentu medzi aplikáciami. Dokument môže obsahovať identifikačné a logické štruktúry, ktoré umožujú vyhľadávanie v dokumente, upravovanie a následné použitie v inej aplikácii. Spomenuté informácie sú čerpané z oficiálnej dokumentácie k formátu [1].

Na prevod formátu PDF do podoby jednoduchého textu je použitý nástroj pdftotex. Je to open-source program príkazového riadku Linuxu. V mnohých distribúciách Linuxu je tento program nainštalovaný priamo s operačným systémom. Je použiteľný aj v operačnom systéme Windows a to ako súčasť programu Xpdf.

2.2 Popis metódy OCR

OCR je metóda slúžiaca na automatické rozpoznávanie opticky spracovaných znakov. Rozpoznané dokážu byť texty písané strojovým, ale aj ručným písmom. Kvalita výsledku závisí od kvality vstupného dokumentu. Rozpoznávanie znakov môže prebiehať off-line, alebo on-line. Off-line rozpoznávanie sa vykonáva z finálnej podoby textu, pričom on-line rozpoznávanie prebieha už počas písania.

Postup pri rozpoznávaní znakov pozostáva väčšinou z týchto krokov:

- Optické scanovanie: digitalizovanie analógového dokumentu.
- Segmentácia polohy: izolovanie znakov alebo slov.
- Predspracovanie: normalizácia, teda získanie znakov jednotného tvaru, veľkosti a rotácie.
- Výber charakteristických znakov: určenie charakteristických znakov jednotlivých symbolov, ako napríklad ich tvar.
- Klasifikácia: identifikácia jednotlivých znakov a ich priradenie do danej skupiny znakov.
- Konečné spracovanie: spojenie znakov do slov a viet, vyhľadanie chýb a ich oprava.

Informácie sú čerpané z knihy od Eikvila [4].

V práci prebieha rozpoznávanie znakov off-line, teda z finálnej podoby textu v PDF súbore, alebo nascanovaného textu v podobe obrázku. Na rozpoznanie je použitá linuxová verzia nástroja tesseract-ocr, často dostupná priamo v distribúcii. Podľa Smitha [8] zaostáva tesseract-ocr v presnosti rozpoznávania znakov v porovnaní s komerčnými OCR nástrojmi. Je to však open-source nástroj, ktorého kvalita je na účel práce dostatočná. Pre nástroj existuje mnoho rozšírení, vďaka dokáže rozpoznať text z veľkého počtu formátov vo vyše 60 jazykoch, čo je dobrý základ na prípadné rozšírenie tejto práce v budúcnosti.

2.3 Výslovnostné slovníky

Výslovnostné slovníky sú jednou z najdôležitejších častí systémov, ktoré sa zaoberajú spracovaním reči. Podľa práce zaoberajúcej sa touto tematikou [9], je úlohou výslovnostných slovníkov mapovať ortografickú reprezentáciu slova a jeho výslovnosť. Pre každý jazyk existujú špecifické pravidlá pre výslovnosť znakov. Slovníky obsahujú často tisíce záznamov, preto je nutné ich tvorbu z čo najväčšej časti automatizovať. Výslovnostné slovníky majú využitie hlavne pri automatizovanom prevode textu do reči a naopak. Napríklad v aplikáciách slúžiacich pre ľudí so zrakovím, alebo rečovím postihnutím.

2.4 Definícia gramatiky a regulárnych výrazov

V práci sú pri priraďovaní slov do skupín, alebo pri ich nahradzovaní použité regulárne výrazy. Nasleduje ich formálna definícia (prevzatá z [3]).

Definícia 2.4.1. Nech Σ je *abeceda*. *Regulárne výrazy*(RV) nad abecedou Σ a *jazyky*, ktoré označujú sú definované nasledovne:

- $\varnothing$ je RV označujúci prázdnu *množinu* (prázdny jazyk).
- ε je RV označujúci jazyk $\{\varepsilon\}$.
- a, kde $a \in \Sigma$, je RV označujúci jazyk $\{a\}$.
- Nech r a s sú RV označujúce postupne jazyky L_r a L_s, potom:
  - $(r.s)$ je RV označujúci jazyk $L = L_r L_s$.
  - $(r + s)$ je RV označujúci jazyk $L = L_r \cup L_s$.
  - (r^*) je RV označujúci jazyk $L = L_r^*$.

Jednotlivé vytvorené pravidlá, použité pri úprave slov na iné, sa dajú označiť za bezkontextové gramatiky. Ich definícia (prevzatá z [6]) vyzerá nasledovne:

Definícia 2.4.2. *Bezkontextová gramatika* (BKG) je štvorica $G = (N, T, P, S)$, kde

- N je abeceda *neterminálov*.
- T je abeceda *terminálov*, pričom $N \cap T = \varnothing$.
- P je konečná množina *pravidiel* tvaru $A \to x$,kde $A \in N, x \in (N \cup T)^*$.
- $S \in N$ je počiatočný neterminál.

Využitie gramatík a regulárnych výrazov v implementácii je popísané v ďalších častiach práce.

Kapitola 3

Analýza chýb vstupného textu a predchádzajúceho riešenia problému

Pri spracovaní môžu vzniknúť chyby spôsobené deformáciou jednotlivých znakov, ich rôznym tvarom (rôzne fonty, veľkosti, štýly), rôznou veľkosťou medzier medzi znakmi a slovami, alebo zmiešaním textu a grafiky. Preto sa môže stať, že je znak priradený do iného slova, alebo je nesprávne identifikovaný.

3.1 Chyby pri prevode formátov do podoby jednoduchého textu

Patria sem nesprávne zobrazené obrázky a diagramy. Často obsahujú znaky, ktoré sa nesprávne prevedú do podoby jednoduchého textu. Jednotlivé slová, alebo spojenia diagramu sú rozdelené na samostatný riadok, za ktorým nasleduje prázdny riadok.

Podobný problém spôsobujú aj rovnice. Množstvo prevedených rovníc je rozdelených na jednotlivé znaky a každý znak je zobrazený na samostatnom riadku. Ďalším nedostakom je zobrazovanie indexov znakov. Keďže nie je možné v jednoduchom texte zobrazovať horný a dolný index znakov, tieto indexy sú spojené s daným znakom na jednej úrovni. Rovnice sú preto nepoužiteľné a je potrebné zbaviť sa ich. Príklad:

$$\frac{\partial \mathcal{F}}{\partial \lambda_{s,n}}(\Lambda) = \frac{1}{T}\sum_{t=1}^{T}\left(p_\Lambda(s|x_t) - \delta(s,s_n)\right) f_n(x_t) + \lambda_{s,n}, \quad (4)$$

and

$$\frac{\partial^2 \mathcal{F}}{\partial \lambda_{s,n}\partial \lambda_{\bar{s},\bar{n}}}(\Lambda) = \frac{1}{T}\sum_{t=1}^{T} p_\Lambda(s|x_t)(\delta(s,\bar{s}) - p_\Lambda(\bar{s}|x_t)) \cdot f_n(x_t) f_{\bar{n}}(x_t) + \alpha\,\delta(s,\bar{s})\delta(n,\bar{n})\,, \quad (5)$$

```
(p (s|xt ) - (s, sn )) fn (xt ) + s,n , (4)
t=1

p (s|xt )((s, s) - p (¯|xt )) ¯ s
t=1

(5)

· fn (xt )fn (xt ) +  (s, s)(n, n) , ¯ ¯ ¯
```

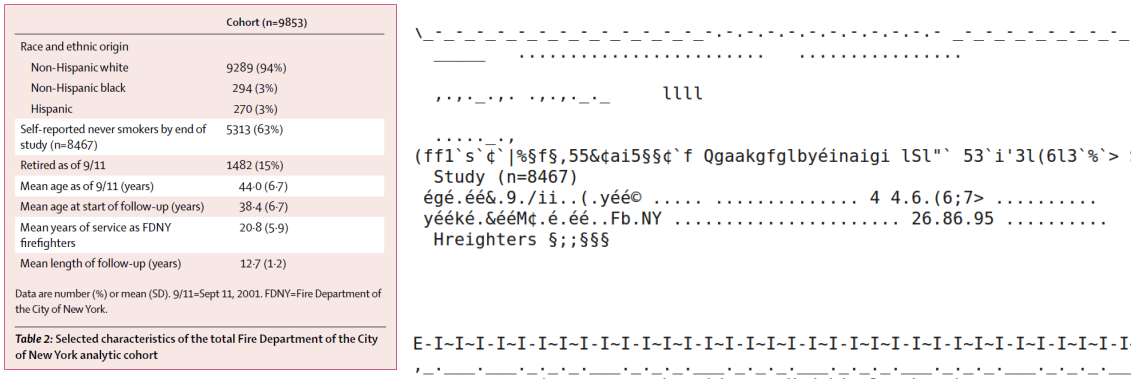
Obrázok 3.1: Extrahovanie rovníc. Vľavo pôvodné PDF, vpravo text

Jednotlivé odseky, alebo bloky textu môžu byť vo výstupe zobrazené na jednom riadku aj napriek tomu, že v zdrojovom súbore sú rozdelené na väčší počet riadkov. V prípade, že je pôvodný text rozdelený do viacerých stĺpcov, môže byť výstupný text nesprávne zoradený.

Napríklad, ak je strana rozdelená na dva stĺpce a každý z nich obsahuje blok textu, objekt (napríklad tabuľku, alebo obrázok) a ďalší blok textu. Na strane sa teda nachádzajú štyry bloky textu. V ľavom stĺpci bloky A a B, v pravom stĺpci bloky C a D. Zoradené sú v poradí A–B–C–D. Vo výslednom textovom súbore môžu byť bloky zobrazené v poradí A–C–B–D. Je to spôsobené tým, že blok C začína na strane v pravom stĺpci vyššie ako blok B v ľavom a preto je chybne spracovaný skôr.

Pri zdrojovej strane s viac stĺpcami môže nastať taktiež chybné spracovanie zápätia strany. Po spracovaní prvého stĺpca na strane je namiesto nasledujúceho stĺpca spracované najprv zápätie. Táto nedokonalosť môže skomplikovať vymazávanie nepotrebných reťazcov vo výstupných textových súboroch. Pretože zápätie nie je vo výstupnom texte umiesnené na záver, ale napríklad v strede. Až potom nasleduje spracovaný druhý stĺpec.

Nesprávne spracované sú aj tabuľky. Ich hodnoty sú vo výslednom texte nevyužiteľné. Často sú rozdelené na viac riadkov ako v danej tabuľke. Technológia OCR môže nesprávne rozpoznať aj orámovanie tabuľky ako znak „/“. Alebo postupnosť rôznych znakov. Napríklad „i2i2i2i2i2i2“, alebo „_____“ čo zvyšuje neprehľadnosť a nepoužiteľnosť výstupu. Príklad:



	Cohort (n=9853)
Race and ethnic origin	
Non-Hispanic white	9289 (94%)
Non-Hispanic black	294 (3%)
Hispanic	270 (3%)
Self-reported never smokers by end of study (n=8467)	5313 (63%)
Retired as of 9/11	1482 (15%)
Mean age as of 9/11 (years)	44·0 (6·7)
Mean age at start of follow-up (years)	38·4 (6·7)
Mean years of service as FDNY firefighters	20·8 (5·9)
Mean length of follow-up (years)	12·7 (1·2)

Data are number (%) or mean (SD). 9/11=Sept 11, 2001. FDNY=Fire Department of the City of New York.

Table 2: Selected characteristics of the total Fire Department of the City of New York analytic cohort

Obrázok 3.2: Extrahovanie tabuliek pomocou OCR. Vľavo pôvodné PDF, vpravo text

Nepresnosti vznikajú aj pri rozpoznávaní jednotlivých znakov. Niektoré špeciálne znaky, ktoré nepatria medzi základné, napríklad znaky s diakritikou, môžu byť vynechané alebo nahradené nesprávnym znakom.

Aj nízka kvalita zdrojových súborov spôsobuje pri technológii OCR nepresnosti pri rozpoznávaní jednotlivých znakov. Napríklad znak „w“ je rozpoznaný ako „vv“. Alebo znak „f“ ako „/“, „l“ ako „I“, „f“ ako „h“ a podobne. V niektorých prípadoch sú slová nesprávne rozdelené jednou, alebo väčším počtom medzier. Napríklad slovo „introduction“ je spracované ako „int r o d u c tio n“. Tento nedostatok sa prejavuje hlavne pri spracovaní textu metódou OCR.

Popis k jednotlivým obrázkom, diagramom, alebo rovniciam, ktorý je v originálnom dokumente na jednom riadku, je vo výslednom texte často rozdelený na viac riadkov. Podobne odkaz na tieto objekty môže byť nesprávne umiestnený na samostatnom riadku ohraničenom prázdnymi riadkami.

Hlavne kôli menej presnému rozpoznávaniu znakov a tabuliek sú texty získané touto metódou menej kvalitné ako pri použití nástroja pdftotext. Tieto texty si preto vyžadujú väčšie úpravy. Ukážka pôvodného pdf súboru a extrahovaného textu touto metódou je zobrazená na obrázku B.1 a B.2 v dodatku B.

3.2 Analýza dodaných skriptov na čistene textu a tvorbu výslovnostných slovníkov

V pôvodnom riešení je text normalizovaný sadou skriptov a následne sú vytvorené pravidlá, podľa ktorých sa jednotlivé slová z textu upravia, alebo vymažú. Takto získaný výsledný text však obsahuje niekoľko nedokonalostí, ktoré sú v novom riešení opravené. Napríklad je to nadmerný počet medzier medzi slovami, prázdne riadky, alebo dlhé postupnosti čísel, ktoré sú zbytočné. Jedným zo skriptov sú aj tieto nadbytočné postupnosti čísel zmenené na ich slovné vyjadrenie a sú ďalej zbytočne spracované, čo môže spomaliť a znekvalitniť riešenie.

Jednotlivé reťazce v texte sú porovnávané so zoznamom dobrých a zlých slov a na základe výsledku sú vytvorené pravidlá. Pri každom spustení skriptov sú tak vytvárané pravidlá znova. V implementovanej práci je tento krok vylepšený. Vytvorené pravidlá sa v ňom používajú aj pri ďalších spusteniach skriptov a sú postupne rozširované. Vyhľadávanie preto nastáva nielen so zoznamom slov, ale aj s doteraz vytvorenými pravidlami. Konkrétne je tento postup opísaný v sekcii 5.3.

Spracované slová sú v dodanom riešení rozdelené do skupín podľa ich vlastností. Toto rozdelenie bolo použité aj v navrhnutom riešení a je bližšie popísané v sekcii 5.1. Skupiny slov sú použité pri úprave textu aj pri tvorbe výslovnostných slovníkov. Ich tvorba prebieha v pôvodnom riešení v niekoľkých iteráciách, v ktorých sú slová vyhľadávané v existujúcich slovníkoch a v prípade neúspechu sú vygenerované nové výslovnosti.

3.3 Potrebné kroky normalizácie textu

Podľa Krzysztofa a jeho kolegov [5], je normalizácia často prvou fázou spracovania textu v systémoch prevádzajúch text na reč. To platí aj v tomto prípade. Na základe vyššie spomenutých chýb, ktoré vznikajú v texte a analýzy predchádzajúceho riešenia boli navrhnuté nasledujúce potrebné kroky. Proces normalizácie je podľa práce zaoberajúcej sa touto témou [7], vhodné rozdeliť na 2 časti. Jazykovo nezávislé normalizačné pravidlá a normalizačné pravidlá špecifické pre jazyk.

Okrem nedostatkov vzniknutých prevedením dokumentov do textovej podoby, ktoré boli spomenuté v predchádzajúcej kapitole, je potrebné vymazať alebo upraviť nasledujúce prvky výstupného textu.

Jazykovo nezávislé normalizačné pravidlá

Vo výstupe je potrebné okrem spomenutých nedostatkov vymazať aj zdroje, väčšinou umiestnené v závere dokumentu, pretože obsahujú množstvo mien, skratiek a ďalších údajov, ktoré sú nepotrebné pri vytváraní štatistík slov a výslovnostných slovníkov.

Z rovnakého dôvodu sú z textu vymazané aj odkazy na internetové stránky a e-mailové adresy. Slová v týchto reťazcoch nie sú oddelené medzerou a nemusia mať žiadny význam v danom jazyku. Pretože e-mailovú adresu, alebo odkaz na internetovú stránku si ľudia vytvárajú sami, nemusia vyberať existujúce slová. Rovnako je dôležité vymazať aj zoznam autorov, poprípade abstrakt dokumentu, ak sa v ňom nachádza.

V dokumentoch sa často objavujú záhlavia, zápätia, čísla stránok a iné pravidelne sa opakujúce reťazce. Z výsledného textového súboru je potrebné ich vymazať. Rovnako sú vymazané aj vety, v ktorých je výrazne viac čísel a rôznych symbolov, ako korektných slov.

Normalizačné pravidlá špecifické pre jazyk

Patrí sem napríklad nahradzovanie znakov alebo slov vhodnejšou formou daného slova. Ďalej nahradenie symbolov, ako napríklad „%“ na „percents“, alebo „$“ na „dollars“ a podobne. Taktiež čísla sú nahradené ich slovnou formou. Týka sa to napríklad dátumov, času, alebo jednoduchého číselného údaju.

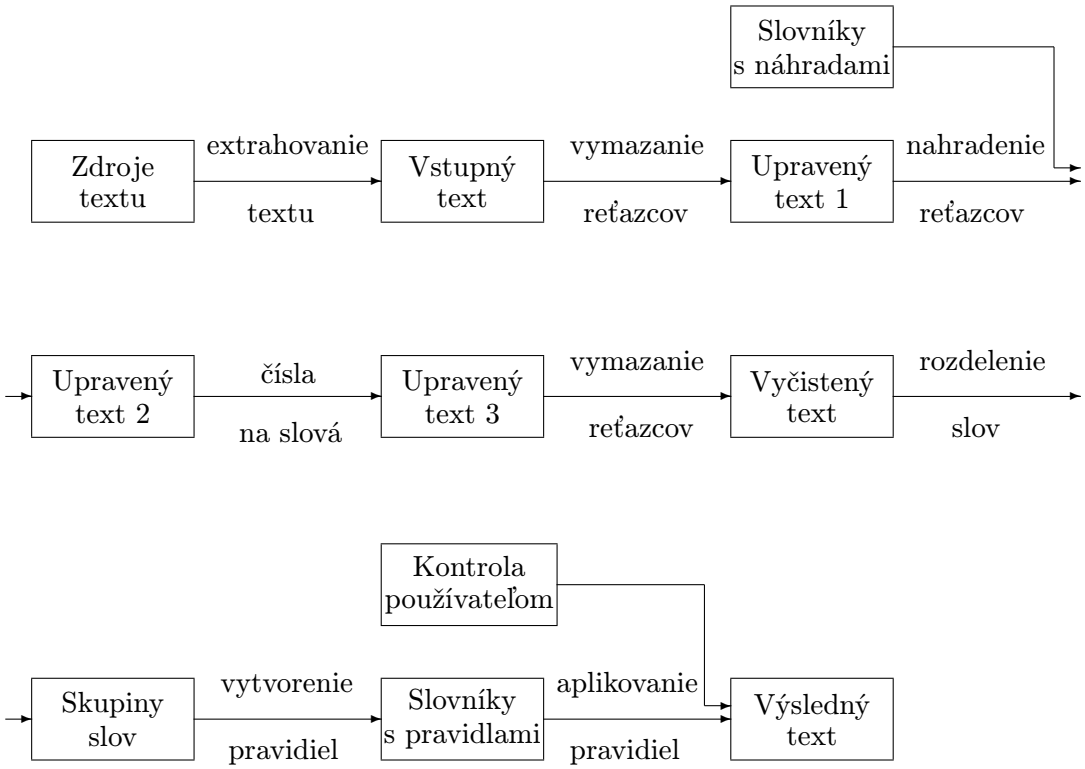
Skratky sú nahradené celými slovami a chybné slová správnou variantou slova. Vymazané sú aj nevhodné znaky, ako napríklad interpunkčné znamienka alebo neznáme symboly.

Väčšina písmen je zmenená na malé. Týka sa to hlavne slov, ktoré začínali vetu alebo tvorili nadpis. V prípade mien, napríklad „New York“, alebo skratiek ako „WTC“ (World Trade Center) sú veľké písmená ponechané.

Kapitola 4

Implementovanie jednotlivých krokov normalizácie

Na základe poznatkov získaných z jednotlivých textových výstupov a predchádzajúceho riešenia daného problému bolo naimplementované nové riešenie. Použité nástroje, v ktorých bolo riešenie vytvorené sú Bash, Awk, Sed a Perl. Boli uprednostnené kôli ich veľmi kvalitnej práci so súbormi a textom. Postup, ktorý bol použitý pri normalizácii vyjadruje nasledujúci obrázok:



Obrázok 4.1: Postup pri úprave textu

Vo väčšine prípadov bol každý krok implementovaný ako samostatný skript. Výstup je teda získaný postupným aplikovaním jednotlivých skriptov na vstupné súbory. Pred spustením skriptov s normalizačnými pravidlami používateľ definuje svoje požiadavky v súbore *config.sh*. Medzi povinné patrí zoznam adresárov, v ktorých sa nachádzaju vstupné súbory.

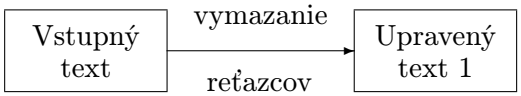
V týchto adresároch a ich podadresároch sú vyhľadané jednotlivé PDF dokumenty, textové súbory (získané uložením webovej stránky ako textu alebo iným spôsobom) a rôzne formáty obrázkov s textom. Tie taktiež definuje používateľ. Ak neurčí žiadne, obrázky spracované nie sú. Používateľ určí aj spôsob konverzie formátu PDF do podoby textu (nástroj pdftotext alebo metóda OCR).

Textové súbory na ktoré budú aplikované pravidlá sú rozdelené do výstupných adresárov podľa príslušnosti k vstupným adresárom, ktoré určil používateľ.

V ďalších častiach práce budú jednotlivé kroky a pravidlá popísané na textových súboroch získaných z formátu PDF. Sú však použiteľné aj na ostatné zdrojové formáty (text získaný z webu alebo z obrázku). Texty z jednotlivých PDF dokumentov sú rozdelené do samostatných adresárov. Tie obsahujú súbory, pričom každý z nich predstavuje textovú podobu jednej strany zdrojového PDF súboru. V prípade, že sú na spracovanie určené štyri dokumenty a každý má osem strán, vo výstupnom adresáry sú vytvorené štyri adresáre a v každom z nich osem súborov.

4.1 Vymazanie používateľom určených častí textu a nevhodných reťazcov

Prvá úprava jednoduchého textu je vymazanie nepotrebných častí textu. Z celkového postupu je to fáza:



Obrázok 4.2: Určenie fázy normalizácie

Používateľ si v súbore *config.sh* môže určiť reťazce, podľa ktorých sa vymazávanie bude riadiť. Na výber sú nasledujúce možnosti:

Vymazanie úvodu celého dokumentu

Táto možnosť má využitie, ak je potrebné vymazať zoznam autorov a detailov o nich, abstrakt, poprípade niektoré z prvych kapitol dokumentu. V tomto prípade sa prechádza každý textový súbor (každá strana), ktorý patrí k danému PDF dokumentu a hľadá sa používateľom zadaný regulárny výraz. Ak bol nájdený napríklad v treťom súbore, znamená to, že prvé dva súbory sú odstránené. Úvodná časť tretieho súboru, vrátane zadaného reťazca je vymazaná. Do ďalšieho spracovania teda začiatočná časť dokumentu už nebude zahrnutá.

Vymazanie záveru celého dokumentu

V závere dokumentu je potrebné odstrániť odkazy, zoznam diel, z ktorých práca čerpal a ďalšie nepotrebné reťazce textu, ktoré by neboli použiteľné pri ďalšej práci s textom. Rov-

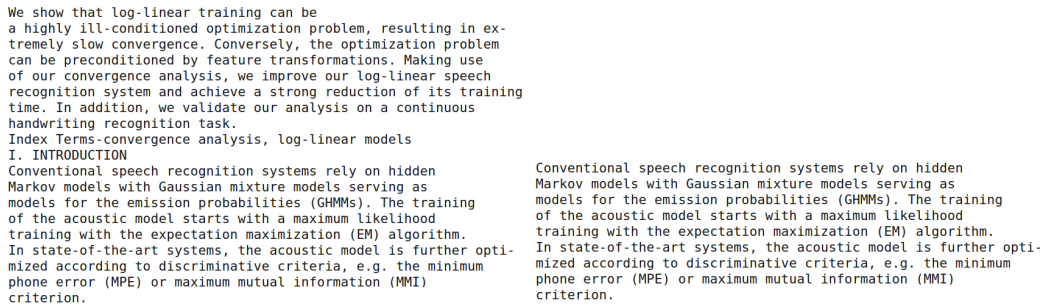
nako, ako pri predchádzajúcom kroku je potrebné zadať regulárny výraz, ktrorý ohraničuje záverečnú časť textu, ktorá sa vymaže.

Súbory prislúchajúce k PDF dokumentu sú kontrolované a pri nájdení posledného výskytu zadaného regulárneho výrazu je ponechaný text až po tento regulárny výraz. To znamená, že ak používateľ chce zmazať napríklad odkazy na použitú literatúru, ako regulárny výraz určí reťazec „references“. Jednotlivé súbory sa prechádzajú od posledného po prvý (od záveru dokumentu). Ak odkazy začínajú na siedmej strane z ôsmich, výraz je na tejto strane nájdený a nasledujúci text je vymazaný. Rovnako ako ôsma strana dokumentu.

Vymazanie úvodu každej strany dokumentu

Táto možnosť je zavedená, pretože pôvodné dokumenty často obsahujú záhlavia, ktoré sa objavujú na každej strane. Sú nevyužiteľné a skresľujú výsledky štatistík. Postupne sa kontrolujú všetky súbory a pri prvom výskyte zadaného výrazu je zo súboru zmazaný tento výraz a aj predchádzajúci text na strane.

Príklad, pri ktorom používateľ určil, že úvod sa má vymazať po výraz „INTRODUCTION“:

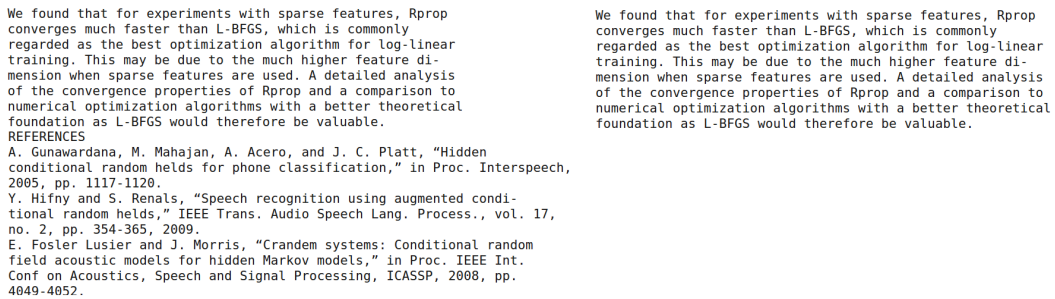


Obrázok 4.3: Vymazanie úvodu strany. Zadaný výraz: „INTRODUCTION“

Vymazanie záveru každej strany dokumentu

Vymazanie záveru je potrebné kôli výskytu zápätí, ktoré sa môžu opakovať a je nutné sa ich zbaviť z rovnakého dôvodu ako pri záhlaviach. Podobne ako pri predchádzajúcom kroku je kontrolovaný každý súbor a pri poslednom výskyte zadaného reťazca na strane je zvyšok textu zároveň s reťazcom vymazaný.

Príklad, pri ktorom používateľ určil, že záver sa má vymazať od výrazu „REFERENCES“:



Obrázok 4.4: Vymazanie záveru strany. Zadaný výraz: „REFERENCES“

Vymazanie každého výskytu reťazca zo všetkých strán dokumentu

Rovnako ako pri predchádzajúch dvoch krokoch je kontrolovaný každý súbor, avšak pri výskyte používateľom zadaného výrazu je vymazaný iba tento výraz. Má to opodstatnenie napríklad pri menách, alebo reťazcoch, ktoré by nasledujúcimi pravidlami nemuseli byť odstránené, avšak v texte sú nežiaduce.

Vymazanie riadku na ktorom sa nachádza zadaný výraz

Ak sa predpokladá, že zápätie sa nachádza na konci jednotlivých strán výstupu, bolo by možné nastaviť vymazanie celého zvyšku textu od určitého reťazca (napríklad od prvých slov zápätia). Takto by sa vymazal nežiaduci koniec strany. Ako už bolo spomenuté, pri spracovaní dokumentu s viac stĺpcami je možné, že bude zápätie spracované skôr ako druhý stĺpec textu. Pri takomto vymazávaní by mohla byť nechtiac odstránená podstatá časť textu

Preto je vhodné vymazať riadok (vetu), ktorá obsahuje daný reťazec, avšak nasledujúce riadky nechať nezmenené. Týmto opatrením sa vymaže zápätie, alebo iný nežiaduci riadok (veta), ale nasledujúci text je ponechaný.

Príklad, pri ktorom používateľ určil, že sa majú vymazať riadky obsahujúce výraz „exp“:

```
A. Discriminative Training of Log-Linear Parameters
The frame level objective function is
Jr Z Z ws1OgpA(St|xt) exp
exp r:1 15:1
exp Alais + otst
P/\(Si|33i) exp I (5)
ZS, exp ()\;;13s + otsf
for a fixed alignment sf where the state parameters are
TA is the regularization parameter to increase
robustness and avoid over-fitting. ws are state weights which
could be tuned to give less weight to some states e.g. silence
which occupies a large number of states in the alignment.
```

```
A. Discriminative Training of Log-Linear Parameters
The frame level objective function is

for a fixed alignment sf where the state parameters are
TA is the regularization parameter to increase
robustness and avoid over-fitting. ws are state weights which
could be tuned to give less weight to some states e.g. silence
which occupies a large number of states in the alignment.
```

Obrázok 4.5: Vymazanie riadkov s výrazom „exp“

Vymazanie nevhodných reťazcov

Všetky nasledujúce pravidlá sú aplikované na jednotlivé samostatné súbory, reprezentujúce stranu pôvodného PDF súboru (alebo internetovú stránku, obrázok).

Prvým krokom je vymazanie riadkov, ktoré neobsahujú žiadne písmeno abecedy. Teda ak nenastane na danom riadku zhoda s regulárnym výrazom [a-zA-Z], daný riadok je vymazaný. Týmto spôsobom sa odstránia rôzne nechcené riadky, ktoré obsahujú napríklad číslo strany, alebo iba postupnosť nepotrebných znakov. Napríklad „(87#+}“. Ukážka z textu je zobrazená na obrázku:

```
posterior probabilities of the form
P/\(S|fL') I N 7 (1)
2 exp (2 A§,i.f..<  )
5 11:1
15:1
/\(0|') 2 3 7 (1)
and
11:1
1 1
where the components of
 1  + , 1" '->(1(9)»---»(5')) (2)
are called feature functions.
```

```
posterior probabilities of the form

and

where the components of

are called feature functions.
```

Obrázok 4.6: Vymazanie nevhodných riadkov

Ako bolo spomenuté, z textu je potrebné vymazať aj e-mailové adresy. Odstránené sú reťazce obsahujúce znak „@“, ktorý je pre tieto adresy charakteristický. Podobne sú vymazané aj odkazy na internetové stránky, teda reťazce obsahujúce reťazce „www“ alebo „http“.

V texte sa nachádza aj množstvo zátvoriek, ktoré sú pozostatkom z rovníc, alebo algoritmov. Často neobsahujú žiadny znak abecedy, ale iba čísla a rôzne špeciálne znaky. Ďalším prípadom zátvoriek, ktoré treba vymazať sú odkazy na zdroje. Tie sú uvedené v hranatých zátvorkách ako napríklad „[1]“. Všetky spomenuté reťazce sú odstránené.

Nasleduje odstránenie riadkov, ktoré obsahujú menej ako štyri znaky. Tieto riadky sú často pozostatkom úprav z predchádzajúcich krokov a nemajú žiadne ďalšie využitie. Ukážka:

```
The regularized MMI criterion is often regarded as the          The regularized MMI criterion is often regarded as the
natural training criterion for log-linear models. In this Work, natural training criterion for log-linear models. In this Work,
We consider the MMI criterion with regularization:              We consider the MMI criterion with regularization:
T
1 C
PRS
T Z
s,n
and
o2I
Here is the training sample and C > 0 is the                    Here is the training sample and C > 0 is the
regularization constant. We refer to the minimization of F as   regularization constant. We refer to the minimization of F as
log-linear training.                                            log-linear training.
```

Obrázok 4.7: Vymazanie nevhodných riadkov

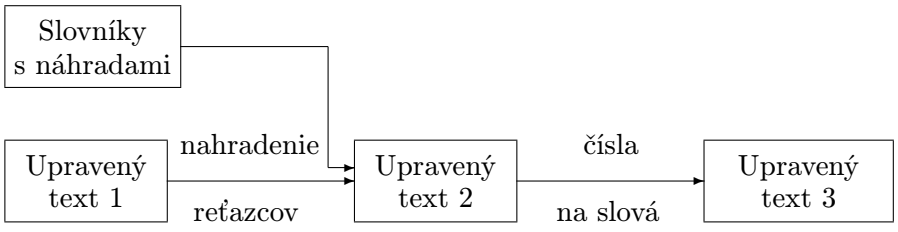
V ďalšom kroku sú vymazané dlhšie postupnosti medzier a znakov, ktoré nepatria do skupiny [a-zA-Z], napríklad je vymazaný reťazec „46 2 65 54-5 6“.

Potrebným krokom je aj vymazanie postupností čísel, ktoré sú dlhšie ako šesť znakov. Rovnako sú odstránené postupnosti iba čísel a medzier, ktorých dĺžka je väčšia ako 15 znakov. Napríklad „12 5 4 52 654 789 5 7“. Takéto postupnosti sú vo väčšine prípadov pozostatky z rovníc, tabuliek diagramov a grafov. Preto nemajú žiadne využitie.

Výstup skriptov po všetkých predchádzajúcich krokoch je zobrazený na obrázku B.3 v dodatku B.

4.2 Aplikovanie normalizačných pravidiel špecifických pre jazyk

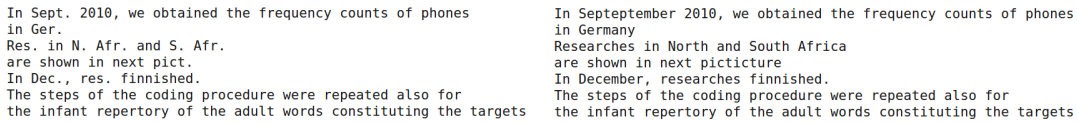
Z celkového postupu je to fáza:



Obrázok 4.8: Určenie fázy normalizácie

Ako prvé je vykonané nahradenie skratiek za celé verzie daných slov. V súbore *abbreviations.wlist* sa nachádza zoznam skratiek a k nim prislúchajúcich slov. Každý textový súbor

sa prechádza slovo po slove a pokiaľ sa vyskytuje reťazec v zozname skratiek, do textu sa vypíše jeho plnohodnotná náhrada. Do tohoto súboru môže používateľ pridať vlastné skratky, poprípade vymazať už existujúce. Na obrázku je možné vidieť príklad:



Obrázok 4.9: Príklad nahradzovania skratiek

Kedže text získaný nástrojom pdftotext má rozdieľne riadkovanie ako text získaný metódou OCR, je potrebné riadkovanie zjednotiť. Riadky, ktoré nekončia jedným zo znakov „:“, „?“, „!“ alebo „.“, sú spojené s ďalším riadkom. Nasleduje druhá časť kroku, ktorá rozdeluje viac viet na jednom riadku do viacerých samostatných riadkov. Ak sa v texte nachádzajú tri znaky regulárneho výrazu [a-zA-Z] a po nich nasleduje jeden zo znakov „:“, „?“, „!“ alebo „.“, je do textu vložený znak nového riadku. Každá veta je teda samostatne na jednom riadku a pri ďalších krokoch bude práca s riadkom pokladaná za prácu s vetou a naopak. Postup tejto úpravy je zobrazený na obrázkoch B.4 a B.5 v dodatku B.

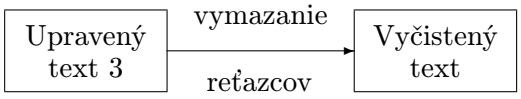
Ďalej je na text aplikované pravidlo, ktoré nahraďuje čísla za reťazce, ktoré ich reprezentujú. Môže sa jednať o čísla vyjadrujúce počet, alebo o dátumy. V implementovanom riešení bol použitý rovnaký skript ako v pôvodnom riešení. Tento skript dostatočne spĺňa potrebné podmienky na úpravu. Nejedná sa teda o vlastnú prácu. Príklad použitia: „345“ sa zmení na „three hundred and forty five“.

Nasleduje aplikovanie pravidla, podľa ktorého sa menia symboly na k nim prislúchajúce slová. Podobne ako pri zmene skratiek na slová, jednotlivé symboly zo vstupného textového súboru sú vyhľadané v súbore s pravidlami *symbols.dict*. Pokiaľ sa v súbore nachádza pravidlo pre symbol, do textového súboru je zapísané výsledné slovo. Súbor s pravidlami je prístupný pre používateľa a pred spustením skriptov ho môže upraviť podľa svojich požiadaviek.

Ukážka výstupu skriptov po všetkých predchádzajúcich krokoch je zobrazená na obrázku B.6 v dodatku B.

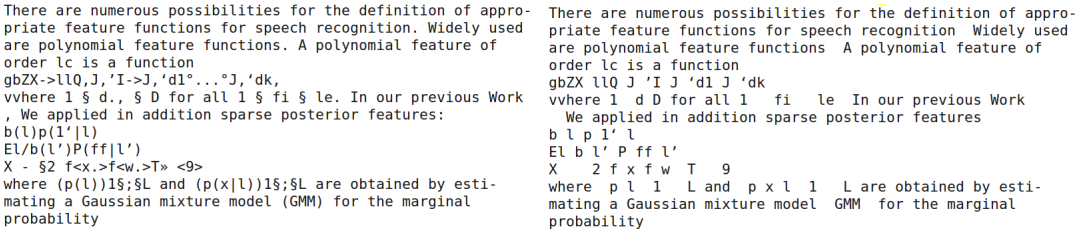
4.3 Vymazanie posledných nevhodných reťazcov

Z celkového postupu je to fáza:



Obrázok 4.10: Určenie fázy normalizácie

Po nahradení vybraných symbolov je potrebné vymazať z dokumentu všetky znaky, ktoré v angličtine netvoria slová. Týmto krokom sa vyčistí dokument od množstva nežiaduceho textu. Jedinými ponechanými znakmi sú znaky abecedy, teda [a-zA-Z], apostrof „’“ a spojovník „-“, pretože môžu byť súčasťou anglických slov. Spojovník slúži na spájanie viacerých slov, napríklad „back-off“ alebo „self-esteem“. Apostrof sa často používa pri skrátení alebo spojení viacerých slov, napríklad „don’t, isn’t“. Príklad:

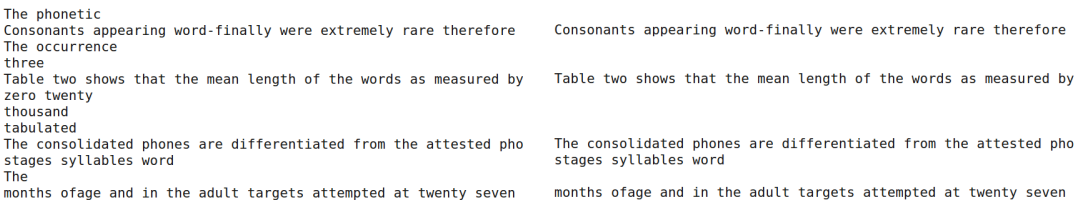


Obrázok 4.11: Vymazanie nevhodných znakov

Kedže okraje tabuliek sa technológiou OCR nespracujú vždy ideálne, zostávajú v texte skupiny nevhodných postupností, napríklad „i_i_i_i_i_i“. Takéto postupnosti často obsahujú aj znaky, ktoré boli vymazané predchádzajúcim pravidlom. Preto nasleduje zmazanie postupnosti rovnakých znakov až v tejto fáze úprav. Z predchádzajúceho prípadu po úprave teda vznikne reťazec „iiiiii“, ktorý sa môže odstrániť.

Zároveň sú odstránené slová dlhšie ako 15 znakov, ktoré boli pri testovaní vo väčšine prípadov nesprávne. Takéto slová vznikajú napríklad prerobením rovníc do podoby textu, a následným vymazaním nežiaducich znakov pomocou jedného z predchádzajúcich pravidiel. Ďalší zdroj takýchto dlhých reťazcov sú už spomínané okraje tabuliek.

Po týchto úpravách sa pravdepodobne v texte nachádzajú riadky, z ktorých bol vymazaný väčší počet (väčšina) slov. Pokiaľ na riadkoch zostali dve alebo jedno slovo, riadky sú zo súboru odstráné. Ukážka:



Obrázok 4.12: Vymazanie riadkov s menej ako 3 slovami

Ukážka textu po aplikovaní všetkých predchádzajúcich krokov je na obrázku B.7 v dodatku B.

Predchádzajúce pravidlá zamerané na vymazávanie riadkov, slov a znakov vytvorili na mnohých miestach nadbytočné medzery, ktoré je potrebné odstrániť. Na to slúži skript, ktorý nahradí postupnosť viacerých medzier jednou. Pokiaľ sa medzery nachádzajú na začiatku riadku, sú všetky odstránené.

Posledným krokom pred vytvorením vytvorením štatistík textu je odstránenie prázdnych riadkov. Tie mohli vzniknúť už pri konverzii pôvodného súboru do podoby textu, alebo pri niektorej z predchádzajúcich úprav. Ukážka textu po aplikovaní všetkých predchádzajúcich krokov je na obrázku B.8 v dodatku B.

Kapitola 5

Vytvorenie, úprava a aplikovanie slovníkov s pravidlami

Po aplikovaní normalizačných krokov sú posledné verzie jednotlivých textových súborov postupne zapísané do jedného súboru. Pre každý vstupný adresár (určený používateľom v súbore *config.sh*) je tak vytvorený celkový upravený textový súbor. Pokiaľ používateľ pred spustením skriptov určil štyri adresáre, v ktorých sa nachádzajú vstupné súbory na úpravu, vzniknú štyri celkové upravené textové súbory.

Zozbieranie textov a získanie štatistiky

Z týchto súborov je ďalej získaná štatistika všetkých slov v danom adresári, ktoré sa v dokumente vyskytujú a neboli vymazané jedným zo skriptov. Štatistika je zoradená podľa najčastejších slov až po najzriedkavejšie. Ďalší zobrazený údaj ku každému slovu je percento početnosti. To je vypočítané ako:

$$percento\_pocetnosti\_slova = \frac{pocetnost\_slova}{celkovy\_pocet\_slov} * 100$$

Každý riadok štatistiky teda obsahuje:

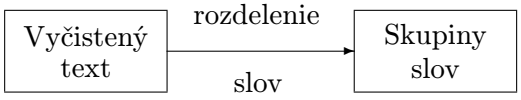
„slovo | početnosť | percento početnosti“

Napríklad:

„and | 1678 | 5.08993842“

5.1 Rozdelenie slov do skupín

Z celkového postupu je to fáza:



Obrázok 5.1: Určenie fázy normalizácie

Slová s ktorými sa manipuluje sú rozdelené do piatich skupín. Toto rozdelenie slúži na urýchlenie práce s pravidlami. Pre slovo, ktoré patrí do určitej skupiny bude hľadané

pravidlo iba v súboroch s pravidlami pre danú skupinu. Pri práci s veľkým množstvom súborov sa môžu spracovávať tisíce, alebo desaťtisíce rôznych slov. Vyhľadávanie pravidiel v jednom súbore s pravidlami, ktorý by obsahoval ďalšie tisíce záznamov by bolo príliž zdĺhavé.

Rozdelenie slov podľa ich vlastností je vhodné aj kôli tomu, že s každou skupinou slov sa pracuje odlišným spôsobom. Jednotlivé slová su rozdelené do skupín rovnakým spôsobom ako v dodanom riešení:

lower

Slová, ktoré obsahujú iba malé znaky. Dajú sa popísať regulárnym výrazom [a-z]+. Kedže väčšina slov v dokumentoch je písaná malými písmenami, je to najpočetnejšia skupina.

UPPER

Výrazy patriace do tejto skupiny sú zložené iba z veľký písmen abecedy, teda [A-Z]+. V textoch sa vyskytujú ako skratky, alebo nadpisy.

Fupper

Sú to slová začínajúce s veľkým písmenom, po ktorom nasledujú iba malé písmená. Regulárny výraz popisujúci túto skupinu je [A-Z][a-z]+. Tieto slová sa nachádzajú väčšinou na začiatku riadkov, poprípade sa jedná o mená.

other

Výrazy zaradenené do tejto skupiny musia obsahovať okrem písmen abecedy aj spojovník „-“, alebo apostrof „’“. Ako bolo spomenuté, tieto znaky sú súčasťou niektorých anglických slov.

unknown

Do tejto skupiny patria všetky zvyšné slová. Sú zložené z kombinácie veľkých a malých písmen, pričom veľké písmeno nemusí byť iba na začiatku slova. Sú sem priradené slová, ktoré nespĺňajú podmienky predchádzajúcich skupín. Do tejto skupiny patria napríklad „PhD“, „kHz“ alebo „dB“.

5.2 Rozdelenie slovníkov

Spolu so skriptami na úpravu textu obsahuje riešenie aj sadu základných pravidiel a zoznam správnych a nesprávnych slov. Pre každú skupinu slov sa tu nachádzajú tri súbory:

Slovník so zozbieranými pravidlami

Tieto pravidlá špecifikujú za aký výsledný výraz sa má slovo v texte zameniť. Každé pravidlo sa nachádza na samostatnom riadku a má podobu:

„pôvodný reťazec | výsledný reťazec | početnosť pôvodného slova | percento početnosti pôvodného slova“

Pri spracovaní jednotlivých textov sa aktualizuje hodnota „početnosť pôvodného slova“ a „percento početnosti pôvodného slova“. Tieto hodnoty reprezentujú štatistiky zo všetkých doteraz spracovaných textov. Jedným z pravidiel je napríklad:

„The | the | 709 | 0.53981“

Toto pravidlo vyjadruje, že každý reťazec „The“, ktorý sa spracuje v texte, sa má nahradiť reťazcom „the“. Vo všetkých spracovaných textoch sa tento výraz nachádzal 709 krát, čo je takmer 0.54% zo všetkých slov.

Výsledné slovo nemusí byť iba jedno. Ak sa v texte nachádza slovo, ktoré vzniklo napríklad nesprávnym spojením dvoch reťazcov, ako „doorand“, môže existovať pravidlo:

„doorand | door and | štatistika. . . “

Slovník zlých slov

Obsahuje reťazce, ktoré v texte nemajú byť, pretože sú chybné. Každý riadok obsahuje jeden záznam. Napríklad „abRevation“ alebo „quetion“.

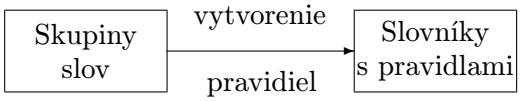
Slovník dobrých slov

Každý riadok obsahuje slovo, ktoré je správne a v upravenom texte sa môže nachádzať v danej podobe. Patria sem napríklad „Hollywood“, „VoiceXML“ alebo „dog“. Tieto reťazce slúžia na vytvorenie nových pravidiel.

Pokiaľ sa výraz z textu nenachádza v slovníku so zozbieranými pravidlami, ale v slovníku dobrých slov áno, je vytvorené nové pravidlo. To určí, že dané slovo bude v texte ponechané. Podrobnejší popis vytvárania pravidiel je popísaný v nasledujúcej časti práce.

5.3 Vytvorenie nových pravidiel využitím slovníkov

Z celkového postupu je to fáza:



Obrázok 5.2: Určenie fázy normalizácie

Jednotlivé záznamy zo štatistiky slov sú použité na vytvorenie nových pravidiel do slovníkov so zozbieranými pravidlami. Skript postupne prechádza jednotlivé položky v zozname a určí do ktorej skupiny slov patrí. Pravidlá určené na skontrolovanie používateľom sú vytvorené v samostatnom priečinku. Podľa skupiny sa s daným výrazom pracuje nasledovne:

skupina lower

Spracovanie skupiny lower je v porovnaní s ostatnými najjednoduchšie. Pre dané slovo sa vyhľadá pravidlo v súbore s pravidlami pre skupinu lower. Ak je pravidlo nájdené, pokračuje sa s ďalšími slovami. Pokiaľ takéto pravidlo neexistuje, slovo sa vyhľadá s zozname zlých lower slov. V prípade, že sa v tomto zozname slovo našlo, nasleduje spracovanie ďalších reťazcov a pre toto slovo nie je vytvorené žiadne pravidlo. Ak bolo aj toto vyhľadávanie neúspešné, kontroluje sa, či sa reťazec nenachádza v zozname dobrých lower slov.

Pokiaľ bolo slovo nájdené, je vytvorené pravidlo, ktoré vyjadruje, že daný reťazec je správny a nie je potrebné ho nahradzovať. V danom pravidle budú teda pôvodné slovo a náhrada totožné.

Posledná možnosť je, že sa slovo nenachádza ani v jednom zo zoznamov. V tom prípade je zapísané do súboru s pravidlami, ktoré musí skontrolovať používateľ. V pravidle je pôvodné slovo rovnaké ako jeho prípadná náhrada a takisto je vypísaná štatistika slova. Pravidlo pre neznáme slovo vyzerá teda takto:

„slovo | slovo | početnosť | percento početnosti“

Je pridané do súboru *lower_handmade.wlist*.

skupina unknown

Postup pri tejto skupine je veľmi podobný ako pri skupine lower. Pre reťazec sa najprv hľadá pravidlo. Ak neexistuje, prechádza sa zoznamom nesprávnych unknown výrazov a pri úspechu je výraz ignorovaný. Pri neúspechu sa prehľadáva zoznam správnych unknown slov a ak je nájdené, vytvorí sa nové unknown pravidlo. Ak nie je nájdené, vytvorí sa pravidlo, ktoré musí používateľ skontrolovať. Na to slúži súbor *unknown_handmade.wlist*.

skupina Fupper

Slová začínajúce veľkým písmenom sú najprv vyhľadané v súbore s pravidlami pre túto skupinu. Ak je hľadanie úspešné, nasleduje spracovanie ďalších reťazcov. Ak pravidlo neexistuje, podobne ako pri skupine lower, prechádza sa najprv zoznam zlých a v prípade neúspechu dobrých Fupper slov. Ak sa slovo nachádza vo zozname zlých slov, pokračuje sa ďalším výrazom. Ak sa nachádza v zozname dobrých slov, je vytvorené pravidlo.

Ak sa výraz nenachádza ani v jednom zo zoznamov, prvé, teda veľké písmeno je zmené na malé. Tento nový reťazec sa vyhľadáva v zozname zlých slov pre skupinu lower. V prípade úspechu sa pokračuje spracovávanie a výraz sa ignoruje. Inak sa prechádza zoznam správnych lower reťazcov. Pri náleze je vytvorené pravidlo pre skupinu Fupper, ktroré vyjadruje, že pôvodné slovo sa zmení na verziu s malým začiatočným znakom. Napríklad

„Force | force | štatistika...“

Pridané su aj potrebné štatické údaje (početnosť a percento početnosti). Pokiaľ nebol ani takto upravený výraz nájdený v zoznamoch, je rovnakým spôsobom ako pri skupine lower vytvorené pravidlo, ktoré je určené na upravenie používateľom. Je uložené do súboru *Fupper_handmade.wlist*.

Pre lepšiu predstavu o postupe pri vytváraní pravidiel slúži nasledujúci pseudokód. Popisuje kroky ktoré sú vykonané pri tvorbe pravidla pre slovo patriace do tejto skupiny:

```
if (word in 'rules_for_Fupper')
{
  create_dict(word);
}
else if (word in 'Fupper_bad_list')
{
  break;
}
else if (word in 'Fupper_good_list')
```

```
{
  create_dict(word);
}
else
{
  if (lower_case(word) in 'rules_for_lower')
  {
    create_dict(word);
  }
  else if (lower_case(word) in 'lower_bad_list')
  {
    break;
  }
  else if (lower_case(word) in 'lower_good_list')
  {
    create_dict(word);
  }
  else
  {
    add_to_Fupper_handmade(word);
  }
}
```

Pseudokód 5.1: Postup pri tvorbe pravidiel pre slová skupiny Fupper

skupina UPPER

Pri tejto skupine je rovnaký postup ako pri skupine Fupper. Avšak najprv sa prehľadávajú pravidlá a zoznamy pre skupinu UPPER. Pri neúspechu je zmenené celé slovo na malé. Je hľadané v pravidlách a zoznamoch pre lower slová. Pravidlá sú vytvárané rovnakým spôsobom ako pri predchádzajúcej skupine. Pravidlá pre reťazce, ktoré neboli nájdené ani v jednom slovníku sú uložené v súbore *upper_handmade.wlist*.

skupina other

Skupina other je z hľadiska porovnávania najnáročnejšia. Rovnako ako v pri predošlých skupinách je výraz hľadaný v pravidlách pre skupinu other, v zozname zlých a nakoniec v zozname dobrých slov.

V prípade neúspechu nie je pre slovo vytvorené pravidlo určené na kontrolu používateľom. Ak reťazec obsahuje znak „-“, je vysoká pravdepodobnosť, že je zložený z viacerých spojených slov. Preto je takýto reťazec rozdelený na dva výrazy a ako ich hranica slúži práve spojovník. Pre takto vzniknuté nové slová prebieha ďalšie vyhľadávanie. V zozname pravidiel a zozname dobrých a zlých výrazov pre prislúchajúcu skupinu.

Ak nový výraz patrí do skupiny Fupper alebo UPPER, pri nenájdení reťazca v pravidlách a zoznamoch sa slová zmenia na malé. Rovnako ako je to pri samostatných slovách patriacich do týchto skupín. Následne sú prehľadávané pravidlá a zoznamy pre skupinu lower. Ak prvá časť pôvodného výrazu nie je nájdená ani v jednom z predchádzajúchi krokov, pre pôvodné slovo je vytovrené pravidlo určené pre používateľa.

Ak bola prvá časť slova nájdená v zozname nesprávnych slov, pôvodný výraz je ignorovaný. Ak bola prvá časť nájdená v pravidlách, alebo v zozname dobrých slov, nasleduje spracovanie druhej časti slova. V prípade, že aj druhá časť reťazca je správna, je vytvorené nové pravidlo do zoznamu zozbieraných pravidiel. Ak je prvá časť správna a druhá nesprávna, pôvodné slovo je ignorované a z textu bude vymazané.Ak sa druhá časť slova nenachádza v pravidlách a zoznamoch, je vytvorené pravidlo pre používateľa. Tieto pravidlá sú v súbore *other_handmade.wlist*.

5.4 Úpravy pravidiel a vytvorenie kontextu

Pokiaľ bolo vyhľadávanie v pravidlách úspešné, je potrebné zmeniť početnosť slova, pre ktoré platí pravidlo. K pôvodnej početnosti, ktorá je súčasťou pravidla, je prirátaná početnosť slova v spracovanom texte (určená v súbore so štatistikou). Nová hodnota je zapísaná namiesto pôvodnej. V prípade, že pre reťazec pravidlo neexistovalo a bolo vytvorené, početnosť slova je rovnaká ako početnosť slova v spracovanom texte. V prípade, že pre niektoré slovo nebolo vytvorené pravidlo, znamená to, že dané slovo je nesprávne a vo výslednom texte sa neobjaví.

Po vytvorení pravidiel teda vzniklo pre každý vstupný priečinok päť súborov s pravidlami pre reťazce, pre ktoré nebolo jednoznačne určené, či sa jedná o slová zlé. Tie by mali byť z textu odstránené, poprípade upravené. Ak sú to dobré slová, majú v texte zostať.

Kedže pravidlá obsahujú často slová, ktorých význam nemusí byť vždy zjavný, je potrebné, aby bol k jednotlivým slovám pridaný aj kontext v ktorom sa nachádzajú. To znamená, že na každom riadku je po štatistických údajoch zobrazený aj reťazec s daným a s maximálne štyrmi okolitými slovami, ako sa objavujú v texte. Tieto reťazce vyzerajú nasledovne:

„predchádzajúce_slovo1 predhádzajúce_slovo2 slovo nasledujúce_slovo1 nasledujúce_slovo2“

Zobrazených je maximálne päť výskytov daného slova v texte. Ak sa slovo vykytuje v texte menej ako päť krát, je zobrazený každý výskyt slova. V prípade, že je slovo použité v texte viac ako päť krát, reťazce ktoré sa zobrazia ako kontext sú náhodne vybrané z celého textu. Všetky výskyty sú vypísané náhodnom poradí.

Ak je neznámy reťazec, pre ktorý je vytvorené pravidlo napríklad „comorbidities“, kontext môže vyzerať:

„’identified extensive comorbidities associated with’ ’and quantify comorbidities within and’ ’Mental health comorbidities in the’ ’four Physical comorbidities in the’ ’mental health comorbidities ’“

Ak sa slovo nachádza na začiatku alebo na konci vety, sú zobrazené iba nasledujúce, respektíve predchádzajúce slová. Vo výnimočných prípadoch môžu byť zobrazené napríklad iba jedno predchádzajúce, alebo nasledujúce slovo.

5.5 Opravenie chýb vzniknutých nesprávnym rozpoznaním znakov

Ako už bolo spomenuté, pri konverzii z iného formátu do podoby textu môžu byť jednotlivé znaky nesprávne rozpoznané a následne zobrazené. Tento problém sa najviac prejavuje pri použití metódy OCR.

Na odstránienie týchto nedokonalostí bol implementovaný nasledujúci skript. Súbor *ocr_fix.wlist* obsahuje zoznam najčastejších chýb pri rozpoznávaní a k nim prislúchajúce opravy. Napríklad, ak je chybne rozpoznané písmeno „w“, v texte sú zobrazené znaky „vv“. Pravidlo v tomto prípade vyzerá nasledovne:

„vv | w“

Jednotlivé prvky *_handmade.wlist* súborov sú postupne spracované a vyhľadáva sa v nich reťazec z lavej časti pravidiel z *ocr_fix.wlist*. Ak sa v slove tieto reťazce nenachádzajú, pravidlo na nijak neupravuje. Ak nastala zhoda, tento reťazec je v slove nahradený pravou stranou pravidla.

Novovzniknuté slovo je porovnávané s existujúcimi pravidlami a zoznamom správnych slov. Porovnanie so zoznamom nesprávnych slov nie je potrebné. Ak nie je nové slovo medzi pravidlami, alebo správnymi reťazcami, je ignorované. Podobne ako pri vytváraní pravidiel sú jednotlivé novovzniknuté výrazy spracované podľa príslušnosti pôvodného slova k skupine:

lower

Výrazy patriace do tejto skupiny sú najprv hľadané v zozname lower pravidiel. Ak takéto pravidlo existuje, vytvorí sa nové pravidlo pre používateľa, ktoré má na ľavej strane pôvodné slovo a na pravej strane výsledný výraz z lower pravidla. V prípade, že sa tu slovo nenachádza, je prehľadávaný zoznam správnych lower slov. Ak reťazec nie je nájdený, algoritmus pokračuje s ďalším slovom. V opačnom prípade sa vytvorí nové pravidlo do *lower_handmade.wlist*. Na ľavej strane pravidla bude pôvodné slovo a na pravej nový reťazec.

Napríklad, ak je pôvodné slovo „hnally“ a v súbore *ocr_fix.wlist* existuje pravidlo „h | fi“, novovzniknutý výraz je „finally“. Pre ten sa nájde pravidlo v zozname lower pravidiel, vyzerúce napríklad:

„finally | finally | štatistika. . . “

Alebo je vyhľadané v zozname dobrých slov. Nové lower pravidlo teda vyzerá:

„hnally | finally | pôvodná štatistika a kontext. . . “

Fupper

Nové reťazce sú najskôr vyhľadané v pravidlách a zozname správnych Fupper slov. V prípade úspešného vyhľadania sú vytvorené pravidlá rovnakým spôsobom ako pri lower slovách. Ak sa vo Fupper zoznamoch slovo nenachádza, všetky znaky sú zmenené na malé a nasleduje vyhľadávanie nového reťazca v pravidlách a zoznamoch pre skupinu lower. Ak je reťazec nájdený, je vytvorené nové pravidlo do *Fupper_handmade.wlist*, meniace pôvodné Fupper slovo s OCR chybou na nové lower slovo bez chyby.

unknown, UPPER

Tieto dve skupiny sú spracované rovnako ako skupina Fupper, avšak jednotlivé porovnávania sú vykonané s pravidlami a zoznamami skupiny unknown, respektíve UPPER. Rovnako je vykonaná úprava slov na lower v prípade neúspešného vyhľadávania pôvodnej podoby slova.

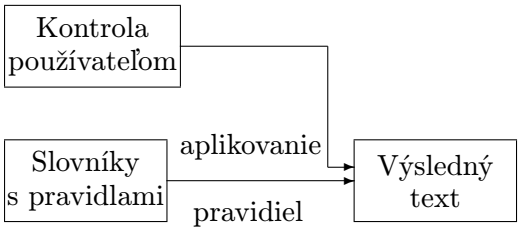
other

Slová patriace do tejto skupiny sú porovnávané s pravidlami a zoznamom dobrých slov skupiny. Pri nájdení slova je vytvorené pravidlo rovnako ako pri ostatných skupinách. Pri neúspechu sa kontroluje, či slovo obsahuje spojovník. Ak ano, je rozdelené na dve časti, rovnakým spôsobom ako pri vytváraní pravidiel pre túto skupinu. V oboch častiach slova sú postupne nahradzované chybné znaky za správne a tieto novovzniknuté výrazy sú porovnávané s pravidlami a zoznamami ostatných skupín.

Ak výraz nebol nájdený v pravidlách, alebo zoznamoch pre danú skupinu, je zmenený na lower a prebieha vyhľadávanie v súboroch pre skupinu lower. Týmto spôsobom sú nájdené jednotlivé varianty pre slová other, prípadne ich časti.

5.6 Upravenie nových pravidiel používateľom a ich aplikovanie na text

Z celkového postupu je to fáza:



Obrázok 5.3: Určenie fázy normalizácie

Súbory s pravidlami pre používateľa obsahujú okrem nových pravidiel aj všetky pôvodné, ktoré boli v súbore uložené pred aplikovaním posledného kroku. Používateľ má preto na výber medzi pôvodným pravidlom a novými variantami (vzniknutými pomocou *ocr_fix.wlist*). Nesprávne varianty vymaže a správna sa uloží do zoznamu pravidiel.

Ešte pred samotnou úpravou pravidiel sú jednotlivé *_handmade.wlist* súbory skopírované do nových súborov. Tie majú pomocnú funkciu. Používateľ si po úpravách v _handmade.wlist súbore môže skontrolovať zmeny, poprípade opraviť nesprávne vymazadané pravidlo. Prvý skript je ukončený a používateľ môže vymazať, alebo upraviť pravidlá z *_handmade.wlist* súborov.

Posledná časť práce s pravidlami nastane po spustení skriptu *app_rules.sh*. Najprv sú skontrolované jednotlivé *_handmade.wlist* súbory. Každý riadok pravidla je pridaný do zozbieraných pravidiel pre danú skupinu. Okrem toho sa kontroluje, či sa ľavá strana pravidla odlišuje od pravej. Napríklad, ak používateľ zmenil pravidlo pre neznáme slovo „tabl | tabl“ na správny výraz „tabl | table“. Takéto upravené pravidlá sú skopírované do súborov *_handmade_app.wlist*. Používateľ môže skontrolovať zmeny ktoré vykonal v jednotlivých pravidlách.

Pravidlá sú zoradené podľa početnosti, aby sa znížil čas potrebný na nájdenie správneho pravidla pre slovo. Ak sa výraz nachádza v texte častejšie, bude častejšie použité jemu prislúchajúce pravidlo. Keď je umiestnené na začiatku zoznamu, nie je potrebné neúspešne prechádzať množstvo pravidiel a tým sa skript urýchli.

Po každom spracovaní textov a vytvorení nových praviel sa slovníky so zozbieranými pravidlami rozširujú o nové položky. Preto je pri spracovaní nasledujúcich textov vyššia

pravdepodobnosť, že pravidlo pre dané slovo existuje. Nutnosť kontrolovať slovníky dobrých a zlých slov a bude čoraz nižšia. Upravovanie nových pravidiel používateľom taktiež.

Aplikovanie zozbieraných gramatík na text

Každý textový súbor reprezentujúci jednu stranu pôvodného PDF dokumentu (alebo obrázok, internetoveú stránku) je postupne prechádzaný. Každému výrazu je priradená skupina. Podľa nej sa hľadá pre daný výraz pravidlo v slovníku. Ak sa tam pravidlo nachádza, výraz je v texte nahradený pravou stranou pravidla. Inak je slovo z textu vymazané. Príklad:

```
thirty A as L one 'rf ' ' IIL splitting IIL training IIL splitting IIL
training five IIL splitting discrim
training twenty eight minus rx discrim
training - LNo of densitiesstate LNo of densitiesstate L X
Q Q é é Q - l T El V one number of parameters Fig
three EPPS Comparison of WER of split and ML split log-linear models gr
The MLLR is a feature linear transform while cMLLR transforms the param
Their purpose is to re-move the speaker specific information
For the EPPS task SAT gives a WER improvement of three percent absolute
```

```
thirty a as model one splitting training splitting training five spl
training five splitting
training twenty eight minus
training of densities state of densities state model
model sat El one number of parameters
three EPPS comparison of of split and split models regression and co
the is a feature linear transform while transforms the parameters of
their purpose is to the speaker specific information
for the EPPS task sat gives a improvement of three percent absolute
```

Obrázok 5.4: Aplikovanie pravidiel na text

Aplikovaním pravidiel mohli byť na niektorých riadkoch vymazané všetky slová. Preto je na text aplikovaný ešte raz skript, ktorý vymazáva prázdne riadky. Výsledný text je zhromaždený do jedného súboru pre všetky pôvodné PDF súbory, textové súbory a obrázky umiestnené v vstupných adresároch. Pre každý adresár teda existuje jeden kompletný textový súbor. Ukážka finálneho výstupu skriptov je zobrazená na obrázku B.9 v dodatku B.

Z výsledných textových súborov je vytvorený zoznam dobrých slov pre každú skupinu. Tieto zoznamy môžu byť použité ako vstupné súbory v ďalšej časti práce, pri vytváraní výslovnostných slovníkov.

Kapitola 6

Vytvorenie a úprava výslovnostných slovníkov

Pri tvorbe výslovnostných slovníkov je použitá základná verzia slovníkov. Pre každú skupinu slov existuje jeden súbor so slovníkom. Každý záznam slovníka obsahuje slovo a výslovnosť. Príklad záznamu:

„Bill b ih l“, alebo „yes y eh s“

Do základnej verzii slovníkov patrí ešte súbor *all.dict*, obsahujúci všeobecný výslovnostný slovník. Jednotlivé položky sú v ňom písané velkým písmom. Tento slovník sa použije po neúspešnom hľadaní slov v slovníkoch pre špecifické skupiny. Záznam tu má rovnakú podobu ako v ostatných slovníkoch, ale s veľkými písmenami, napríklad „LIPS l ih p s“. Ako vstup sú použité zoznamy slov roztriedené podľa skupín.

6.1 Vyhľadanie slov v existujúcich slovníkoch a vytvorenie nových výslovností

Po spustení skriptu *create_dict.sh* sú jednotlivé vstupné slová vyhľadé v základnej verzii slovníkov. Nájdené záznamy budú použité do finálnej verzie. Ak sa tu slová nenachádzali, sú zmenené na velké a vyhľadané v slovníku *all.dict*. Pokiaľ bolo hľadanie úspešné, záznamy sú použité vo výslednej verzii.

Napríklad, ak slovo „national“ nebolo nájdene v slovníkoch pre skupinu lower, je zmenené na „NATIONAL“ a tento výraz sa hľadá v slovníku *all.dict*. Pri úspechu existuje záznam:

„NATIONAL n ae sh n el“

Nový záznam teda bude:

„national n ae sh n el“

Tento postup platí pre všetky skupiny slov.

Slová skupiny other, ktoré obsahujú spojovník sú však pri neúspešnom vyhľadaní v *all.dict* rozdelené práve týmto znakom na dva výrazy. Ak pre oba výrazy existuje výslovnosť v *all.dict*, je vytvorené nové pravidlo pre pôvodné slovo.

Napríklad reťazec „half-life“ sa nenachádza v slovníku pre skupinu other. Reťazec „HALF-LIFE“ sa nenachádza v slovníku *all.dict*. Reťazec je rozdelený na „HALF“ a „LIFE“. Tieto výrazy sa nachádzajú v *all.dict* ako:

„HALF hh ae f“ a „LIFE l ay f“

Nový záznam do pravidiel teda bude:

„half-life hh ae f l ay f“

Slová, pre ktoré neexistuje záznam v slovníkoch sú použité ako vstup do programu Sequitur G2P. Je to program s voľnou licenciou, ktorý bol použitý aj v pôvodných skriptoch a pre implementované riešenie sú jeho výsledky dostatočné. Ten vygeneruje nové *_handmade.dict* výslovnostné slovníky pre každú skupinu slov. Tieto slovníky sú určené na kontrolu používateľom.

Pre lepšie predstavu je uvedený pseudokód, vyjadrujúci ako sú vyhľadané, alebo vytvorené výslovnosti pre každé slovo skupiny other:

```
if (word in 'other.dict')
{
  create_dict(word);
}
else if (uppercase(word) in 'all.dict')
{
  create_dict(word);
}
else if ("-" in word)
{
  split(word);
  if ((upper_case(word_first_part) in 'all.dict') &&
      (upper_case(word_second_part) in 'all.dict'))
  {
    create_dict(word);
  }
}
else
{
  generate_pronunciation(word); # to 'other_handmade.dict'
}
```

Pseudokód 6.1: Postup pri vytvorení výslovnosti pre slová skupiny other

Pred kontrolou pravidiel používateľom je upravený výstup programu Sequitur G2P pre slová skupiny UPPER. Ak tieto slová neboli nájdené v predchádzajúcich slovníkoch, je pravdepodobné, že sa jedná o skratky. Napríklad „WTC“. Tie sa často vyslovujú inak ako normálne slová. Preto je pre slová z *upper_handmade.dict* slovníkov vytvorená nová výslovnosť. Každé písmeno je teda hláskované zvlášť. Nová výslovnosť je pridaná do *upper_handmade.dict* a pôvodná výslovnosť je ponechaná tiež.

6.2 Zozbieranie výslovností

Následne používateľ upraví alebo vymaže záznamy z vygenerovaných *_handmade.wlist* slovníkov. Po spustení skriptu *app_handmade.sh* sú zhromaždené všetky novovzniknuté pravidlá:

- Tie, ktoré vznikli vyhľadaním slov v pôvodných slovníkoch pre skupiny.
- Pravidlá vyhľadané v súbore *all.dict*.
- Vygenerované a následne používateľom upravené pravidlá.

Sú pretriedené, zoradené a zbavené prípadných vulgarizmov, ktoré sa mohli v textoch objaviť. Výsledné výslovnostné slovníky pre zdrojové zoznamy slov sú teda vytvorené.

Posledným krokom je spojenie týchto slovníkov so základnou verziou slovníkov, zoradenie a pretriedenie. Rozšírená verzia základných slovníkov môže byť použítá pri vytváraní výslovností pre ďalšie zoznamy slov.

Základné slovníky sú rozširované pri každom spracovaní súborov s neznámimi slovami. Postupne je tak neznámych slov čoraz menej. Program Sequitur G2P kôli tomu musí spracovať menej slov. Pretože generovanie výslovnosti je najpomalšia činnosť, jej obmedzením sa vytváranie slovníkov zrýchluje.

Kapitola 7

Vyhodnotenie výsledkov a porovnanie s pôvodným riešením

Pre vyhodnotenie výsledkov implementovaného riešenia bolo potrebné vytvoriť referenčné riešenie. To predstavuje očakávaný výsledok, ku ktorému sa jednotlivé výstupy mali čo najviac priblížiť. Aby bolo testovanie dostatočne hodnoverné, referenčné riešenie bolo vytvorené z 20 PDF súborov, pričom každý z nich mal 4 až 9 strán. Celkovo bolo použitých 118 PDF strán.

Vstupné dokumenty boli dodané spolu s pôvodným riešením a jeho výsledkami. Sú to zbierky dokumentov k rôznym témam, podľa ktorých sú rozdelené. Každá zbierka preto obsahuje špecifické prvky, ako záhlavie, zápätie a podobne. Aby vyhodnotenie pokrývalo čo najväčšiu časť zbierok, z každej boli použité väčšinou dva dokumenty. Pre ne bolo vytvorené referenčné riešenie.

Implementované riešenie bolo spustené vždy pre dané dva dokumenty, pričom boli nastavené parametre špecifické pre zbierku. Pôvodné riešenie extrahovalo text z dokumentov nástrojom pdftotext, alebo metódou OCR. Toto rozdelenie záviselo od jednotlivých zbierok. Dokumenty pre referenčné riešenie boli preto vybrané tak, že pre polovicu z nich bola v pôvodnom riešení použitá metóda OCR a pre druhú nástroj pdftotext. Tým vznikla možnosť porovnať riešenia aj v závislosti od toho, akým spôsobom bol získaný text z dokumentov.

7.1 Výpočet percentuálnej zhody pomocou Levenshteinovej vzdialenosti

Na porovnanie bol použitý modul jazyka Perl[1], ktorý vyhodnotí Levenshteinovu vzdialenosť [2] medzi výstupmi skriptov a referenčnými súbormi. Táto vzdialenosť vyjadruje počet úprav, ktoré je potrebné vykonať na reťazci, aby sa zhodoval s druhým reťazcom. Napríklad Levenshteinova vzdialenosť medzi reťazcami „kitten“ a „sitting“ je 3. Aby sa tieto reťazce zhodovali, sú potrebné tieto tri kroky:

- Nahradenie „s“ za „k“: vznikne „sitten“.
- Nahradenie „i“ za „e“: vznikne „sittin“.
- Vloženie „g“: vznikne „sitting“.

[1] http://search.cpan.org/~jgoldberg/Text-LevenshteinXS-0.03/LevenshteinXS.pm

Zo získanej vzdialenosti bola vypočítaná percentuálna zhoda s referenčným riešením vzťahom:

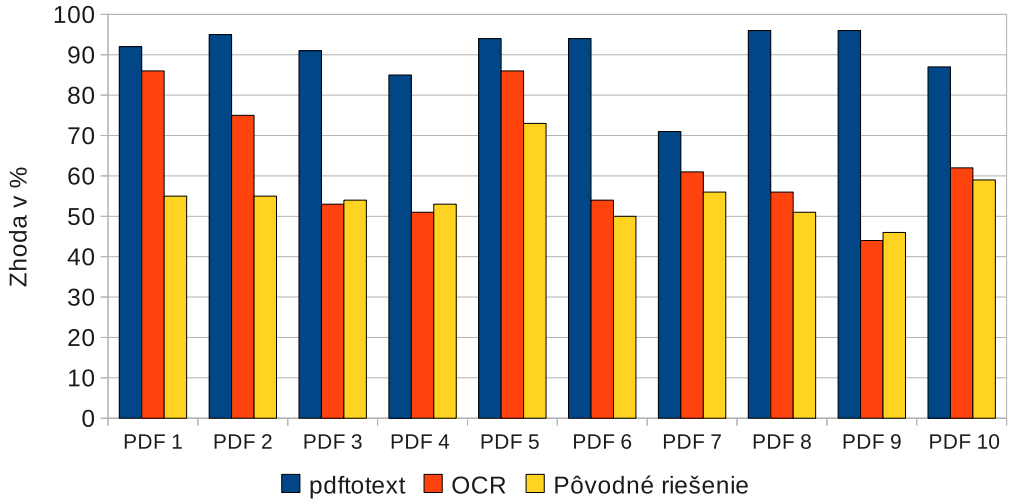
$$percentualna\_zhoda = (1 - \frac{Levenshteinova\_vzdialenost}{dlzka\_referencneho\_riesenia}) * 100$$

Percentuálna zhoda predchádzajúceho príkladu, pričom ako referencia je použitý reťazec „sitting“ je teda 57% a výpočet vyzerá nasledovne:

$$(1 - \frac{3}{7}) * 100 = 57\%$$

7.2 Analýza výsledkov a porovnanie s pôvodným riešením

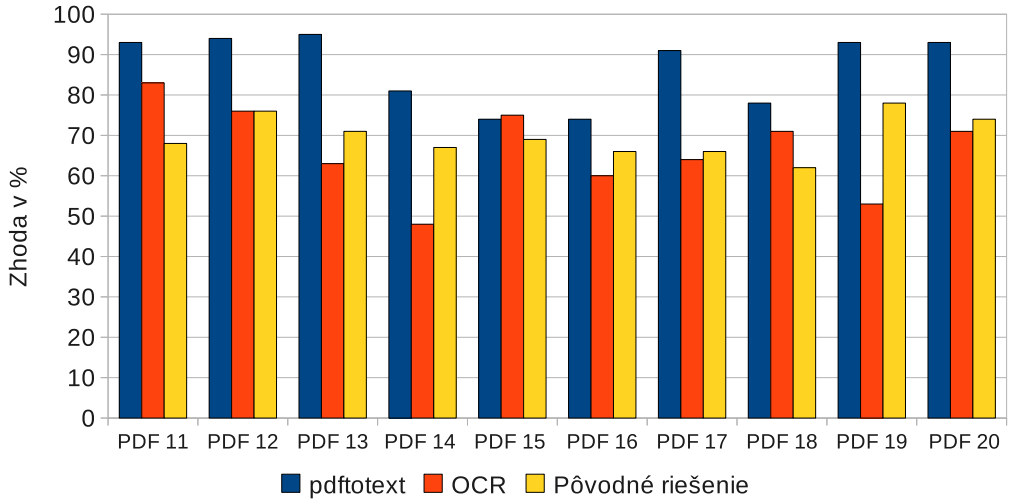
Implementované riešenie bolo spustené pre každý dokument dva krát a text bol extrahovaný metódou OCR aj nástrojom pdftotext. Na grafoch sú zobrazené percentuálne zhody výsledkov jednotlivých riešení s referenčným riešením. Prvý graf zobrazuje percento zhody s očakávaným výsledkom, pričom text bol v pôvodnom riešení získaný metódou OCR.



Obrázok 7.1: Percentuálna zhoda s ref. riešením. Pôvodné riešenie s metódou OCR

Výsledky ukazujú, že pôvodné riešenie pracujúce s metódou OCR vykazuje oveľa horšie výsledky ako implementované riešenie pracujúce s nástrojom pdftotext. Rozdiel je niekedy až 40%. Taktiež nové riešenie pracujúce s metódou OCR vykazuje vo väčšine prípadov lepšie výsledky. Rozdiel však nie je taký veľký ako v prvom prípade.

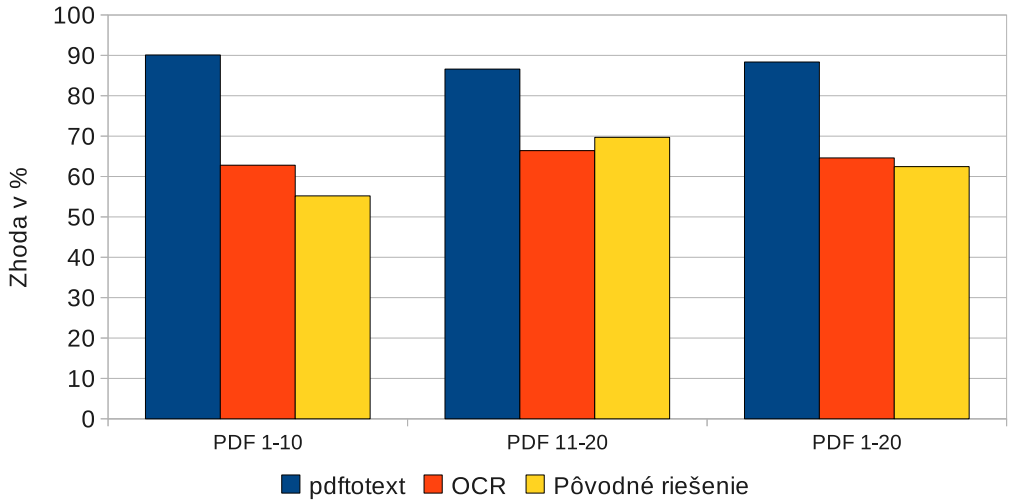
Na nasledujúcom grafe sú zobrazené zhody s referenčným riešením ďalších detiatich dokumentov, pričom v pôvodnom riešení bol text z dokumentov extrahovaný nástrojom pdftotext.



Obrázok 7.2: Percentuálna zhoda s ref. riešením. Pôvodné riešenie s nástrojom pdftotext

Nové riešenie používajúce pdftotext dosahuje aj pri druhej časti porovnávania často 90% zhodu, priemernú však 86%. Pôvodné riešenie dosahuje priemerne 69% zhodu. Nové riešenie používajúcu metódu OCR je na tom veľmi podobne, čo je vzhľadom na menej kvalitný vstupný text dobrý výsledok.

Posledný graf vyjadruje celkové výsledky implementovaného a pôvodného riešenia, vzhľadom k tomu, akým spôsobom bol vstupný text získaný. Posledná trojica hodnôt v grafe vyjadruje priemernú úspešnosť pre všetkých 20 dokumentoch.



Obrázok 7.3: Priemerná percentuálna zhoda s ref. riešením

Celkové porovnanie ukazuje, že v prípade, ak bol vstupný text spracovaný v pôvodnom riešení metódou OCR, nové riešenie pracujúce s touto metódou dosahuje takmer o 10% lepšie výsledky. Pri novom riešení s nástrojom pdftotext je zlepšenie takmer 40%. Nové

riešenie pracujúce s menej presnou OCR metódou dosahuje podobné výsledky ako pôvodné riešenie používajúce nástroj pdftotext. Nové riešenie používajúce pdftotext dosahuje o 17% lepšie výsledky.

Posledná časť grafu ukazuje, že bez ohľadu na použitú metódu spracovania textu pôvodného a nového riešenia, implementované riešenie dosahuje lepšie výsledky. Najkvalitnejší výstupný text je dosiahnutý pri extrahovaní textu nástrojom pdftotext.

Kapitola 8

Záver

V tejto práci boli analyzované chyby, ktoré vznikajú extrahovaním anglických textov z rôznych formátov. Rovnako aj pôvodné riešenie, ktoré tieto chyby opravuje a vytvára z výsledného zoznamu slov výslovnostné slovníky. Na základe získaných poznatkov bolo navrhnuté a implementované nové riešenie v podobe sady skriptov. Pri návrhu boli použité gramatiky v podobe slovníkov s pravidlami.

Výsledky oboch riešení boli porovnané s vytvoreným referenčným textom, pričom implementované kroky dosiahli lepšie výsledky, v niektorých prípadoch až o 30%. Práca má využitie napríklad, ak je potrebné zmeniť texty, ktorých pôvodný formát zmenu neumožňuje (napríklad nascanovaný dokument v obrázkovom formáte). Dokument je prevedený do jednoduchého textu, normalizovaný a používateľ ho môže ďalej využiť. Výslovnostné slovníky určené pre text získaný výstupom z upraveného vstupného textu, alebo iným spôsobom majú uplatnenie pri strojovom spracovaní reči.

Jedným zo smerov, ktorým môže práca pokračovať je rozšírenie zoznamu jazykov, na ktoré sa dajú implementované kroky aplikovať. Použité nástroje toto rozšírenie umožňujú. Tým by vznikli nové kroky špecifické pre konkrétne jazyky. Je možné vytvoriť aj prívetivejšie prostredie v ktorom používateľ určí svoje požiadavky na úpravu. Napríklad vo forme HTML stránky.

K práci bol vytvorený aj plagát vyjadrujúci postup pri práci s uvedenými príkladmi.

Literatúra

[1] *PDF Reference*. Addison-Wesley, tretí vydání, 2001, ISBN 0-201-75839-3.

[2] Andoni, A.; Onak, K.: Approximating Edit Distance in Near-Linear Time. In *SIAM Journal on Computing, 41(6), 2012 (special issue on STOC 2009)*, 2009, s. 199–204.

[3] Chakraborty, S.: Formal Languages and Automata Theory - Regular Expressions and Finite Automata. 2003: str. 4.

[4] Eikvil, L.: *OCR - Optical Character Recognition*. Norsk regnesentral, 1993, ISBN 9788253903712, 70 s.

[5] Graliński, F.; Krzysztof, J.; Agnieszka, W.; aj.: Text Normalization as a Special Case of Machine Translation. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, ročník 1, editace A. Denisjuk, 2006, ISSN 1896-7094, s. 51–56.

[6] Meduna, A.: *Automata and Languages: Theory and Applications*. Springer, London, 2000, ISBN 1-85233-074-0, 916 s.

[7] Schlippe, T.; Zhu, C.; Gebhardt, J.; aj.: Text Normalization based on Statistical Machine Translation and Internet User Support. Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany, 2010, str. 4.

[8] Smith, R.: An Overview of the Tesseract OCR Engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, Washington, DC, USA: IEEE Computer Society, 2007, ISBN 0-7695-2822-8, s. 629–633.

[9] Stüker, S.: Automatic Generation of Pronunciation Dictionaries. Technická zpráva, Carnegie Mellon University, Pittsburgh, PA, 2002.

Dodatek A

Zoznam príloh

Plagát formátu A2
CD s riešením
Ukážka normalizácie textu

Dodatek B

Ukážka normalizácie textu

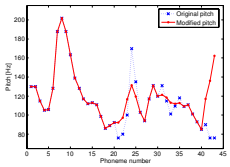


Figure 4: Adjustment and smoothing of the pitch curve at a point of concatenation (unit 23), for pause insertion (unit 31) and due to conversion from declarative to interrogative sentence (beginning at unit 40).

frames ($N2M$ algorithm [7]) and the energy discontinuities are globally minimized by means of a frame-to-frame smoothing process.

5. Experiments

CCRTV provided 900 weather reports for testing the RU-TTS system, containing the weather forecast for 175 different cities. Firstly, the system is analyzed in terms of objective performance, and secondly, after informally ratifying the highly natural synthetic speech achieved within the application, its dependence with respect to the prosodic adjustment is also analyzed.

5.1. Objective system performance

As the RU-TTS system described in this paper is a short-term real application, a performance test has been conducted on 900 synthesized reports. In average, each report lasted 39.1 ± 5.7 sec, and it was synthesized in 16.3 ± 2.75 sec. The test has been carried out over a Windows PC (PIV 3GHz - 1GB RAM) using the Visual .NET 2003 compiler. In terms of an objective speech quality measure, the number of units (diphones and triphones) per report was 43.02 ± 5.23 and the average number of concatenations (ANC) was only 0.55 ± 0.15 per sentence (each report contained 11.35 sentences in average).

In addition, if the unit selection process is conducted on the corresponding subcorpus (*welcome*, *forecast* or *farewell*), an average reduction of 40% of the execution time is achieved when compared to full corpus search, like in [4]. However, the same ANC (i.e. the same speech quality) is obtained in this case due to the totally application-oriented corpus design.

5.2. Subjective test

A preference test was developed in order to evaluate the PAM performance, a critical module for achieving highly natural speech. This test was composed of 10 pairs of audio files, each containing a sentence. One member of the pair was generated with the PAM on and the other with the PAM off. The pairs were randomly presented to 14 listeners, who were asked to choose between each pair according to their preference in terms of naturalness. The analysis of the results (see figure 5) yields a 76% preference for sentences generated with the PAM on. The results are very significant in terms of the analysis of variance (ANOVA) ($F(2, 39) = 259.13$, $p < 0.000$).

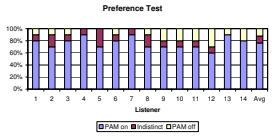


Figure 5: Preference test of 14 users judging 10 sentence pairs.

6. Conclusions

The restricted but unlimited domain TTS system described in this paper achieves highly natural speech and good performance in terms of computational requirements within the application. However, the speech quality decreases notably whenever the target sentences are not composed of the recorded set of key components. Future work is oriented towards improving this issue, conducting more exhaustive evaluation tests and implementing new strategies (e.g. clustering) to reduce the computational load of the synthesis process as full unit search is currently being conducted.

7. Acknowledgements

This work was partly supported by the Catalan Broadcasting Corporation (CCRTV). The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable suggestions and discussions during the interfaces definition process.

8. References

[1] A. W. Black and K. Lenzo, "Limited Domain Synthesis," in *ICSLP*, Beijing, China, 2000.

[2] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Säuberlich, and B. Möbius, "Restricted unlimited domain synthesis," in *EuroSpeech*, Geneve, 2003, pp. 1321–1324.

[3] R. E. Donovan, A. Ittycheriah, M. Franz, B. Ramabhadran, E. Eide, M. Viswanathan, R. Bakis, W. Hamza, M. Picheny, P. Gleason, T. Rutherfoord, P. Cox, D. Green, E. Janke, S. Revelin, C. Waast, B. Zeller, C. Guenther, and J. Kunzmann, "Current Status of the IBM Trainable Speech Synthesis System," in *The 4th ISCA Tutorial and Research Workshop on Speech Sintesis*, Perthshire, Scotland, 2001.

[4] F. Alías, I. Iriondo, and P. Barnola, "Multi-domain text classification for unit selection Text-to-Speech Synthesis," in *The 15th International Congress of Phonetic Sciences (ICPhS)*, Barcelona, 2003, pp. 2341–2344.

[5] X. Sevillano, F. Alías, and J. Socoró, "ICA-Based Hierarchical Text Classification for Multi-domain Text-to-Speech Synthesis," in *ICASSP*, vol. 5, Montreal, 2004, pp. 697–700.

[6] W3C, "Speech synthesis markup language, version 1.0." http://www.w3.org/TR/speech-synthesis.

[7] I. Iriondo, F. Alías, J. Sanchis, and J. Melenchón, "A Hybrid Method Oriented to Concatenative Text-to-Speech Synthesis," in *EuroSpeech*, Geneve, 2003, pp. 2953–2958.

Obrázok B.1: Strana PDF súboru (dodaného s pôvodným riešením)

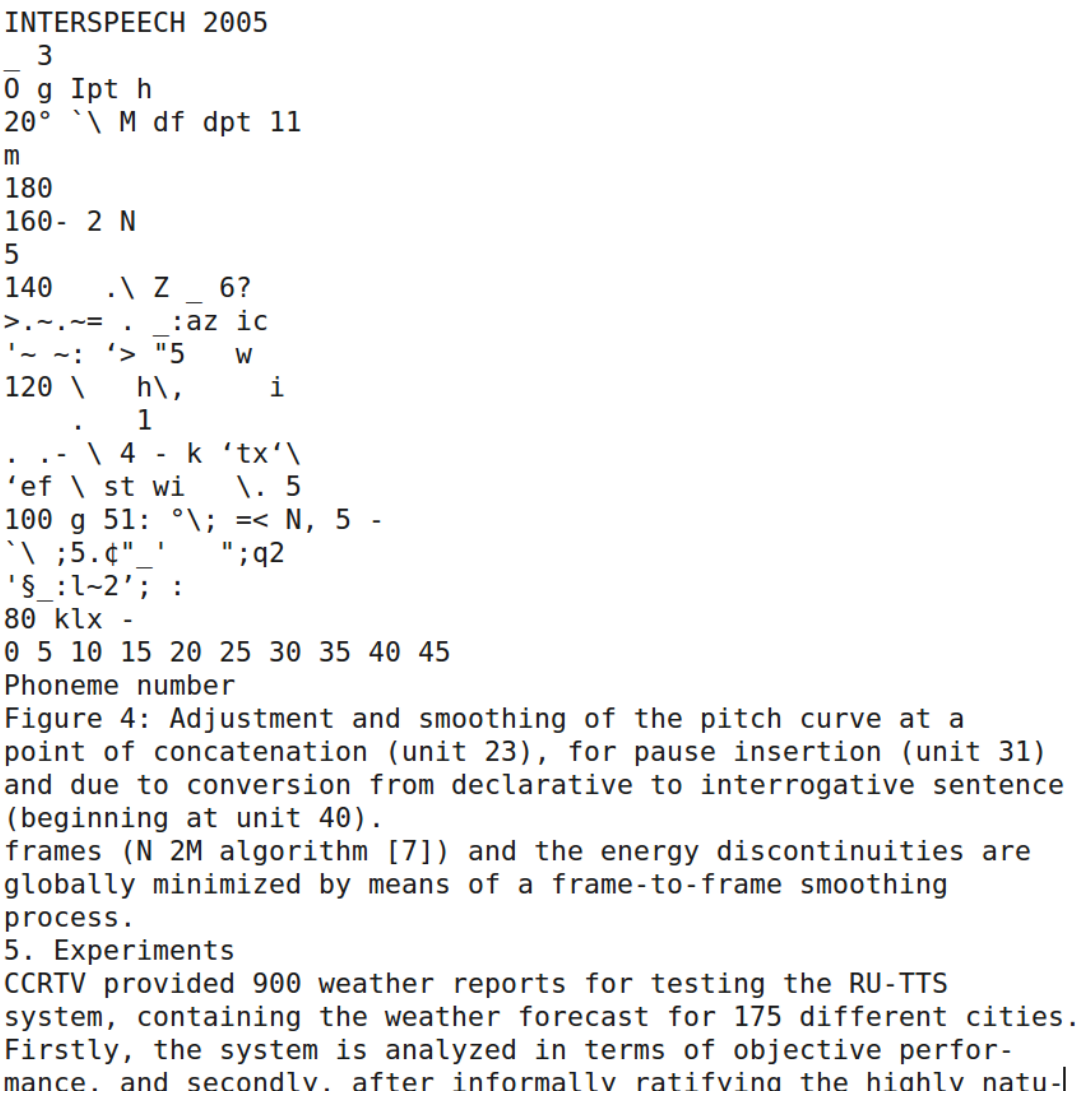


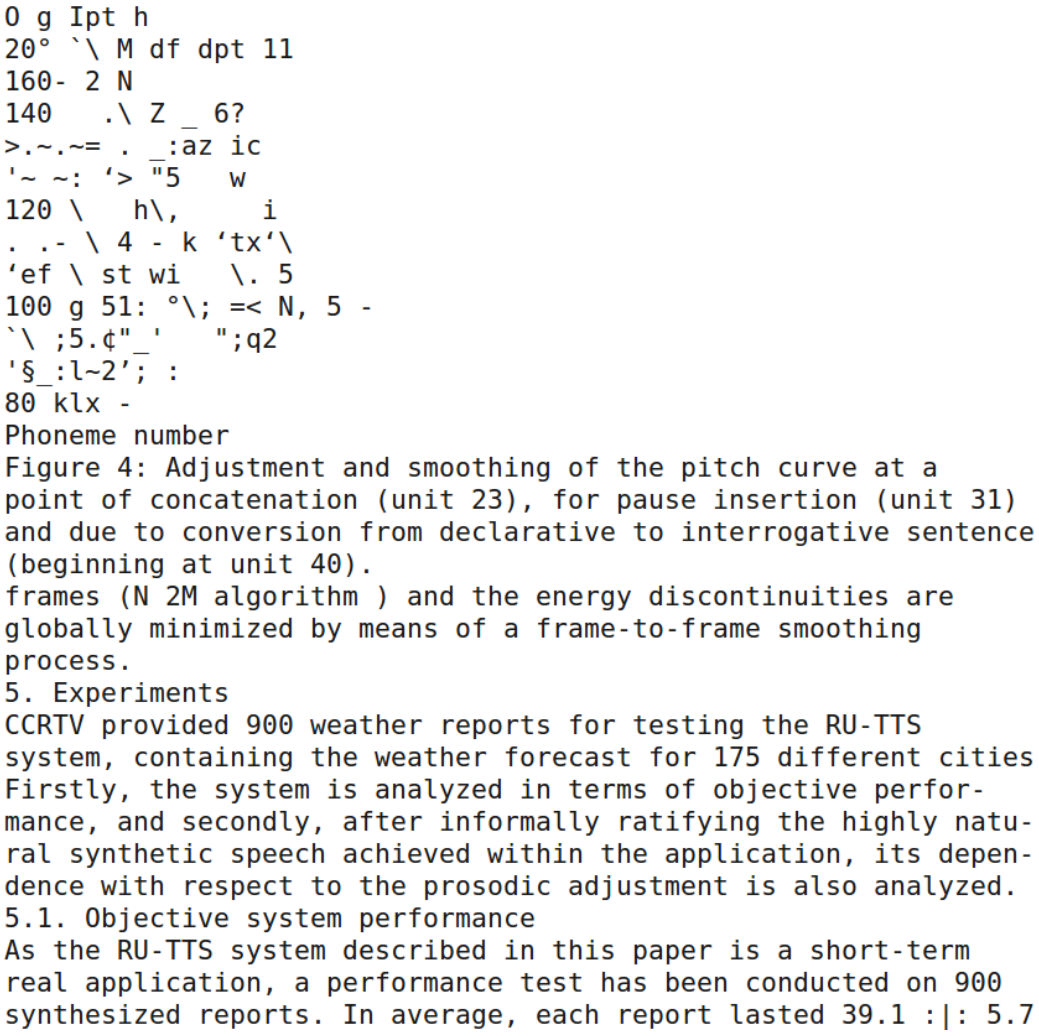
Figure 4: Adjustment and smoothing of the pitch curve at a point of concatenation (unit 23), for pause insertion (unit 31) and due to conversion from declarative to interrogative sentence (beginning at unit 40).

frames (N 2M algorithm [7]) and the energy discontinuities are globally minimized by means of a frame-to-frame smoothing process.

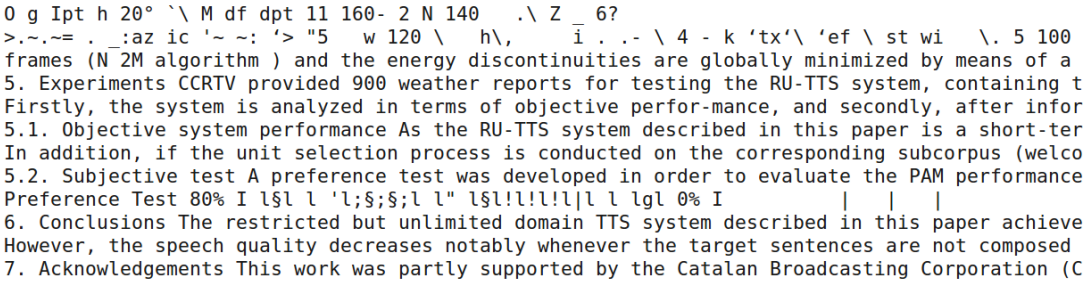
5. Experiments

CCRTV provided 900 weather reports for testing the RU-TTS system, containing the weather forecast for 175 different cities. Firstly, the system is analyzed in terms of objective performance, and secondly, after informally ratifying the highly natu-

Obrázok B.2: Časť textu extrahovaného metódou OCR



Obrázok B.3: Časť výstupu skriptov po úvodnom vymazaní nevhodných reťazcov



Obrázok B.4: Časť výstupu počas upravenia riadkovania

```
O g Ipt h 20° `\ M df dpt 11 160- 2 N 140   .\ Z _ 6?
>.~.~= . _:az ic w 120 \   h\,     i k ‘tx‘\ ‘ef \ st wi g 51: °\; =< N q2 '§_:l klx -Phoneme nu
frames N 2M algorithm  and the energy discontinuities are globally minimized by means of a frame

5. Experiments CCRTV provided 900 weather reports for testing the RU-TTS system, containing the

Firstly, the system is analyzed in terms of objective perfor-mance, and secondly, after informal

5.1. Objective system performance As the RU-TTS system described in this paper is a short-term r
 In average, each report lasted 39.1 :|: 5.7 sec, and it was synthesized in 16.3 :|: 2.75 sec.
 The test has been carried out over a Windows PC PIV 3GHz - 1GB RAM using the Visual .NET 2003 c
 In terms of an objective speech quality measure, the number of units diphones and triphones per

In addition, if the unit selection process is conducted on the corresponding subcorpus welcome,

5.2. Subjective test A preference test was developed in order to evaluate the PAM performance, a
 This test was composed of 10 pairs of audio Hles, each containing a sentence.
 One member of the pair was generated with the PAM on and the other with the PAM off.
 The pairs were randomly presented to 14 listeners, who were asked to choose between each pair a
 The analysis of the results see Hgure 5 yields a 76% preference for sentences generated with th
Preference Test 80% I l§l l 'l;§;§;l l" l§l!l!l!l|l l lgl 0% I Avg Listener ]mPAM an <1 1 t|I|PA

6. Conclusions The restricted but unlimited domain TTS system described in this paper achieves h

However, the speech quality decreases notably whenever the target sentences are not composed of
 Future work is oriented towards improving this issue, conducting more exhaustive evaluation tes

7. Acknowledgements This work was partly supported by the Catalan Broadcasting Corporation CCRTV
 The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable sugg
```

Obrázok B.5: Časť výstupu po upravení riadkovania

```
O g Ipt h twenty ° `\ M df dpt eleven one hundred and sixty two N one hundred and forty .\ Z _ six ?
>.~.~= . _:az ic w one hundred and twenty \ h\, i k ‘tx‘\ ‘ef \ st wi g fifty one : °\; =< N q two '§_:
frames N two M algorithm and the energy discontinuities are globally minimized by means of a frame-to-f

five . Experiments CCRTV provided nine hundred weather reports for testing the RU-TTS system, containin

Firstly, the system is analyzed in terms of objective perfor-mance, and secondly, after informally rati

five point one . Objective system performance As the RU-TTS system described in this paper is a short-t
 In average, each report lasted thirty nine point one :|: five point seven sec, and it was synthesized
 The test has been carried out over a Windows PC PIV three GHz minus one GB RAM using the Visual .NET t
 In terms of an objective speech quality measure, the number of units diphones and triphones per report

In addition, if the unit selection process is conducted on the corresponding subcorpus welcome, forecas

five point two . Subjective test A preference test was developed in order to evaluate the PAM performan
 This test was composed of ten pairs of audio Hles, each containing a sentence.
 One member of the pair was generated with the PAM on and the other with the PAM off.
 The pairs were randomly presented to fourteen listeners, who were asked to choose between each pair ac
 The analysis of the results see Hgure five yields a seventy six percent preference for sentences gener
Preference Test eighty percent I l§l l 'l;§;§;l l" l§l!l!l!l|l l lgl zero percent I Avg Listener ]mPAM

six . Conclusions The restricted but unlimited domain TTS system described in this paper achieves highl

However, the speech quality decreases notably whenever the target sentences are not composed of the rec
 Future work is oriented towards improving this issue, conducting more exhaustive evaluation tests and

seven . Acknowledgements This work was partly supported by the Catalan Broadcasting Corporation CCRTV.
 The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable suggestions
```

Obrázok B.6: Časť výstupu po aplikovaní normalizačných pravidiel špecifických pre jazyk

```
O g Ipt h twenty M df dpt eleven one hundred and sixty two N one hundred and forty Z six
az ic w one hundred and twenty h i k tx ef st wi g fifty one N q two 'l klx -Phoneme number Figure four Ad
frames N two M algorithm and the energy are globally minimized by means of a frame-to-frame smoothing proc

five Experiments CCRTV provided nine hundred weather reports for testing the RU-TTS system containing the

Firstly the system is analyzed in terms of objective perfor-mance and secondly after informally ratifying

five point one Objective system performance As the RU-TTS system described in this paper is a short-term r
In average each report lasted thirty nine point one five point seven sec and it was synthesized in sixteen
The test has been carried out over a Windows PC PIV three GHz minus one GB RAM using the Visual NET two th
In terms of an objective speech quality measure the number of units diphones and triphones per report was

In addition if the unit selection process is conducted on the corresponding subcorpus welcome forecast or

five point two Subjective test A preference test was developed in order to evaluate the PAM performance a
This test was composed of ten pairs of audio Hles each containing a sentence
One member of the pair was generated with the PAM on and the other with the PAM off
The pairs were randomly presented to fourteen listeners who were asked to choose between each pair accordi
The analysis of the results see Hgure five yields a seventy six percent preference for sentences generated
Preference Test eighty percent I ll l 'll l l lgl zero percent I Avg Listener mPAM an one one tIPAMoff Fig

six Conclusions The restricted but unlimited domain TTS system described in this paper achieves highly nat

However the speech quality decreases notably whenever the target sentences are not composed of the recorde
Future work is oriented towards improving this issue conducting more exhaustive evaluation tests and imple

seven This work was partly supported by the Catalan Broadcasting Corporation CCRTV
The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable suggestions and
```

Obrázok B.7: Časť výstupu po aplikovaní ďalších čistiacich krokov

```
O g Ipt h twenty M df dpt eleven one hundred and sixty two N one hundred and forty Z six
az ic w one hundred and twenty h i k tx ef st wi g fifty one N q two 'l klx -Phoneme number Figure four Ad
frames N two M algorithm and the energy are globally minimized by means of a frame-to-frame smoothing proc
five Experiments CCRTV provided nine hundred weather reports for testing the RU-TTS system containing the
Firstly the system is analyzed in terms of objective perfor-mance and secondly after informally ratifying
five point one Objective system performance As the RU-TTS system described in this paper is a short-term r
In average each report lasted thirty nine point one five point seven sec and it was synthesized in sixteen
The test has been carried out over a Windows PC PIV three GHz minus one GB RAM using the Visual NET two th
In terms of an objective speech quality measure the number of units diphones and triphones per report was
In addition if the unit selection process is conducted on the corresponding subcorpus welcome forecast or
five point two Subjective test A preference test was developed in order to evaluate the PAM performance a
This test was composed of ten pairs of audio Hles each containing a sentence
One member of the pair was generated with the PAM on and the other with the PAM off
The pairs were randomly presented to fourteen listeners who were asked to choose between each pair accordi
The analysis of the results see Hgure five yields a seventy six percent preference for sentences generated
Preference Test eighty percent I ll l 'll l l lgl zero percent I Avg Listener mPAM an one one tIPAMoff Fig
six Conclusions The restricted but unlimited domain TTS system described in this paper achieves highly nat
However the speech quality decreases notably whenever the target sentences are not composed of the recorde
Future work is oriented towards improving this issue conducting more exhaustive evaluation tests and imple
seven This work was partly supported by the Catalan Broadcasting Corporation CCRTV
The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable suggestions and
```

Obrázok B.8: Časť výstupu pred aplikovaním slovníkov s pravidlami

```
to training with twenty HMM eleven one hundred and sixty two on one hundred and forty six
phonetic new one hundred and twenty with quoi task first training fifty one on two number
frames on two HMM algorithm and the energy are globally minimized by means of a smoothing
five experiments provided nine hundred weather reports for testing the system containing
firstly the system is analyzed in terms of objective and secondly after informally ratify
five point one objective system performance as the system described in this paper is a re
in average each report lasted thirty nine point one five point seven and it was synthesiz
the test has been carried out over a Windows three minus one using the visual net two tho
in terms of an objective speech quality measure the number of units diphones and triphone
in addition if the unit selection process is conducted on the corresponding subcorpus wel
five point two subjective test a preference test was developed in order to evaluate the p
this test was composed of ten pairs of audio each containing a sentence
one member of the pair was generated with the on and the other with the off
the pairs were randomly presented to fourteen listeners who were asked to choose between
the analysis of the results see five yields a seventy six percent preference for sentence
preference test eighty percent all model model model zero percent listener an one one fig
six conclusions the restricted but unlimited domain system described in this paper achiev
however the speech quality decreases notably whenever the target sentences are not compos
future work is oriented towards improving this issue conducting more exhaustive evaluatio
seven this work was partly supported by the Catalan broadcasting corporation
the authors are also grateful to the of the university for their valuable suggestions and
```

Obrázok B.9: Časť konečného výstupu po aplikovaní všetkých pravidiel