SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

Ing. Matúš Vasek

Autoreferát dizertačnej práce

Inteligentné učenie výslovnosti rečového syntetizátora

na získanie akademickej hodnosti doktor (philosophiae doctor, PhD.)

v doktorandskom študijnom programe: 5.2.15 Telekomunikácie
v študijnom odbore: Telekomunikácie

Bratislava, máj 2017

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Matúš Vasek

Autoreferát dizertačnej práce

Inteligentné učenie výslovnosti rečového syntetizátora

na získanie akademickej hodnosti doktor (philosophiae doctor, PhD.)

v doktorandskom študijnom programe: 5.2.15 Telekomunikácie

Bratislava, máj 2017

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na Ústave telekomunikácií FEI STU v Bratislave.

Predkladateľ: **Ing. Matúš Vasek**
Ústav telekomunikácií FEI STU Bratislava
Ilkovičova 3, 812 19, Bratislava

Školiteľ: **prof. Ing. Gregor Rozinaj, PhD.**
Ústav telekomunikácií FEI STU Bratislava
Ilkovičova 3, 812 19, Bratislava

Oponenti:

Autoreferát bol rozoslaný: .....

Obhajoba dizertačnej práce sa koná: ..... o ..... hod.

v zasadacej miestnosti dekana FEI STU v Bratislave, Ilkovičova 3, 812 19 Bratislava

.....

prof. Dr. Ing. Miloš Oravec
dekan FEI STU Bratislava
Ilkovičova 3, 812 19 Bratislava

Obsah

1 Úvod

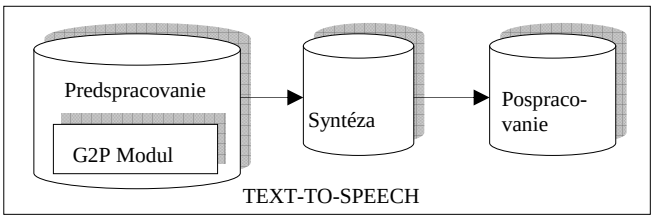
Syntéza reči má v oblasti rozvoja informačných technológii nenahraditeľné postavenie. Jej význam spočíva predovšetkým v tom, že otvára nové možnosti komunikácie človeka s informačným systémom. S tým súvisí otvorenosť pre široké spektrum ľudí, bez ohraničenia vekom, vzdelaním alebo spoločenským postavením. Do popredia sa stále viac dostáva naliehavá potreba intuitívnosti vo vzťahu k užívateľom. Flexibilita, stabilita, vzájomná kompatibilita a technologická vyspelosť produktu obohatená užitočnými aplikáciami sú východiskovými atribútmi kvalitného systémového návrhu. Syntéza reči už v minulosti dosiahla mnohé dobré výsledky. Kvalitatívne pokroky boli podporené okrem iného aj rozdelením práce do rozličných špecifických oblastí.

Fonetická transkripcia má veľký význam pre skvalitnenie syntézy reči. Fonetická transkripcia je klúčová úloha, ktorá ja zodpovedná za prepis textu do fonetickej abecedy SAMPA. V súvislosti s fonetickou transkripciou spomíname problematiku konverzie grafém do foném (Grapheme to Phoneme Conversion) a používame skratku G2P.

Systém pre fonetický prepis je podstatný prvok predspracovania textu (syntézy vyššej úrovne). Patrí sem: detekcia členenia textu, normalizácia textu, teda spracovanie čísloviek, skratiek, dátumu, času, matematických symbolov a podobne. Mnohé z týchto procesov výrazne prispievajú ku kvalite syntézy, ale predsa nie sú súčasťou základného rámca nevyhnutných prvkov syntetizátora. Kľúčovým procesom tejto úrovne syntézy je fonetický prepis textu.

2 Prehľad súčasného stavu problematiky

Pri spracovaní textu v syntéze reči sledujeme nieoľko krokov, ktoré sú potrebné, kým sa z textu stane reč. V rôznej literatúre sa kroky spracovania a ich pomenovania mierne rôznia. Základná schéma ktorá má tri základné fázy je na (Obr. 1.).



Obr. 1. Všeobecné kroky spracovania textu v syntéze reči

Fonetická transkripcia a G2P modul sú súčasťou predspracovania textu.

2.1 Fonetická transkripcia

G2P konverzia je veľmi komplexný problém. Existuje veľa možností ako sa vysporiadať s touto úlohou. Napríklad môžme použiť klasifikačné stromy, look-up tabuľkovo založené modely, slovníkovo založené prístupy, použitie modulov lingvistických pravidiel, hybridné systémy, neurónové siete, Finite State Transducers (FST), Joint Multigram Models (JMM), Conditional Random Fields (CRFs) alebo štatistické prístupy [14].

2.2 Modulárny syntetizátor

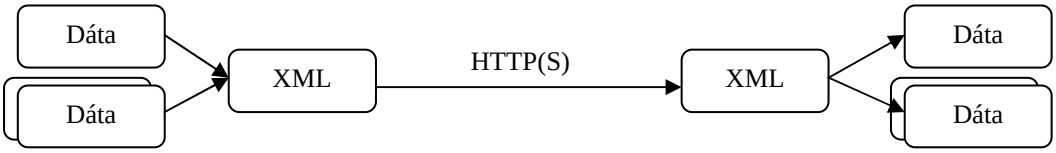
Syntetizátor reči môžeme implementovať rôznymi spôsobmi. Treba pri tom brať do úvahy, aké funkcie má v sebe zahŕňať. Môžeme predpokladať rôzne funkcie. Tieto funkcie môžu byť naprogramované rôznym spôsobom. Je potrebné, aby vedeli tvoriť jeden funkčný celok. Takto sa dostáva do popredia potreba rozdelenia syntetizátora do viacerých funkčných blokov a potreba zosúladiť činnosť jednotlivých modulov. Preto definujeme funkčné bloky, ktoré nazývame moduly, ich

funkcionalitu a rozhranie. Príkladom modulov sú modul fonetickej transkripcie, modul prepisu čísloviek, modul vytvárania prozódie atď.. Príkladom funkcií sú prepis do fonetickej abecedy, prepis číslovky textovou formou, úprava frekvencie signálu. Rozhranie (interface) definuje komunikačný protokol stanovením vstupných a výstupných parametrov a ich formátu. Extensible Markup Language (XML) je jednoduchý, veľmi flexibilný textový formát, odvodený z SGML (ISO 8879). XML má významnú úlohu pri výmene najrôznejších dát na webe aj inde [15]. XML je súbor pravidiel pre kódovanie elektronických dokumentov. Takto je to definované v XML 1.0 špecifikácii vytvorenej W3C a v niekoľkých ďalších súvisiacich špecifikáciách. Všetko sú otvorené štandardy [16]. XML dizajn sa zameriava na jednoduchosť, všeobecnosť a použiteľnosť cez internet [17]. Hlavné výhody XML sú: škálovateľnosť, transparentnosť, štandard, jednoduchosť spracovania, flexibilita a vhodný kompromis medzi nízko-úrovňovými reprezentáciami dát, vhodnými pre rýchle a jednoduché spracovanie a vysoko-úrovňovými reprezentáciami dát zameranými na priateľský a prístupný štýl pre človeka, ktorý ocenia správcovia systému (Obr. 2.).

```
<?xml version="1.0" encoding="UTF-8"?>
<text>
 <sentence text="Ahoj" delim=".">
  <word val="Ahoj">
   <transkript>LTS-RULES</transkript>
   <syl>
    <pho>a</pho>
    <pho>h</pho>
    <pho>O</pho>
    <pho>j</pho>
   </syl>
  </word>
 </sentence>
</text>
```

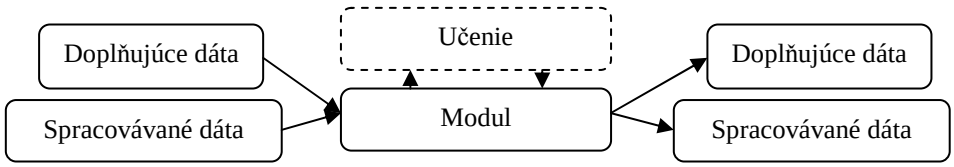
Obr. 2. XML dáta na rozhraní modulov

XML správy sa medzi modulmi prenášajú pomocou XML RPC protokolu. XML RPC je protokol pre vzdialené volanie procedúr, ktorý pracuje cez internet. XML RPC správy sú HTTP POST požiadavky. Telom správy je požiadavka v XML (Obr. 3.). Procedúra sa vykoná na serveri a hodnoty sa vrátia vo formáte XML. Parametre procedúry môžu byť skalárne veličiny, čísla, reťazce, údaje atď., a môžu nimi byť tiež zložité záznamy a štruktúrované zoznamy.



Obr. 3. XML RPC

Vstupné a výstupné dáta môžu obsahovať spracované dáta a doplnkové dáta. Napríklad modul fonetickej transkripcie poskytne prepis vo fonetickej abecede ako spracované dáta a okrem toho poskytne informáciu o použitých nástrojoch spracovania ako doplnkovú informáciu. Táto doplnková informácia môže byť potrebná napríklad pri oprave transkripcie nejakého slova. V súčasnom návrhu chýba rozhranie pre riadenie modulu pomocou tzv. učenia. Predstavuje to možnosť zasahovať do samotnej funkcie bloku činnosti syntetizátora. Spomenuté vlastnosti sú naznačené na (Obr. 4.).



Obr. 4. Modul modulárneho syntetizátora

Nedostatkom súčasnej verzie modulárneho syntetizátora je nemožnosť pridávať alebo odoberať moduly počas behu syntetizátora. Chýba bližšia špecifikácia rozhraní a definovanie toku správ pri komunikácii v inteligentnom učení. Taktiež nie je celkom jasne definované, v akých stavoch sa jednotlivé prvky systému môžu nachádzať, a čo iniciuje prechod z jedného stavu do iného.

2.3 G2P modul

G2P modul vykonáva fonetickú transkripciu. Fonetická transkripcia musí byť hotová predtým ako sa začne vykonávať konkatenatívna syntéza. My sme navrhli a používame pre fonetickú transkripciu hybridný systém. Ona pozostáva z dvoch hlavných častí (nástrojov): slovníka a pravidiel. Nasledujúca tabuľka vyobrazuje grafémy a fonémy v slovenskom jazyku. V tabuľke sú uvedené aj vzťahy medzi SAMPA abecedou a IPA abecedou (Tab. 1.).

Tab. 1. Grafémy a fonémy v slovenskom jazyku

Grafé	Fonéma		Príklad	
	IP	*SAMPA*	*Slovensk*	*Anglick*
< a >	/ a	{ a }	zajac	rabbit
< á >	/ a:	{ a: }	pás	strip
< ä >	/ ä	{ { }	mäso	meat
< e >	/ e	{ E }	pero	pen
< c >	/ e:	{ E: }	dcéra	daughte
< i >	/ i /	{ I }	pivo	beer
< í >	/ i:	{ I: }	pískať	whistle
< o >	/ o	{ O }	popol	ash
< ó >	/ o:	{ O: }	pól	pole
< u >	/ u	{ U }	puto	bond
< ú >	/ u:	{ U: }	púpava	dandeli
< ia >	/ ia	{ I_^a }	piatok	Friday
< ie >	/ ie	{ I_^E }	papier	paper
< iu >	/ iu	{ I_^U }	cudziu	foreign
< ô >	/	{ U_^O	vôla	backlas
< b >	/ b	{ b }	žaba	frog
< c >	/ c	{ ts }	cap	goat
< č >	/ č	{ tS }	čečina	branche
< dž >	/ ʒ	{	džavot	tweet
< d >	/ d	{ d }	voda	water
< t >	/ t /	{ t }	vata	widow
< ď >	/ E	{ J\ }	ďakovať	thank

Grafé	Fonéma		Príklad	
	IP	*SAMPA*	*Slovens*	*Anglick*
< ť >	/ k’	{ c }	ťahať	pull
< dz >	/ ʒ	{ [dz_>]	hádzať	throw
< v >	/ v	{ v }	slovo	word
< g >	/ g	{ g }	gagot	cackle
< h >	/ h	{ h }	hodina	hour
< ch >	/ x	{ x }	chata	cottage
< j >	/ j /	{ j }	raj	paradise
< k >	/ k	{ k }	kúkoľ	cockle
< l >	/ l /	{ l }	skala	rock
< ľ >	/ ľ	{ L }	ľavák	southpa
< Í >	/ l:	{ l=: }	vĺča	wolf
< m >	/ m	{ m }	mama	mother
< n >	/ n	{ n }	nos	nose
< ň >	/ ň	{ J }	baňa	mine
< p >	/ p	{ p }	páperie	fluff
< r >	/ r /	{ r }	para	steam
< ŕ >	/ r:	{ r=: }	vŕba	willow
< s >	/ s	{ s }	sneh	snow
< z >	/ z	{ z }	zima	winter
< š >	/ š	{ S }	šošovica	lentil
< f >	/ f /	{ f }	fajka	tobacco
< ž >	/ ž	{ Z }	žiara	glare

G2P berieme ako klasifikačný problém. Klasifikačné problémy sú všeobecne tie, kde sa pokúšame predikovať hodnotu kategoricky závislej premennej z jednej, alebo viacerých kategorických predikčných premenných [19].

G2P konvertory sú používané v systémoch automatického rozpoznávania reči (ASR) a v Text-To-Speech (TTS) systémoch na generovanie modelov výslovnosti. Existujú aj špeciálne systémy, kde sa vzťahy medzi fonémami a grafémami učia na akustickej báze. V grafémovo založenom ASR systéme v rámci Kullback-Leibler divergenčných skrytých Markovových modelov (KL-HMM) [20], [21], sú modelované vzťahy medzi akustickou formou a grafémovými podslovnými jednotkami. Viacvrstvový perceptron (MLP) je natrénovaný, aby zachytil vzťah medzi akustickou podmienkou ako napríklad kepstrom a fonémovou triedou. Potom použitím fonémovej posteriórnej pravdepodobnosti odhadovanej pomocou MLP ako pozorovanie, je určená pravdepodobnosť vzťahu medzi grafémou a fonémou, ktorý je zachytený cez KL-HMM systém [14].

G2P konverzia môže byť realizovaná i pomocou Conditional Random Fields (CRFs). CRFs sú sekvenčným modelom ktorých hlavnou výhodou je Maximum entropy Markov models (MEMMs) ale taktiež riešia „label bias problem“ principiálnym spôsobom. „Label bias problem“ znamená že prechody odchádzajúce z daného stavu konkurujú len proti sebe navzájom a nie voči všetkým ostatným prechodom v modeli [22].

2.3.1 Slovník fonetickej transkripcie

Jedným zo základov fonetickej transkripcie je slovník Ábela Kráľa, ako to už bolo vyššie spomenuté. Tento slovník bol zapojený do fonetickej transkripcie ako prvý. Časom sa ukázalo, že je potrebné tento slovník viacerými spôsobmi upravovať. Bol upravovaný jeho formát i jeho obsah. Slovník Ábela Kráľa obsahuje predovšetkým výnimky vzhľadom k základným pravidlám v slovenskom jazyku. Nakoľko vznikol už dávnejšie, obsahuje veľké množstvo slov, ktoré sa dnes už vôbec nepoužívajú, alebo len veľmi zriedkavo. Naopak tento slovník neobsahuje moderné výrazy, najmä výrazy prebrané z anglického jazyka a pojmy technickej povahy.

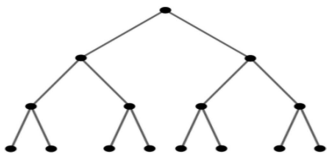
2.3.2 Pravidlá výslovnosti (LTS pravidlá)

Sú dva spôsoby ako možno LTS pravidlá vytvoriť: data-driven a knowledge-based. Jeden spôsob používa korpus a trénovanie, druhý znalosti fonetiky a jazyka. Korpusovo založený prístup bol doposiaľ považovaný za alternatívu k tradičnému knowledge založenému prístupu [27]. Korpusovo založená metóda má jednu veľkú výhodu. Pravidlá môžu byť generované automaticky. Zvlášť pozitívne možno hodnotiť, že vieme vytvárať pravidlá ušité na mieru pre aplikáciu do ktorej ich chceme nasadiť. LTS pravidlá zapisujeme v štruktúre programovacieho jazyka LISP (Obr. 5.).

```
(c
  ((n.name is h)
   ((_epsilon_))
   ((n.name is _)
    ((p.name is e) ((dz)) ((p.name is u) ((dz)) ((ts))))
    ((n.name is m) ((dz)) ((p.name is ô) ((dz)) ((ts)))))))
(i
  ((n.name is e)
   ((I_^e))
   ((n.name is a) ((I_^a)) ((n.name is u) ((I_^u)) ((i))))))
```

Obr. 5. Časť LTS pravidiel – dva stromy: písmeno “c” a “i”

Vo viacerých rečiach existuje systematický vzťah medzi písanou formou a výslovnosťou [1]. Trénovanie LTS pravidiel vyžaduje zarovnané dáta. Toto platí pre všetky jazyky, napr.: Český výslovnostný lexikón [28]. Slovenčina má niekoľko podobností s Portugalčinou, ktorá je rečou mnohých fonologických zákonitostí a teda pravidlový prístup je ekonomickejší ako slovníkový z hľadiska kapacity potrebnej na uloženie databázy. Hlavne si treba uvedomiť, že pravidlá sú schopné prepísať každé slovo a dobrá sada pravidiel sa vie vysporiadať s takmer každým transkripčným problémom [26]. Navyše pravidlá sú pre slovenčinu nevyhnutné z dôvodu flektívnych tvarov. CART je binárny rozhodovací strom, ktorý rozdeľuje každý uzol na dva detské uzly (Obr. 6.). Delenie sa robí opakovane, na začiatku stojí koreňový uzol, ktorý pokrýva celú vzorku pre trénovanie [19].



Obr. 6. Vetvenie binárneho stromu

CART predstavuje metódu vytvárania klasifikačných a regresných stromov, ktoré sa využívajú na predikciu spojitých závislých premenných, teda regresiu alebo, ako je to v našom prípade, na kategorickú predikciu premenných, teda klasifikáciu. Základy CART algoritmu položili Breiman, Friedman, Olshen a Stone v roku 1984 [32]. Takto vytvorené pravidlá predstavujú významnú techniku, ktorá kombinuje na pravidlách založené poznatky a učenie založené na štatistických metódach [31].

2.4 G2P zarovnanie

Počet znakov v ortografickom texte musí byť rovnaký ako počet znakov v SAMPA ortoepickom prepise. Zo slovníka sa vyberie len podmnožina, ktorá spĺňa základné kritérium, ktorým je

zhodnosť počtu písmen ortografickej reprezentácie slova s počtom foném ortoepického vyjadrenia v SAMPA abecede. Pri tejto validácii sa zohľadňuje skutočnosť, že v SAMPA abecede niektorým fonémam zodpovedá viacero znakov, napr.: „a:“, „J\“, „u_^“.

G2P zarovnanie je proces vykonávaný algoritmom, ktorý modifikuje SAMPA znaky v databáze. Cieľom tohto procesu je dosiahnuť stav rovnováhy, kedy sa počet písmen rovná počtu prvkov v reťazci SAMPA znakov. Pre nasadenie učiacich algoritmov je nevyhnutné G2P zarovnanie. Tento problém sa čiastočne rieši vkladaním epsilonu [9]. Epsilon je definovaný len vnútri procedúry.

2.4.1 G2P zarovnanie pomocou CRFs

CRFs je nesmerový grafický model všeobecne podmienený pozorovanou postupnosťou. Výhodou CRFs je pravdepodobnostný model, ktorý dovoluje viaceré faktory. Nevýhodou je pomalá konvergencia počas trénovania [22].

CRFs predstavuje účinný diskriminatívny model, ktorý vedie k dobým výsledkom pre prirodzené spracovanie jazyka. Tento model môže byť ľahko použitý pre jednoduchý preklad reťazca na iný reťazec, kde je zachovaná podmienka zarovnania 1 k 1 medzi vstupnou a cieľovou stranou. G2P je práve takýto problém, pretože potrebujeme pre danú sekvenciu grafém vygenerovať im zodpovedajúcu správnu výslovnosť vo forme postupnosti foném.

Lineárny reťazec CRFs je definovaný ako podmienená pravdepodobnosť (1) výstupnej postupnosti $t_1^N = t_1, \ldots, t_N$ danej vstupnou postupnosťou $s_1^N = s_1, \ldots, s_N$ použitím log-lineárnej reprezentácie:

$$p(t_1^N \mid s_1^N) = \frac{\exp H(t_1^N, s_1^N)}{\sum_{\tilde{t}_1^N} \exp H(t_1^N, s_1^N)} \tag{1}$$

$$H(t_1^N, s_1^N) = \sum_{n=1}^{N} \sum_{l=1}^{L} \lambda_l h_l(t_{n-1}, t_n, s_1^N) \tag{2}$$

$H(t_1^N, s_1^N)$ (2) definuje pozične závislú funkciu (prediktorov) $h_l(t_{n-1}, t_n, s_1^N)$.

CRFs predpokladajú rovnakú dĺžku vstupnej a výstupnej postupnosti. Preto sa tu uplatňuje tradičný prístup integrovania skrytej zarovnanej premennej (3):

$$p(\tau_1^M \mid s_1^N) = \sum_{a_1^M} p(\tau_1^M, a_1^M \mid s_1^N) \tag{3}$$

Pre každý zdrojový symbol s_n je tag t_n množinou zarovnaných výstupných symbolov τ_m a značiek „začiatok“ („begin”) (B) alebo „vnútri“ („inside”) (I). Príklad možného 1 k 1 zarovnania pre slovo/výslovnosť pár s použitím tejto schémy je v nasledujúcej tabuľke (Tab. 2.):

Tab. 2. Dvojica slovo – výslovnosť pri metóde CRFs

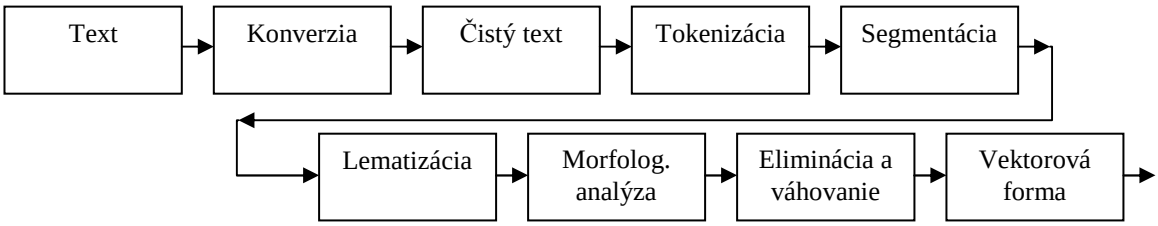
	„throw” → „ [θrɐ]”						
Word	throw	=	t	h	r	o	w
IPA	θrɐ		θ_B	θ_I	r_B	ɐ_B	ɐ_I

Táto schéma nám dovoľuje modelovať monotónne zarovnania s obmedzením na jeden výstupný symbol a viacero vstupných symbolov [34].

V slovenčine potrebujeme vyriešiť prípady v ktorých dochádza k zarovnaniu jedného vstupného symbolu na viacero výstupných symbolov, preto je potrebné hľadať novú metódu pre G2P zarovnanie, ktorá bude umožňovať ošetrenie všetkých možných prípadov.

2.5 Kontextová analýza

Extrakcia informácie sa vykonáva automatiky pomocou algoritmov, preto musia byť vstupné textové dáta vhodne predspracované. Sekvencia čiastkových krokov je znázornená na (Obr. 7.).



Obr. 7 Predspracovanie textových dát

Teoretický výskum, ktorý uskutočnili na Košickej univerzite (TUKE) ďalej podrobne vysvetluje jednotlivé fázy [10]. Principiálne je tento návrh dobrý, avšak na implementáciu môže byť pomerne komplikovaný, pretože treba vytvoriť viacero modulov pre jednotlivé kroky spracovania. Považujeme za vhodné pokúsiť sa navrhnúť taký systém, ktorý by bol svojou implementáciou aj štruktúrou jednoduhší a napriek tomu by dosahoval dobré výsledky v praxi.

2.5.1 Súčasný stav na pracovisku

Naposledy sa touto problematikou na Ústave telekomunikácii zaoberala práca Kontextová analýza textu [42]. Práca sa zaoberala procesom trénovania doménových profilov a následnou kategorizáciou textu. V tejto práci bol navrhnutý postup kategorizácie textu založenej na n-gramoch. Metóda sa opiera o frekvencie výskytu n-gramov v textoch. N-gram je postupnosť n znakov, pochádzajúcich zo slov a výrazov. Každý výraz alebo slovo sa rozdelí na n-gramy. Porovnáva vytvorené profily vstupného textu s natrénovaným profilom kategórie. Na základe rozdielu sa vypočítajú vzdialenosti, podľa ktorých sa určí výsledná téma. Implementácia tohto riešenia rozpoznáva zatiaľ medzi tromi kategóriami, a to športom, informatikou a kultúrou.

V procese kategorizácie boli najprv vytvorené profily jednotlivých kategórií, s použitím Slovenského národného korpusu. Proces využíva z neho texty, ktoré boli zatriedené do kategórií na základe manuálnej anotácie. Na týchto textoch sa metóda natrénuje tak, že sa vytvoria zo vzoriek textu n-gramy každého výrazu. Výsledkom je profil frekvencií n-gramov každej kategórie. Potom sa môže priviesť na vstup testovacia vzorka textu, ktorej sa tiež vygeneruje frekvenčný profil n-gramov. Tieto profily sa porovnajú a vypočíta sa vzdialenosť od každého profilu. Výsledná kategória sa určí podľa najmenšej vzdialenosti od profilu domény [42].

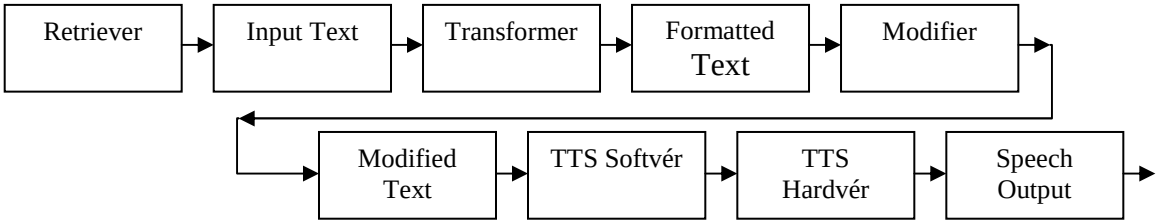
Úspešnosť tohto riešenia závisí od dĺžky vstupného textu. Čím je text dlhší, tým je výsledok presnejší. Ako bolo spomenuté, implementácia tejto metódy umožňovala rozhodovanie medzi tromi kategóriami. Pri dĺžke textu 150 znakov bola úspešnosť určenia kategórie šport 90%, informatiky 86% a kultúry 49%, kde sa vyššia presnosť prejavila pri dlhších textoch.

Táto metóda je prínosom pre kontextovú analýzu, má však niektoré nedostatky, ktoré by bolo potrebné vyriešiť. V prvom rade ide o nedostatočný počet domén klasifikácie. Nakoľko je našim cieľom použiť výsledky kontextovej analýzy pre riadenie fonetickej transkripcie a iných procesov v rámci vyššej syntézy, potrebujeme mať metódu, ktorá nám umožní rozlíšiť rádovo desiatky tematických domén. Ďalším problémom je miera úspešnosti pri detekcii kultúry. Úspešnosť 49% je potrebné zvýšiť navrhnutím metódy, ktorá bude prípadne zohľadňovať ďalšie aspekty pri kontextovej analýze.

2.6 Inteligentné učenie výslovnosti

Na základe analýzy nedávnej histórie a súčasného stavu výskumu v tejto oblasti sme dospeli k nasledovným informáciam. Problematikou inteligentnej TTS syntézy sa zaoberá patent US 6,446,040 B1 zapísaný v roku 2002. V tomto patente sa spomína, že nie je dostatočné robiť syntézu spôsobom slovo na zvuk. Starším systémom ako „Multilingual Text-to-Speech Synthesis, The Bell Labs Approach“, ktorý opísal Richard Sproat a IBM ViaVoice vyčíta nevedomosť toho čo čítajú. Ďalej navrhuje, ako neignorovať dôležité faktory pri čítaní textu, ako napríklad používateľský profil, význam textu a človeka ktorý text číta. Tento patent ponúka riešenie s použitím tzv. transformátora, modifikátora, TTS systému a rečového hardvéru. Transformátor analyzuje vstupný text a transformuje

na formátovaný text. Modufikátor modifikuje text do formy spôsobilej pre TTS. Výstup TTS sa následne prevedie na reč. Tento postup je znázornený na (Obr. 8.).



Obr. 8 Inteligentná TTS syntéza podľa US 6,446,040 B1

Nedostatkom uvedeného riešenia je absencia bližšej špecifikácie, ako má proces inteligentnej syntézy vyzerať pri čiastkových krokoch predspracovania textu. Túto medzeru by sme chceli v rámci tejto práce vyplniť. Okrem toho zásadnou skutočnosťou je, že tento návrh nepočíta so spätnou väzbou a teda s možnosťou učenia. Túto časť je potrebné samostatne navrhnúť. Spoločnosť IBM vyvinula trénovací systém syntézy reči. Tento systém je schopný natrénovať sa na novú reč v priebehu 3 až 4 dní. Tento systém obsahuje okolo 70 hlasov 10 rôznych rečí [41].

2.6.1 Prístupy súčasnej vedy k inteligentným opravám chýb v Slovenčine

Problematiku inteligentnej syntézy a opráv chýb riešili vo svojich prácach Ing. Renáta Rybárová PhD. [45] a Ing. Jozef Gonšor [46].

V práci Metódy učenia pre syntézu reči je popísaný návrh inteligentného systému pre syntézu reči a pre zmenu prozodických vlastností reči. Práca predkladá celkový návrh inteligentného systému, architektúru modulárneho rečového syntetizátora a aplikovanie učenia do jednotlivých modulov. Implementácia učenia v moduloch je na rôznej úrovni, najviac rozvinutá funkcionalita je v module fonetickej transkripcie. Moduly si vymieňajú informácie pomocou XML súborov. Architektúra je flexibilná a umožňuje zapojiť do procesu syntézy všetky alebo iba vybrané moduly. Proces učenia môže prebiehať na rôznych stupňoch. Najjednoduchším spôsobom je manuálna oprava chýb. Vyššia forma je, ak syntetizátor ponúkne možnosti opravy chyby pri označení nesprávneho slova alebo vety, v inteligentnejšom prípade sú možnosti zoradené podľa určitého pravidla (napríklad pravdepodobnosť výskytu danej možnosti v jazyku). Práca bližšie nešpecifikuje na základe akého princípu, alebo postupu by bolo možné zoradiť alternatívne fonetické prepisy. Práca ďalej uvádza že najdokonalejšou formou učenia je integrácia syntetizátora s rečovým rozpoznávačom, pretože v tomto prípade používateľ iba číta správne formy slov a viet, všetky úpravy prebiehajú automaticky. Celý proces učenia si však vyžaduje dlhší čas, počas ktorého by systém testovalo, a tým zdokonaľovalo niekoľko používateľov [37].

Práca Metodika opráv chybnej syntézy reči predstavuje aplikáciu, ktorá vytvára slovník výnimiek z nesprávne preložených slov TTS systémom vyvinutým na našej fakulte. Táto práca podobne spomína ako budúce vylepšenia inteligentné zoradenie vygenerovaných možností opravovaných slov. Pričom na začiatku zoznamu by boli slová s vysokou pravdepodobnosťou výberu používateľa. Ďalšie vylepšenia by boli v implementovaní rozpoznávania slov. Práca podrobnejšie nešpecifikuje, akou metodikou by bolo možné tieto ciele naplniť.

Učením prízvuku rečového syntetizátora sa zaoberá práca Eugenia Oancena a Adriana Badulescu [48]. Ide o aplikáciu syntézy reči pre rumunský jazyk. Dobrú kvalitu syntetizovanej reči nemožno dosiahnuť ak sa zanedbá určenie prízvuku slov v texte. Nakoľko v runumčine nie je prízvuk pevne daný, je problematické vyriešiť tento problém jednoduchým súborom pravidiel. V tejto práci je navrhnuté dynamické určenie prízvuku na základe viacerých parametrov, ako napríklad: lexikálny charakter slova, morfológia a fonetika. Nimi navrhnutá metóda sa skladá z troch častí:

- príprava textu a spracovanie slova
- klasifikácia slova
- analýza tried a formulovanie pravidiel pre prízvuk

3 Ciele dizertačnej práce

Na základe uvedenej teórie, prehľadu súčasného stavu a analýzy problémov boli vytýčené nasledovné ciele tejto dizertačnej práce. Témou práce je fonetická transkripcia textu a možnosti adaptácie výslovnosti syntetizátora počas procesu syntézy. Práca si kladie za cieľ v prvom rade komplexne analyzovať možnosti inteligentného učenia rečového syntetizátora, navrhnúť riadiaci modul pre systém inteligentného učenia, ďalej zdokonaliť algoritmus fonetickej transkripcie a následne optimalizovať vzájomnú súčinnosť slovníka fonetickej transkripcie a LTS pravidiel vzhľadom na ich rozsah. Okrem toho je cieľom práce predstaviť nové metódy váhovania alternatívnych fonetických prepisov slov v systéme inteligentného učenia s cieľom dosiahnuť na výstupe syntetizátora správnu výslovnosť reči. Tieto ciele je možné formulovať v nasledovných bodoch:

- **Navrhnúť metódu inteligentného spracovania textu pre syntézu reči.** Tento cieľ znamená hľadanie takých postupov, ktoré umožnia ponímanie textu ako celok, nie len ako postupnosť nesúvisiacich slov. S tým súvisí hľadanie metódy ako predspracovať text pre syntézu reči v závislosti od kontextu. K tomu je potrebné mať k dispozícii kontextovú analýzu textu. Teda súčasťou tohoto bodu je návrh kontextovej analýzy textu. V záujme kontextovej analýzy ďalej potrebujeme postup tvorby databázy pre kontextovú analýzu textu. Pre skvalitnenie výsledkov kontextovej analýzy je našim cieľom navrhnúť postup pre odhad gramatickej správnosti slova, mieru špecifickosti slova a mieru dôležitosti slova v tematickej doméne. Napokon potrebujeme vytvoriť spektrum tém textu, aby bolo možné zrozumiteľné interpretovanie výsledkov.
- **Navrhnúť metódu korekcie výslovnosti rečového syntetizátora na základe interakcie s používateľom.** Korekciu výslovnosti možno uskutočniť viacerými spôsobmi. Cieľom tejto práce sú dva spôsoby. Prvým je navrhnúť korekciu výslovnosti pomocou interaktívnych SAMPA znakov. Tento spôsob potrebuje poznať alternatívne výslovnosti. Aby sme ich získali dávame si za cieľ navrhnúť postup pre nájdenie výslovnostných alternatív písmen abecedy a ich pravdepodobností. S tým súvisia dva podporné ciele, ktorými sú určenie fonémovej intenzity písmen podľa korpusu a určenie distribúcie grafemickej reprezentácie podľa korpusu. Nakoľko chceme využívať korpus potrebujeme pripraviť text-to-SAMPA korpus. Druhým spôsobom je korekcia celých slov. Pre tento spôsob potrebujeme všetko to čo pre prvý a navyše ešte automatické generovanie alternatívnych výslovností slova podľa masky a ich usporiadanie podľa pravdepodobnosti.
- **Navrhnúť koncept syntetizátora s učením, komunikáciu modulov a prvky GUI pre korekciu výslovnosti v rámci existujúceho modulárneho syntetizátora na školiacom pracovisku.** Tento cieľ znamená návrh grafického rozhrania pre inteligentné učenie výslovnosti ako aj návrh sieťovej architektúry modulárneho syntetizátora s podporou učenia.
- **Analyzovať nástroje fonetickej transkripcie s ohľadom na možnosti učenia.** S týmto cieľom súvisí definovanie klasifikačného priestoru fonetickej transkripcie a hodnotenie rozsahu slovníka a pravidiel fonetického prepisu.
- **Navrhnúť metódu pre nájdenie optimálnej súčinnosti slovníka fonetickej transkripcie a LTS pravidiel.** V tejto časti sa chceme zaoberať aj frekvenčnou analýzou slov. Pre doplnenie týchto cieľov načrtnúť postup tvorby tematicky optimalizovaných LTS pravidiel. Nakoľko sa trénovanie pravidiel výslovnosti môže uskutočniť len zo G2P zarovnaných dát je potrebné navrhnúť metódu G2P zarovnania. Nakoľko predpokladáme s prítomnosťou výnimiek, je našim cieľom navrhnúť postup pre spracovanie výnimiek G2P zarovnania a pre automatické nájdenie nových G2P pravidiel zarovnania.

4 Metóda inteligentného spracovania textu pre syntézu reči

Pod inteligentným prístupom k textu rozumieme súbor metód, ktoré napodobňujú správanie inteligentnej ľudskej bytosti. Človek zvyčajne nezačína čítať text od začiatku slovo po slove. Človek sa radšej najprv pozrie na článok ako celok a všimne si, aký je článok dlhý, aký má nadpis a podnadpisy, ako je členený na odseky. Prípadne podstatnou informáciou sú obrázky a v prípade odborného textu aj matematické výrazy, tabuľky a grafy. Treba si uvedomiť, že väčšinou človek už pred začiatkom čítania samotného textu má istú predstavu o tom čo ide čítať. To je zásadný rozdiel voči syntetizátoru reči. Teda človek približne, alebo aj celkom presne vie predpokladať aký štýl a aký obsah článok má. Na základe pozorovania človeka ako inteligentnej osoby vieme navrhnúť postupy, ktoré budú pracovať podobne a to rýchlo a efektívne.

Navrhnutá metóda pracuje s kontextovou analýzou reči. Kontextová analýza je potrebná na to, aby syntetizátor vedel presnejšie, lepšie interpretovať text. Na to potrebujeme určiť do akej tematickej kategórie text patrí. V prvom kroku modul kontextovej analýzy zdetekuje tematickú doménu textu. Tento modul túto informáciu ďalej poskytne ostatným modulom, ktoré sa na základe témy textu môžu lepšie rozhodnúť pre vhodnú transkripciu.

4.1 Číslovky a skratky

Táto časť zahŕňa dva najvýznamnejšie moduly. Modul prepisu čísloviek a skratiek. Podrobnému popisu týchto modulov sa nebudeme venovať, pretože je to nad rámec tejto práce. Nás zaujíma uplatnenie princípu inteligentného učenia v týchto moduloch. Najlepšie to bude ukázané na jednoduchom príklade (Tab. 3.).

Tab. 3. Kontextovo závislá transkripcia čísloviek

1	„Hokejový zápas skončil 16:9.“
Doména = Šport	„šestnásť deväť“
2	„Pre študenta v druhom ročníku bolo ťažké vypočítať 16:9.“
Doména = Matematika	„šestnásť **deleno** deväť“
3	„Monitor má displej formátu 16:9.“
Doména = Informatika	„šestnásť **ku** deväť“
4	„Autobus odchádza zo zástavky o 16:09.“
Doména = Cestovanie	„šestnásť hodín **aj** deväť minút“

Kontextovo závislá transkripcia skratiek je navrhovaná na rovnakom princípe. Ako ilustrácia slúži príklad rôznej interpretácie skratky „FA“ v závislosti od témy textu zistenej kontextovou analýzou (Tab. 4.).

Tab. 4. Kontextovo závislá transkripcia skratiek

1	„FA“
Doména = Šport	„Futbalová asociácia“
2	„FA“
Doména = Fyzika	„frekvenčná analýza“
3	„FA STU“
Doména = Školstvo	„Fakulta architektúry STU“
4	„kozmetika značky FA“
Doména = Nakupovanie	„kozmetika značky FA“

Modul skratiek, resp. čísloviek dostane okrem textu na spracovanie aj doplnkovú informáciu od modulu kontextovej analýzy, ktorá hovorí o tom do ktorej kategórie text spadá. Na základe tejto informácie sa potom modul ľahko rozhodne pre správnu možnosť. V danom module sa musí nachádzať podpora tejto funkcionality a rôzne možnosti prepisu spolu s charakteristikou, pre ktorú tematickú doménu prislúcha ktorý prepis. V prípade že sa nenájde zhoda so špecifickou tematickou doménou, možno použiť všeobecné vyslovenie skratky, napr. „ef á“.

4.2 Metóda kontextovej analýzy

Za cieľ dolovania znalostí sme si určili zistenie kategórie vstupného textu, teda kontextovú analýzu pre určenie textovej domény. Predspracovanie textu sme navrhli podobne, ako je uvedené v schéme na (Obr. 7.), len sme vynechali proces lematizácie. Vytvorili sme vlastný systém váhovania slov, ktorý vychádza zo zistených údajov o počte výskytov slova v slovenských textoch. Z vhodného vektorového výstupu z predspracovania textu spracujúcim algoritmom vytvoríme škálu kategórií, ktoré budú číselne reprezentované a tak sa zobrazí tematické rozloženie textu vo vhodnom formáte.

Rozhodovanie v kontextovej analýze je riadené nasledovnými charakteristikami:

- Informačná hodnota slova
- Gramatická správnosť
- Dôležitosť slova
- Dôležitosť domény
- Špecifickosť slova

Globálne najčastejšie používané slová sú tie, ktoré sa používajú pri tvorbe viet a výrazov. Tie slová neobsahujú žiadnu informačnú hodnotu. No čím je slovo menej používané, tým je jeho informačná hodnota väčšia.

Dôležitosť slova sa dá vyčítať z kategorizovaných frekvenčných slovníkov. Dá sa povedať, že je to inverzná vlastnosť k vlastnosti informačnej hodnoty. Čím je slovo v doméne častejšie používané, tým je pre danú kategóriu charakteristickejšie, teda „dôležitejšie“. Každé slovo má v niektorej kategórii väčšiu dôležitosť ako v ostatných kategóriách. To znamená, že jeho „najdôležitejšia“ doména je tá, kde je jeho dôležitosť od ostatných kategórií najvyššia. Špecifickosť slova poukazuje na to, aké slovo je bežne používané a ktoré iba v určitej oblasti. Zároveň tak vyjadruje informačnú hodnotu slova, ale o niečo komplikovanejšie. Pri tejto vlastnosti porovnávame frekvencie toho istého slova z každej domény. Pri rôznych odlišných frekvenciách rovnakého slova vieme určiť, že dané slovo je špecifické a netradičné. Pri rovnakých frekvenciách môžeme povedať, že slovo je veľmi všeobecné a dokonca tým aj informačne nehodnotné. Text, ktorý chceme kontextovo analyzovať, musíme najprv rozdeliť na súbor slov, čiže tokenizovať. Slová rozdelíme podľa medzier v texte. Aby sme to mohli urobiť, text najprv očistíme od interpunkcie, cudzích znakov, číslic a dvojitých medzier, ako aj prázdnych znakov.

4.2.1 Algoritmus klasifikácie

Klasifikácia je založená na výpočte váhy, ktorej úlohou je správne ohodnotiť slová podľa dôležitosti a správnosti. Je potrebné, aby slová, ktoré nesú hlavnú myšlienku textu, boli saturované a slová, ktoré sú bezvýznamné, aby boli odfiltrované. Výstupom tejto časti je vektorová reprezentácia textu, v našom prípade ide o súbor 55 vektorov.

Celková váha slova sa skladá z:

- Globálnej váhy
- Lokálnej váhy

Interval všetkých váh je v rozmedzí <0,1>. Celkovú váhu získame súčinom lokálnej a globálnej váhy (4).

$$W_i = W_{l_i} * W_g \quad (4)$$

Globálna váha reprezentuje súčin všeobecných vlastností slova. Pre jej určenie sme vybrali spomínané vlastnosti, a to gramatickú správnosť a špecifickosť slova (5). Ich číslicové spracovanie nám zaručí správne ohodnotenie slov v texte. Tak získame z textu výrazy, ktoré sú gramaticky správne a zároveň aj informačne hodnotné.

$$W_g = r_w * g \quad (5)$$

- rw – špecifickosť slova,
- g – gramatická správnosť

4.2.2 Gramatická správnosť

Pôvodný frekvenčný slovník sme upravili pomocou krátkeho slovníka slovenského jazyka. Túto upravenú verziu sme neskôr využili tak, že sme do spoločnej tabuľky v databáze vytvorili okrem doménových stĺpcov ešte jeden stĺpec, tzv. gramatickú správnosť, ktorý obsahuje pri konkrétnom slove jednotku, keď sa slovo nachádzalo v tejto upravenej verzii globálneho frekvenčného slovníka. Slovám, ktoré sa tam nenachádzali, sme priradili nulu.

Takto by sme všetkým slovám, ktoré obsahujú názov nejakej oblasti alebo meno známej osobnosti, ale sa nenachádzajú v krátkom slovníku slovenského jazyka, priradili taktiež nulu. Pomocou početnosti výskytu slova vieme do istej miery určiť, či dané slovo je známe, resp. používané v slovenskom jazyku, alebo je nespisovné, čiže obsahuje chybu alebo preklep a nevyskytuje sa tak často. Pomocou frekvenčného slovníka sme získali počet výskytov známych mien a tiež preklepov, resp. častých chýb. Dospeli sme k nasledujúcim číslam. Pre slová, ktoré sa vyskytovali v celom korpuse v intervale <0,100>, sme priradili nulu. Ostatným, ktorých výskyt bol od hodnoty 700 a viac, sme priradili 1. Pre ostatné slová sme vytvorili matematickú prechodovú funkciu, ktorej vstupom je pomer výskytu slova k počtu slov z celého korpusu a výstupom je číslo z intervalu <0,1> (6).

$$g(x) = \sin\left(\frac{\pi\left(\left(sin^{\frac{\pi(3950*x^{0,629}-0,9)}{2}}+0,5\right)-0,5\right)}{2}+0,5\right) \quad (6)$$

4.2.3 Špecifickosť slova

Vlastnosť špecifickosti nám pomôže vyfiltrovať slová, ktoré sú bežné a informačne bezcenné. Pomocou nej dokážeme určiť, ktoré slová v texte sú dôležité a majú veľkú významovú váhu. Pri tejto vlastnosti si môžeme predstaviť koláčový graf konkrétneho slova, napríklad slova „liek“, ktorý reprezentuje rozloženie vybraného slova v jednotlivých doménach. Slovo liek je pomerne špecifické slovo, ktoré sa vyskytuje najmä v medicíne. Naopak, keď pozorujeme takmer rovnomerné rozloženie spojky „a“, čo je pochopiteľné, keďže spojka „a“ nemá informačný význam, je často využívaná pri skladbe viet a preto sa nachádza vo všetkých textoch a doménach rovnako.

Z týchto poznatkov sme usúdili, že špecifickosť slova je ukrytá v rozložení grafu. Určili sme, že parameter bude mať minimálnu hodnotu 0, keď všetky časti grafu budú rovnaké a naopak, keď rozloženie bude také, že jedna doména pokrýva celý graf, tak hodnota bude maximálna, čiže 1. Aby sme tento výsledok docielili, museli sme vytvoriť matematickú funkciu, ktorá z N vstupov (podľa počtu domén) dá na výstup číslo z intervalu <0,1>. Toto výstupné číslo poukazuje na mieru špecifickosti konkrétneho slova. Matematickú funkciu sme vytvorili tak, že saturuje rozdiely výskytov medzi doménami a tieto rozdiely sčítavame. Súčet rozdielov podelíme najväčšou možnou hodnotou tak, aby sme dostali pomer. Keďže rozdelenie slov medzi doménami nie je rovnomerné, bolo potrebné, aby sme jednotlivé výskyty znormovali podľa rozsahu jednotlivej domény (7). Celú funkciu umocňujeme na kontrolnú konštantu, aby sme mohli výsledky ladiť (8).

$$\boldsymbol{I_{d,w} = \frac{c_{d,w}}{w_d}} \quad (7)$$

- W_d – počet všetkých slov v doméne „d“, $C_{d,w}$ – počet výskytov slova „w“ v doméne „d“
- $I_{d,w}$ – dôležitosť slova „w“ v doméne „d“

$$r_w = \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N-1} \left(\frac{I_{i,w} - I_{j,w}}{\sum_{m=1}^{N} I_{m,w}} \right)^2}{\max\left(\sum_{i=1}^{N} \sum_{j=1}^{N-1} \left(\frac{I_{i,w} - I_{j,w}}{\sum_{m=1}^{N} I_{m,w}} \right)^2 \right)} \right)^k \quad (8)$$

- r_w – špecifickosť slova, N – počet domén, k – ladiaca (kontrolná) konštanta, w – slovo

4.2.4 Lokálna váha

Označenie lokálna vyjadruje, že hodnota váhy závisí od kategórie, v ktorej toto slovo hodnotíme. Čiže jedno slovo obsahuje toľko lokálnych ohodnotení, koľko je kategórií. V našom prípade, každému slovu priraďujeme 55 lokálnych hodnôt, lebo toľko máme domén. Pri tejto váhe použijeme vlastnosť dôležitosti slova a od nej sa odvíjajúcu vlastnosť dôležitosť domény. Lokálna váha kategórie, v ktorej má vybrané slovo najviac výskytov medzi doménami, dostane najvyššiu hodnotu z intervalu <0,1>, čiže 1. Ostatné lokálne váhy získajú hodnoty prislúchajúce k miere dôležitosti kategórie. Túto dôležitosť vypočítame ako podiel dôležitosti slova v kategórii ku sume všetkých (9). Lokálnu váhu vypočítame ako znormovanú hodnotu dôležitosti kategórie k maximálnej hodnote dôležitosti (10).

$$D_{d,w} = \frac{I_{d,w}}{\sum_{m=1}^{N} I_{m,w}} \quad (9)$$

- $D_{d,w}$ – Dôležitosť domény „d“ pre slovo „w“

$$W_{l_i,w} = \frac{D_{i,w}}{\max(D)} \quad (10)$$

- $W_{li,w}$ – Lokálna váha slova „w“ v i-tej doméne

Výstupom je N vektorov. Každý vektor patrí jednej kategórii. Každý prvok tohto vektora je slovo z textu, ku ktorému bola priradená lokálna váha podľa kategórie, prenásobená globálnou váhou slova.

4.2.5 Vytvorenie spektra tém

Keď už máme vhodne spracovaný text, môžeme pristúpiť ku kroku, ktorým získame žiadané informácie. Výsledkom tohto kroku bude doménová škála, ktorá bude reprezentovať zaradenie textu do jednotlivých kategórií. To znamená, že táto škála ukazuje, nakoľko text patrí do jednotlivých kategórií.

Spektrum vytvoríme tak, že prvky každého vektora, ktorý reprezentuje jednu doménu, sčítame medzi sebou. Výsledná suma je jedna hodnota spektra (11), (12).

$$S = (s_1, s_2, s_3, \dots s_N) \quad (11)$$

$$s_i = \sum_{j=1}^{N} W_{l_{i,j}} * W_{g_j} \quad (12)$$

- S – Spektrum, s_i – i-ta spektrálna zložka

Spektrum je vytvorené zo súm vektorov. Určiť výslednú kategóriu môžeme tak, že vyberieme najväčšiu sumu zo spektra a tú kategóriu, ktorej daná suma patrí, vyhlásime za výslednú. Ako si môžeme všimnúť, informácie, ktoré nám spektrum súm ponúka, siahajú ďalej. Ukazujú nám totiž doménové rozloženie textu a tým sa dá určiť téma aj ako prienik kategórií, ktorých hodnoty sú najvyššie.

Výsledky rozpoznávania textu boli veľmi dobré. Chyby detekcie nastávali v prípade, keď skúmaná téma nebola pokrytá kategóriami v databáze, alebo bola pokrývaná kategóriou, ktorej rozsah v databáze nebol dostatočne veľký. Tak vznikli všeobecnejšie a nepresnejšie určenia, ktoré neboli úplne správne. Kontextová analýza by sa dala ešte zlepšiť, keby sme mali možnosť vytvoriť databázu s dostatočne veľkými a správne zatriedenými dátami.

5 Metóda korekcie výslovnosti rečového syntetizátora na základe interakcie s používateľom

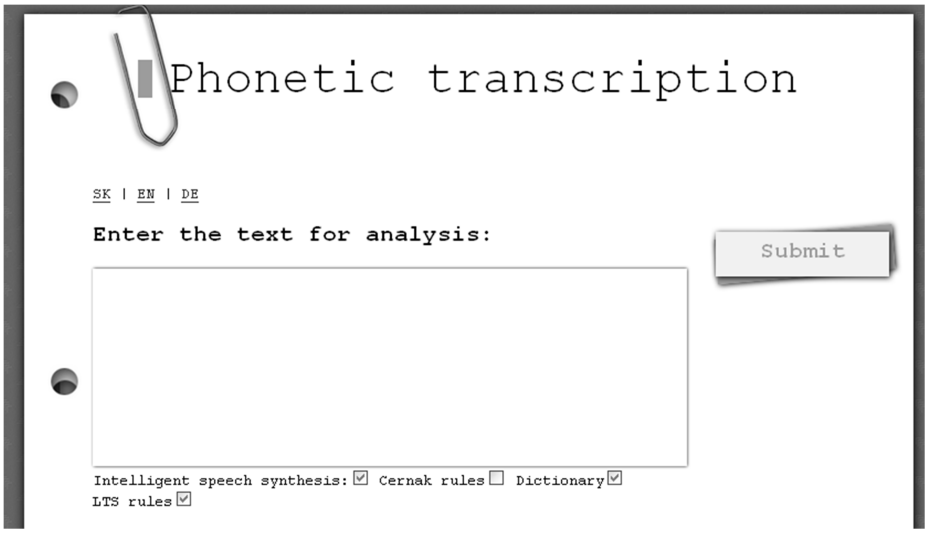
Spätnú väzbu môžme realizovať pri rôznych moduloch. Pri spätnej väzbe a učení vytvárame komunikačný kanál pre výmenu správ. GUI potrebuje byť vybavené prvkami pre učenie. Prvky učenia musia byť jednoduché a intuitívne. Užívateľ vytvorí požiadavku na učenie, GUI túto požiadavku interpretuje ako správu, ktorú pošle do HUB, ktorý ďalej smeruje požiadavku do príslušného modulu na rozhranie učenia.

Pre získanie adaptability je potrebné zohľadniť tri nasledovné prístupy:

- GUI manažment
- distribúcia správ učenia medzi modulmi
- metódy učenia pre špecifické moduly

Elementy GUI majú za úlohu motivovať užívateľa k interakcii so systémom a k použitiu metód učenia.

Na nasledovnom obrázku môžme vidieť základné časti GUI pre fonetickú transkripciu. Sú nimi: výber jazykovej varianty, pole pre zadanie textu, checkboxy na nastavenie parametrov transkripcie a tlačidlo na odoslanie (Obr. 9.).



Obr. 9 GUI pre fonetickú transkripciu

Pre univerzálne opravy výslovnosti v rámci systému inteligentného učenia je potrebné mať poznatok o alternatívnych výslovnostiach písmen z kontextovo nezávislého hľadiska. Pre získanie týchto informácii nám poslúži metóda pre nájdenie LTS alternatív.

5.1 Metóda pre nájdenie LTS alternatív a ich pravdepodobností

Navrnhutá metóda pre LTS konverziu váhovanú pravdepodobnosťami alternatívnych výslovností pozostáva z nasledujúcich krokov:

- Príprava text-to-SAMPA korpusu

- Zarovnanie G2P dát
- Vytvoriť trénovacie vektory
- Trénovanie LTS pravidiel
- Selekcia podstatných dát zo štatistík trénovania
- Interpretácia výsledkov

5.1.1 Príprava text-to-SAMPA korpusu

Text-to-SAMPA korpus môžme definovať ako množinu dát, ktoré sú definované nasledovne (13). Tu treba poznamenať, že zdrojová postupnosť s_1^M a výstupná postupnosť t_1^M nemusia mať rovnakú dĺžku. Sú prípady keď M ≠ N. Tieto prípady nazývame ako nezarované.

$$s_1^M \rightarrow t_1^N \tag{13}$$

Korpus sme dostali zlúčením niekoľkých databáz. Najdôležitejšiou z nich je slovník Ábela Kráľa. Tento slovník je jeho celoživotnou prácou, obsahuje desaťtisíce slovenských slov a ich fonetickú transkripciu v SAMPA. Tento slovník však obsahje pomerne málo cudzích slov a takmer žiadne moderné výrazy, teda výrazy, ktoré prenikli a udomácnili sa v slovenčine v posledných rokoch. My sme pridali do tohto slovníka databázu z automatického rozpoznávača reči, (ASR) a inteligentného rečového komunikačného rozhrania IRKR. Tieto projekty boli vedené tiež na Ústave telekomunkácií.

5.1.2 Štatistické dáta získané z trénovania LTS pravidiel

Trénovanie pravidiel robíme pomocou programu Wagon na základe trénovacej databázy. Vo vygenerovaných binárnych sttromoch sa nachádzajú pravdepodobnosti pre každú fonému v každom vybranom kontexte. Táto pravdepodobnosť je počítaná ako podiel počtu trénovacích vektorov podporujúcich dané pravidlo a počtu všetkých trénovacích vektorov, ktoré sa týkajú tohto výskytu.

Výsledkom trénovania sú okrem samotných stromov aj štatistiky G2P priradení. Práve tieto štatistiky boli predmetom nášho záujmu pre nasledovný výskum. Na základe týchto štatistík sa vieme dopracovať k všeobecným kontextovo nezávislým pravdepodobnostiam alternatívnych výslovností.

Ukážkou spomenutých štatistík je nasledovná tabuľka. Tieto dáta vznikli pri generovaní binárneho klasifikačného stromu pre písmeno „d“ metódou „stepwise“ pri nastavení stop value na hodnotu „1“ pri trénovaní zdatabázy vytvorenej z G2P zarovnaného korpusu (Obr. 10.).

J\	1303	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1303	[1303/1303]	100
[d_>]	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	[28/28]	100
ε	0	0	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	[53/53]	100
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[0/0]	100
[ts_>]	0	0	0	0	76	0	0	0	0	1	0	0	0	0	0	0	0	77	[76/77]	98.7
[tS_>]	0	0	0	0	0	54	0	0	0	0	0	0	0	0	0	0	0	54	[54/54]	100
[dZ_>]	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	[2/2]	100
[dz_>]	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	21	[21/21]	100
h	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	[1/1]	100
ts	0	0	0	0	1	0	0	0	0	202	0	0	0	0	0	0	0	203	[202/203]	99.51
J-	6	0	1	0	0	0	0	0	0	0	26	0	0	0	0	0	0	33	[26/33]	78.79
[t_>]	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	26	[26/26]	100
dz	0	0	0	0	0	0	0	0	0	0	0	0	283	0	0	0	0	283	[283/283]	100
t	0	0	0	0	0	0	0	0	0	0	0	0	0	737	0	0	0	737	[737/737]	100
[c_>]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	20	[20/20]	100
[J_>]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	12	[12/12]	100
#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	[0/0]	100
d	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4468	4469	[4468/4469]	99.98
	1310	28	54	0	77	54	2	21	1	203	26	26	283	737	20	12	0	4468		

Obr. 10 Štatistické dáta z trénovania LTS pravidiel

5.1.3 Interpretácia výsledkov

Výsedky sú uvedené v tabuľke (Tab. 5.). V tejto tabuľke sa pridŕžame nasledovnej konvencie. Prvý stĺpec je písmeno abecedy. Potom nasleduje zoznam foném. Pre každú fonému je uvedená pravdepodobnosť jej priradenia k danému písmenu (Obr. 11.) neberúc do úvahy kontext.

Pravdepodobnosť sme vypočítali z počtu výskytov jednotlivých priradení v zarovnanom korpuse. Fonémy, ktorých pravdepodobnosť je menšia ako 1% sú označené s „*“.

LTS($letter_i$) = $phone_1$ (**P** ($phone_1$ | $letter_i$)) $phone_2$ (**P** ($phone_2$ | $letter_i$)) $phone_3$ (**P** ($phone_3$ | $letter_i$)) … $phone_N$ (**P** ($phone_N$ | $letter_i$))
if (**P** ($phone_k$ | $letter_i$)) < 0.01 then: $phone_k$*
Example:
LTS(“c”) = “ts (0.7433) x (0.2104) ε (0.0179) dz (0.0153) k* tS* s* ε* z* [ts_>]* [k_>]* S* [tS_>]*”
LTS(“c”) = “ts” ((**P** (“ts”| “c”) = 0.7433)
“x” ((**P** (“x”| “c”) = 0.2104)
“ε” ((**P** (“ε”| “c”) = 0.0179)
“dz” ((**P** (“dz”| “c”) = 0.0153)
“k” ((**P** (“k”| “c”) = 0.0066 < 0.01)
“tS” ((**P** (“tS”| “c”) = 0.0022 < 0.01)
…

Obr. 11 Vektor pravdepodobností

V niektorých pípadoch sa písmeno nečíta. V takomto prípade mu priradíme špeciálny znak "ε" (epsilon). Tento znak vyjadruje neexistujúcu fonému, resp. zvuk s nulovým trvaním v reči. Epsilon je definovaný vo fáze pre LTS konverziu. Musíme brať do úvahy, že niektoré grafémy sú formované viacerými písmenami. Na predchádzajúcom obrázku sme uviedli príklad pre písmeno „c“. Prepis písmena pomocou LTS pravidiel môžme symbolicky zapísať ako „LTS(letter)“. Výsledkom takejto operácie je množina všetkých foném, ktorých pravdepodobnosť je vyššia ako nula, je nenulová, a ich pravdepodobností. Výsledky jasne potvrdzujú prítomnosť očakávaných fonetických zákonitostí výslovnosti, avšak navyše môžme vidieť aj to nakoľko je ktoré pravidlo rozšírené, alebo naopak zriedkavé. Takáto informácia má pre nás hodnotu, ktorú nám knowledge-based prístup nevie priamo ponúknuť. V nasledujúcom si tento príklad detailnejšie rozoberieme.

Všimnime si teraz písmeno „d“. Toto písmeno nám veľmi dobre poslúži za príklad, pretože sa dá na ňom ukázať mnoho rozličných jazykových javov (Obr. 12.). Ako prvé môžme vidieť dominantú transkripciu na fonému {d}. Hneď vzápätí sa stretáme s fonémou {J\}, ktorá znamená zmäkčovanie písmena „d“, ktoré sa uplatňuje predovšetým v tvaroch „de, te, ne, le“. Potom nasleduje spodobovanie, ktoré vyjadruje fonéma {t}. Okrem toho sa dá pozorovať väzba na dvojhlásku {dz}. V niektorých prípadoch použije transkripcia {ε}, pretože zvuk daného písmena „d“ sa realizuje prepisom kontextu a ďalší zvuk by bol nadbytočný. Všetky tieto doposiaľ spomenuté prípady môžme považovať za dobre známe jazykové zákonitosti, ktoré by azda človeku hneď napadli. O niečo zložitejšie sú však spoluhláskové zhluky a spodobovanie ako { ts, [d_>], [J_>], [t_>], [dz_>], [dZ_>], [ts_>], [tS_>]}. Napokon ostáva ešte {J-} ktoré je zvukovo podobné {J\}. Ak by sme pridali do korpusu anglické slovo "made", ktroré sa i v slovenčine dosť často môže vyskytovať, pribudla by fonéma {I_ ^}. Teraz uvedieme konkrétne príklady slov v ktorých prichádza k rôznemu vyslovovaniu toho istého písmena (Obr. 12.). Môžme pozorovať 15 odlišných výslovností len medzi čisto slovenskými slovami!

{ d }	**d**al	*gave*
{ J\ }	**d**ebna	*box*
{ t }	o**d**kaz	*link*
{ ε }	od**d**eľ	*detach*
{ ts }	o**d**skok	*rebound*
{ dz }	cu**dz**ina	*outland*
{ [d_>] }	o**d**dnes	*from today*
{ [J_>] }	o**d**deľ	*detach*
{ [t_>] }	o**d**tok	*outflow*
{ [ts_>] }	o**d**sať	*suck*
{ [c_>] }	o**d**ťať	*sever*
{ [dz_>] }	o**dz**adu	*backwards*
{ [dZ_>] }	o**dž**ať	*harvest*
{ [tS_>] }	na**d**šene	*enthusiastically*
{ J- }	**d**edom	*grandpa*
{ I_^ }	ma**d**e	*made*

Obr. 12 Príklady rôznej výslovnosti písmena “d”

Časť výsledkov získaných touto metódou je uvedená v nasledujúcej tabuľke (Tab. 5.).

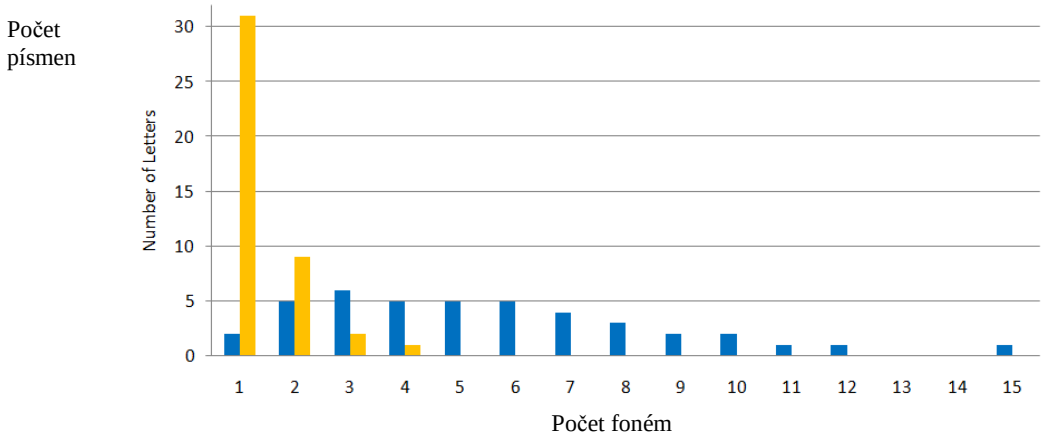
Tab. 5. LTS vzťahy

Písmeno	Štandardný vzťah		Vzťah s pravdepodobnosťou podľa korpusovej metódy	
	Fonéma	*Počet foném*	*Fonéma (p(fonéma|písmeno))*	*Počet foném*
< Í >	{ l=: }	1	{ l=: (1.0000) }	1
< ô >	{ U_^O }	1	{ U_^O (0.9980), O* }	2
< é >	{ E: }	1	{ E: (0.9864), E*, ɛ* }	3
< m >	{ m }	1	{ m (0.9846), mF*, F*, ɛ*, [m_>]* }	4
< k >	{ k, g }	2	{ k (0.9922), g*, ɛ*, [k_>]*, [g_>]* }	5
< ž >	{ Z }	1	{ Z (0.8484), S (0.1369), [S_>]*, ɛ*, z*, s* }	6
< i >	{ i, i: }	2	{ I (0.7864), I_^E (0.1372), I_^a (0.0624), I_^U*, I_^*, I:*, ɛ* }	7
< n >	{ n }	1	{ n (0.6615), J (0.2369), N (0.0607), ɛ (0.0179), [n_>], (0.0111) nz*, [J_>]*, [N\]* }	8
< ď >	{ J\, c }	2	{ J\ (0.6999), [c_>] (0.2559), c (0.0194), J- (0.0138), ɛ*, [dZ_>]*, ts*, [t_>]*, t* }	9
< t >	{ t }	1	{ t (0.7364), c (0.2242), ɛ (0.0191), d*, [ts_>]*, [t_>]*, j*, J*, [c_>]*, [d_>]* }	10
< ť >	{ c, J\ }	2	{ J\ (0.7807), c (0.1562), [c_>] (0.0215), ts (0.0139), J-*, [J_>]*, ɛ*, t*, [ts_>]*, [t_>]*, [d_>]* }	11
< t >	{ ts, dz, k }	3	{ ts (0.7433), x (0.2104), ɛ (0.0179), dz (0.0153), k*, tS*, s*, z*, [ts_>]*, [k_>]*, S*, [tS_>]* }	12
< d >	{ d, J\, t, ts }	4	{ d (0.6173) J\ (0.1657) t (0.0895) ts (0.0435) dz (0.0288) [ts_>] (0.0123) [tS_>]* ɛ* [t_>]* [d_>]* J-* [c_>]* [dz_>]* [J_>]* [dZ_>]* }	15

V tejto tabuľke (Tab. 5.) môžme vidieť aj také fonémy, ktoré nie sú uvedené v tabuľke základných foném (Tab. 1.). Platí, že každá graféma má svoju základnú fonému. Je to fonéma, ktorá reprezentuje zvuk, ktorým sa daná graféma vysloví ak stojí samostatne, alebo aj v mnohých kontextoch v slovách. Avšak nastávajú prípady mutácie na iné fonémy v závislosti od kontextu.

5.2 Pravdepodobnosti alternatívnych LTS prepisov

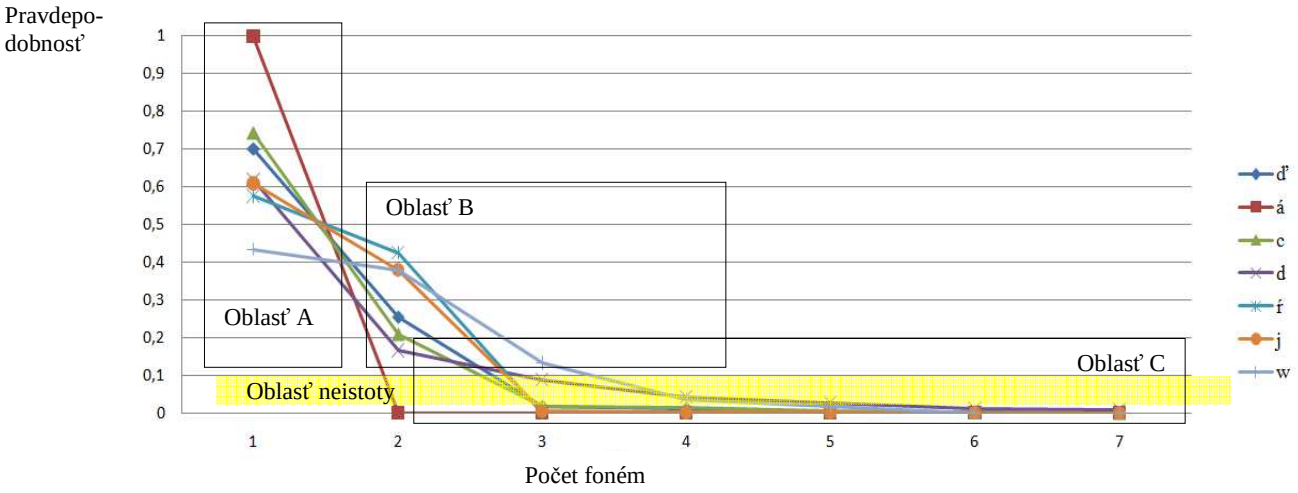
Distribúcia fonetickej reprezentácie získanej navrhnutou korpusovou metódou je porovnaná s prístupom založeným na lingvistických pravidlách (Obr. 13.). Vybrali sme niekoľko písmen aby sme ich zobrazili v grafe. Vezmime si konkrétne písmená: “ď“, „á“, „c“, „d“, „ŕ“, „j“ and „w“ (Obr. 14.).



Obr. 13 Distribúcia fonetickej reprezentácie (žltá: prístup založený na lingvistických pravidlách; modrá: navrhnutá korpusovo založená metóda).

Ostatné písmená majú charakter podobný niektorému z týchto vybraných písmen. V grafe môžme rozlíšiť tri oblasti, ktoré sú charakteristické pre všetky písmená. Prvou oblasťou je oblasť A, ktoré osahuje základnú fonému každého písmena. Je to fonéma, ktorá najčastejšie zodpovedá danej

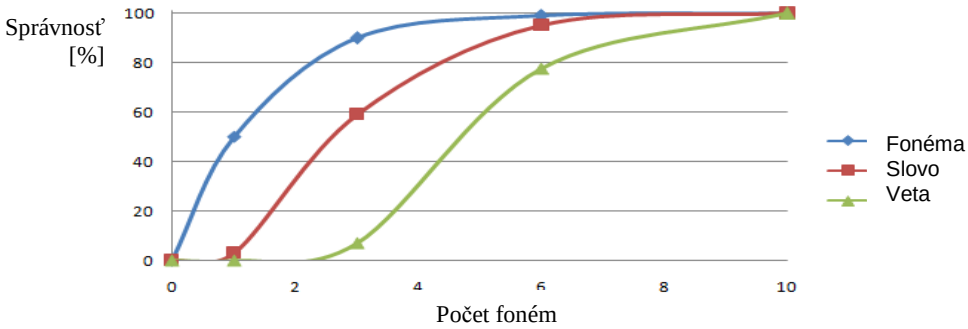
graféme. Pravdepodobnosti týchto foném sú obvykle v rozsahu 0,4 až 1. Fonémy, ktoré vznikajú uplatnením riadnych jazykových javov sú súčasťou oblasti B. Pravdepodobnosti týchot foném sa pohybujú zvyčajne v rozmedzí 0,05 až 0,45. Oblasť C obsahuje fonémy, ktoré majú svoj pôvod vo výnimkách. Príkladom výnimiek môžu byť dialektizmy a cudzie slová. Pre túto skupinu je charakteristická hodnota pravdepodobnosti nižšia ako 0,01. Veľkosť týchto oblastí je rozdielna pri jednotlivých písmenách v abecede. V prípade niektorých písmen obsahuje oblasť B nula foném. Sú to písmená, ktoré nevstupujú do hry pri obvyklých javoch. Môžme povedať, že ich výslovnosť v slovenčine je ustálená a nemenná. V takomto prípade za oblasťou A nasleduje hneď obasť C. Príkladom je písmeno „á“.



Obr. 14 Pravdepodobnosti vybraných LTS vzťahov

5.2.1 Vplyv počtu foném na správnosť transkripcie

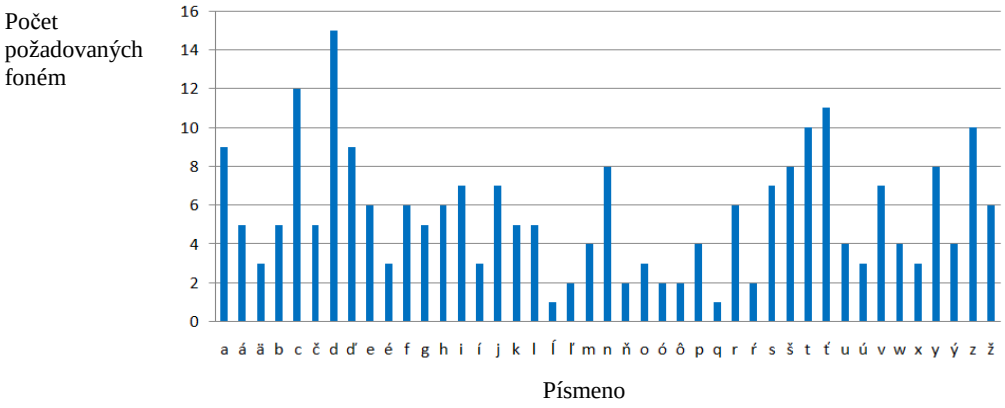
Zaujímalo nás, minimálne koľko alternatívnych foném je potrebné v transkripcii pripustiť, aby sme umožnili dosiahnuť istú správnosť prepisu. Vzali sme do úvahy chybovosť pri prepise fonémy, slova a vety. Chybovosť slova a vety sme vypočítali na základe priemernej dĺžky slova a vety v slovenčine. Z grafu (Obr. 15.) vidíme, že už základná fonéma s dvomi alternatívnymi dosahuje pomerne vysokú fonémovú presnosť. Presvedčili sme sa však, že pre celý text táto korektnosť nie je dostačujúca.



Obr. 15 Maximálna správnosť transkripcie, ktorú možno dosiahnuť pri použití istého počtu alternatívnych foném

Pri prieskume súčasného stavu výskumu v tejto oblasti sme sa stretli s tabuľkami alternatívnych foném, ktoré ponúkali maximálne 4 alternatívy. Takýmto postupom by sme dosiahli fonetický prepis, ktorý by mal prinajlepšiom v priemere chybu v každej piatej vete. A to je len chyba daná nastavením samotného systému, ešte okrem toho treba rátať s chybou kontextovo závislej klasifikácie. Takáto úspešnosť sa nám javí ako nedostatočná. Navrhujeme zvýšiť správnosť prepisu

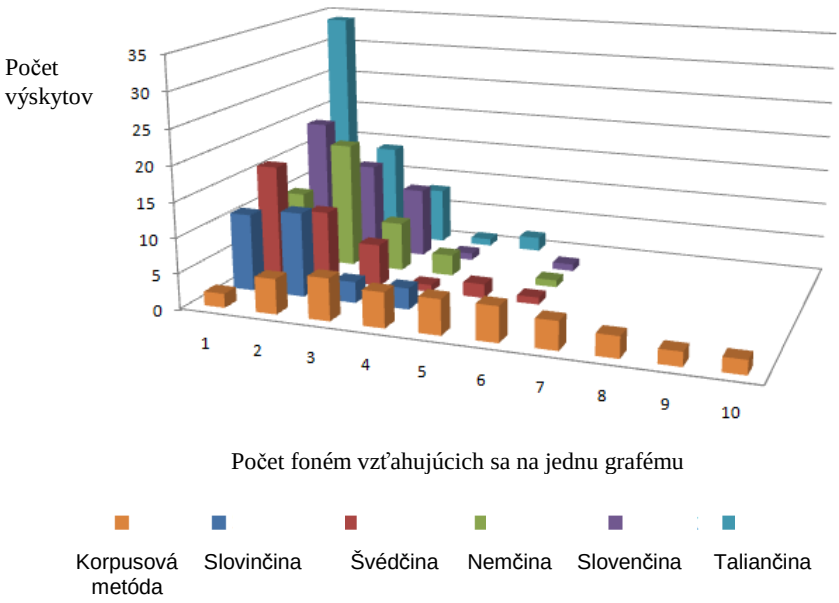
vety na vyše 95% tým že použijeme 9 a viac alternatívnych foném. Nasledovný graf (Obr. 16.) znázorňuje fonémovú intenzitu písmen.



Obr. 16 Fonémová intenzita písmen

5.2.2 Porovnanie s európskymi jazykmi

V tejto časti sa budeme venovať porovnaniu výsledkov s európskymi jazykmi. Výskum v tejto oblasti bol doposiaľ založený na lingvistických pravidlách. Metóda založená na korpuse sa mierne líši. Líši sa najmä v tom, že zahŕňa všetky javy, ktoré sa v korpuse nájdu. Jednotlivé javy sú prioritizované jedine počtom trénovacích vektorov. Korpusová metóda odhalí všetky jazykové pravidlá a navyše zachytí všetko čo je nad rámec týchto pravidiel. G2P vzťahy nad rámec základných pravidiel majú svoj pôvod predovšetým v penetrácii jazyka cudzími slovami, dialektovými formami a zloženými slovami. Výsledky získané touto metódou znázorňujú že najvyšší počet jednoznačných priradení má taliančina [33]. V nasledujúcom grafe (Obr. 17.) je porovnanie výsledkov dosiahnutých našou metódou.



Obr. 17 Distribúcia grafemickej reprezentácie - porovnanie s navrhnutou korpusovou metódou

Môžme si všimnúť značnú odlišnosť našich výsledkov od výsledkov tradičných metód. Je to zapríčinené širším pohľadom na problém a robustnou metódou. Relevantnosť týchto výsledkov potvrdzuje aj príklad písmena „d“ (Obr. 12.). V budúcnosti je možné ďalej rozvíjať tento výskum zameraním sa na porovnanie ortografickej neistoty písmen, distribúciu grafemickej reprezentácie, grafémovú veľkosť a užitočnosť písmen [19]. Slovenčinu môžme porovnávať hlavne s európskymi

jazykmi ako je angličtina, nemčina, taliančina, slovínčina, švédčina [33] a ukrajinčina [38]. Rovano možno vziať do úvahy ortoepickú neistotu grafém a aplikovať výsledky do praxe.

5.3 Interaktívna SAMPA

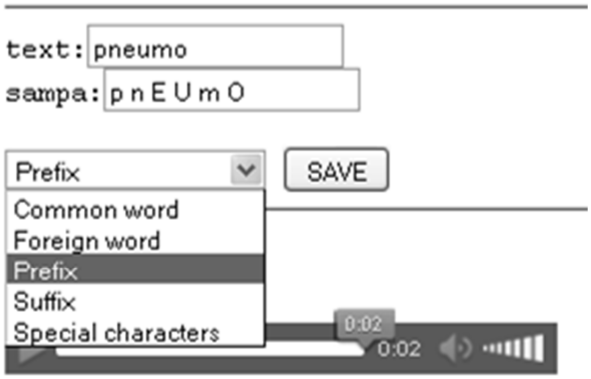
GUI zobrazuje SAMPA znaky v interaktívnom móde. Užívateľ môže kliknúť na každý SAMPA znak jednotlivo. Pre užívateľov, ktorí nemajú znalosť SAMPA abecedy je vhodné nahradiť SAMPA znaky bežnými písmenami tak, aby im užívateľ rozumel a vedel aký zvuk znamenajú. V každom prípade si užívateľ po vykonaní zmien môže prehrať reč a vypočuť, aké zmeny spravil. V ďalšom budeme popisovať systém používajúci SAMPA znaky. Princíp je však rovnaký aj pre intuitívne vyjadrenie výslovnosti. Po kliknutí na nejaký SAMPA znak je tento znak vymenený za iný, ktorý má po ňom najväčšiu pravdepodobnosť. Napríklad ak by syntetizátor chybne prečítal sovo „viedenský“ ako „v I_^E J\ E n s k I:“. Táto transkripcia je nesprávna, chyba je v SAMPA znaku „n“. Mal tu byť SAMPA znak „nz“ namiesto „n“. Písmeno „n“ má 8 možných výslovností. Prvá výslovnosť je štandardná a ostatných 7 je alternatívnych. (Tab. 7.).

Tab. 7. Alternatívne výslovnosti písmena „n“.

	Slovo = „viede**n**ský“			
	Výslovnosť „n“	Stav	Akcia	SAMPA výstup
0	n	NOK	hľadaj ďalej	v I_^E J\ E **n** s k I:
1	J	NOK	hľadaj ďalej	v I_^E J\ E **J** s k I:
2	N	NOK	hľadaj ďalej	v I_^E J\ E **N** s k I:
3	ɛ	NOK	hľadaj ďalej	v I_^E J\ E s k I:
4	[n_>]	NOK	hľadaj ďalej	v I_^E J\ E **[n_>]** s k I:
5	nz	OK	ulož výslovnosť	v I_^E J\ E **nz** skI:
6	[J_>]	-	-	v I_^E J\ E **[J_>]** s k I:
7	m	-	-	v I_^E J\ E **m** skI:

Užívateľ potrebuje kliknúť päť krát aby sa dopracoval k požadovanej výslovnosti. Zvyčajne stačí jeden alebo dva krát. Tento príklad sme vybrali len na ilustráciu. V zásade platí, že čím je SAMPA znak menej frekventovaný, tým neskôr ho systém na generovanie alternatívnych výslovností ponúkne. Po poslednom SAMPA znaku algoritmus ponúkne opäť prvý.

Inteligentné funkcie zahŕňajú aj schopnosť naučenia predpony, prípony a cudzích slov. Cudzie slová zahŕňajú ľubovolný reťazec písmen, ktorý je svojou výslovnosťou netypický pre slovenčinu (Obr. 18.). Na príklade vidíme slovo „pneumo“. Toto slovo sa vyskytuje dosť často v medicínskych textoch ale aj v praxi pri komunikácii.



Obr. 18 Inteligentné funkcie

Ak sa syntetizátor naučí prepisovať predponu, príponu alebo cudzie slovo, prejaví sa to v ďalšej transkripcii ihneď.

5.3.1 Generovanie alternatívnych fonetických prepisov slova

Generovanie alternatívnych fonetických prepisov je založené na niekoľkých krokoch. Ako prvé je potrebné detegovať miesta v ortografickej forme slova, ktoré môžu byť viacznačné pri fonetickom prepise. V týchto miestach najčastejšie nastáva chyba. Konkrétne sa jedná o miesta spodobovania a zmäkčovania. V ďalšom kroku sa kombinatoricky vygenerujú všetky možné kombinácie na základe toho, koľko viacznačných miest bolo v slove zistených a koľko značné sú tie viacznačné slová. Uveďme príklad (Tab. 8.).

V slove „internet“ algoritmus zdeteguje 3 viacznačné miesta, sú to: „te“, „ne“ a „t“. Každé z týchto miest je dvojznačné a to preto, že „te“ a „ne“ sa môže čítať buď tvrdo, alebo mäkko. Písmeno „t“ sa môže, ale nemusí spodobiť na „d“. Teda podľa toho sa určí, že bude spolu 2^3, teda 8 kombinácií. Informácia o tom či sa v danom mieste výslovnosť zmení na druhý možný variant v tej-ktorej možnosti je zachytená binárnym spôsobom v maske. Pre tento prípad bude maska vyzerať nasledovne: „000, 001, 010, 100, 011, 110, 101, 111“.

Podľa tejto masky sa vygenerujú prepisy nasledovne: „internet, interned, interňet, inťernet, interňed, inťerňet, inťerned, inťerňed“. Označme mäkkú výslovnosť veľkým písmenom a tvrdú malým a dostaneme: „internet, interned, interNet, inTernet, interNed, inTerNet, inTerned, inTerNed“.

Po prepise do SAMPA dostávame: „I n t E r n E t, I n t E r n E d, I n t E r d/ E t, I n c E r n E t, I n t E r d/ E d, I n c E r d/ E t, I n c E r n E d, I n c E r d/ E d“. Teraz máme nové možnosti čítania, predpokladáme že jedna z nich je správna, ponúkneme ich užívateľovi, aby si vybral.

Tab. 8. Generovanie alternatívnych fonetických prepisov slova.

	i n *t e* r *n e t* („te“, „ne“, „t“)		
	Binárna maska	Výslovnosť	SAMPA
1	000	in-**te**-r-**ne-t**	I n t E r n E t
2	001	in-**te**-r-**ne-d**	I n t E r n E d
3	010	in-**te**-r-**ňe-t**	I n t E r J E t
4	100	in-**ťe**-r-**ne-t**	I n c E r n E t
5	011	in-**te**-r-**ňe-d**	I n t E r J E d
6	110	in-**ťe**-r-**ňe-t**	I n c E r J E t
7	101	in-**ťe**-r-**ne-d**	I n c E r n E d
8	111	in-**ťe**-r-**ňe-d**	I n c E r J E d

Algoritmus vie automaticky určiť, ktorá z možností je najpravdepodobnejšia. Použijeme štatistiku výskytu jednotlivých SAMPA znakov na zoradenie možností podľa pravdepodobnosti.

Ak by sa syntetizátor riadil len základnými pravidlami výslovnosti v slovenčine slovu „intenet“ by zodpovedala výslovnosť: „in-ťe-r-ňe-t“. Takáto výslovnosť je nesprávna. Spomenutým spôsobom sa rýchlo a jednoducho dopracujeme k oprave. Správna odpoveď by bola medzi prvými.

5.3.2 Dosiahnuté výsledky

Je jasné, že tieto metódy otvárajú možnosti pre zlepšenie kvality fonetického prepisu a opravu chýb. Ich efektivita závisí aj od ich použitia. Napriek tomu sme spravili jenoduchý test ktorý pomôže kvantifikovať dosiahnuté výsledky v praxi. GUI sme vybavili priamo možnosťou pre prepis pomocou starých pravidiel bez podpory inteligentných metód, aby sme vedeli pozorovať rozdiel.

Texty sme vybrali z piatich rôznych tematických oblastí. Každý text bol prepísaný pomocou starých LTS pravidiel, ktoré vytvoril Miloš Cerňák [1]. Potom sme prepisovali ten istý text s použitím týchto metód. (Tab. 9.) udáva o koľko percent sa znížila chybovosť transkripcie. Chybovosť sme počítali ako podiel počtu chybných slov k počtu všetkých slov v texte. Texty boli čerpané z internetu.

Tab. 9. Zníženie chybovosti fonetického prepisu textu.

	Tematická oblasť	Rozsah textu [slová]	Zlepšenie [%]
1	Medicína	160	8.125
2	Šport	154	4.545
3	Kultúra	50	14.285
4	Ekonomika	301	1.661
5	Financie	265	2.641

Výsledok je výrazne ovplyvnený prítomnosťou cudzích slov. Čím viac cudzích slov sa v texte nachádza, tým výraznejšie je zlepšenie dosiahnuté použitím týchto metód.

V budúcnosti by bolo vhodné rozvinúť systém opravy slov s použitím nových modalít. Týka sa to najmä obojstrannej komunikácie medzi užívateľom a zariadením.

Výhodou navrhnutých metód je aj to, že uľahčujú prácu rozpoznávaču reči. Prioritizáciou, generovaním a zoradením alternatívnych výslovností sa zmenšuje množina rôznych možností medzi ktorými sa má rozpoznávač rozhodnúť.

6 Koncept syntetizátora s učením

Potreba inteligentného učenia pri syntéze reči je jednoznačná, pretože požadovaný pokrok v kvalite syntézy nie je možné dosiahnuť len úsilím programátora a jazykovedca. Je veľmi komplikované ba priam nereálne vopred predpokladať a ošetriť všetky problémové udalosti pri syntéze reči. Dobrý príklad je fonetická transkripcia. Ak by sme aj spravili pravidlá a slovník, ktoré budú na 100% správne pracovať podľa zákonitostí slovenského jazyka, vyskytne sa ľahko cudzie slovo, ktoré porušuje tieto zákonitosti. Náš systém nemá byť striktne určený len pre spisovné slovenské slová, ale má byť čo najviac otvorený pre univerzálne použitie. Užívateľ má vnímať čo najväčšiu voľnosť a flexibilitu syntézy podľa jeho potrieb. Niektoré nárečia uprednostňujú tvrdé vyslovovanie, iné naopak zmäkčujú veľmi výrazne. Syntetizátor chce ponúknuť minimálne na úrovni užívateľského profilu podporu aj pre tieto verzie syntézy. Všetky spomenuté vlastnosti sa dajú zhrnúť pod pojem adaptabilnosť. Adaptívne vlastnosti otvárajú nové riešenia a uľahčujú prácu programátorom, obohacujú syntézu a pridávajú jej inteligentnú povahu. Adaptabilnosť sa jednoznačne dosahuje v systéme inteligentného učenia.

Modul fonetickej transkripcie pracuje s LTS pravidlami a slovníkom. Požiadavka označená ako naučenie nového slova spôsobí uloženie tohoto slova a jeho SAMPA prepisu do samostatného slovníka pre bežné výnimky. Zmeny sa prejavia okamžite. Požiadavka na naučenie novej predpomy uloží predponu do databázy predpon. Databáza predpon sa vyhodnocuje ak je pri fonetickej transkripcii aktivovaná funkcia detekcie predpony. Požiadavka na naučenie novej prípony sa spracuje rovnakým spôsobom. Požiadavka na naučenie nového cudzieho slova, resp. slovotvorného základu, ktorého výslovnosť je výnimkou pre slovenčinu sa spracuje uložením cudzieho slova do slovníka cudzích slov. Špeciálny algoritmus fonetickej transkripcie kontroluje potom každé slovo, či v sebe neobsahuje slová zo slovníka cudzích slov.

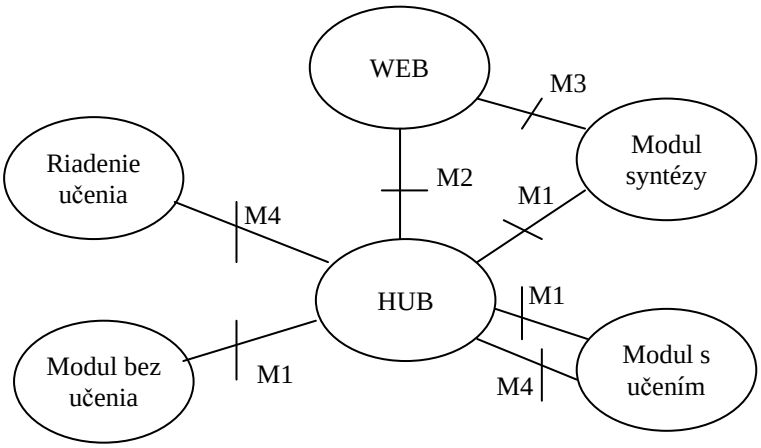
6.1 Návrh

V tejto časti sa budeme venovať podrobnejšiemu popisu návrhu modulárneho syntetizátora s učením.

6.1.1 Zjednodušená bloková schéma modulárneho syntetizátora s učením

Na obrázku (Obr. 19.) sa nachádza zjednodušená bloková schéma modulárneho syntetizátora s učením. Tento koncept vychádza zo zvyčajnej hviezdicovej topológie modulárneho syntetizátora v strede ktorej je umiestnený hub. V takomto systéme si môžme všimnúť 4 druhy rozhraní. Od

typického modulárneho syntetizátora bez učenia sa tento koncept odlišuje pridaním nového modulu „Riadenie učenia“ a definovaním nového rozhrania pre učenie M4.



Obr. 19 Sieťová architektúra modulárneho syntetizátora s učením

V modulárnom syntetizátore sa môžu nachádzať aj moduly, ktoré nepodporujú učenie a nemajú implementované rozhranie M4. Každý modul, ktorý plní nejakú funkcionalitu pri spracovávaní dát v syntéze musí mať rozhranie typu M1 s výnimkou webu. Modul ktorý podporuje učenie disponuje aj rozhraním M4, napriek tomu musí mať aj rozhranie M1. Modul riadenia učenia má iba rozhranie M4, pretože je zapojený do riadenia požiadaviek učenia, avšak na samotnej syntéze sa nepodiela. Osobitosťou je rozhranie M3. Toto rozhranie súži na komunikáciu modulu syntézy s webom. Má len jednu úlohu a ňou je stiahnutie súboru syntetizovanej reči pomocou FTP aby mohla byť prehratá užívateľovi.

6.1.2 Protokolový zásobník a špecifikácia rozhraní

Rozhranie M1 (Obr. 20.) slúži na komunikáciu medzi hubom a modulmi. Používa na najvyššej vrstve protokol XML RPC. Vzdialené volanie procedúry funguje tak, že hub je v roli klienta a modul v úlohe servera. Modul je v stave počúvania a čaká na volanie klienta. Spojenie sa uskutočňuje pomocou soketu, ktorý je charakterizovaný trojicou: IP adresa, port a protokol. Ak hub dostane správu z webu, spracuje ju a potrebuje dáta poslať niektorému modulu na spracovanie. Hub vyberie potrebný modul a jeho funkciu. Jeden modul môže mať ľubovoľne veľa registrovaných funkcií. Aj vnútri samotného modulu môžu byť ďalšie XML RPC servery a klienti. Tým sa tu nebudeme zaoberať, to je otázka vnútornej štruktúry modulu. V tejto chvíli je modul ponímaný ako „black box“, ktorý má definované rozhranie, funkciu a formát vstupných a výstupných dát. Hub iniciuje volanie funkcie a posiela vstupné dáta. Potom hub čaká, kým modul dokončí spracovanie dát. Napokon modul odpovie a pošle výstupné spracované dáta. Hub pokračuje tým že vezme dáta ktoré prijal od modulu a zavolá podľa potreby ďalšiu funkciu toho istého alebo iného modulu. Takto proces syntézy pokračuje, až kým sa nedopracuje k výsledku. Zvyčajne proces prebieha tak, že sa moduly volajú rad za radom, každý jeden krát a na každom module sa volá jedna, vždy tá istá funkcia. Zdokonaľovanie a obohacovanie procesov syntézy však veľmi pravdepodobne zapríčiní vybočenie z tohto zjednodušeného scenáru. Po dokončení sa výsledky zozbierajú a pošlú na web.

XML-RPC
HTTP
TCP
IP
Datalinková vrstva
Fyzická vrstva

Obr. 20 Rozhranie M1

Rozhranie M2 (Obr. 21.) je typické pre komunikáciu medzi hubom a webovým GUI. Pri tejto komunikácii je vhodnejšie použiť čisto HTTP namiesto XML RPC protokol aj keď zvyčajne GUI tiež čaká na odpoveď syntetizátora. Výber protokolov môže byť ovplyvnený aj programovacím jazykom v ktorom je daná časť naprogramovaná.

HTTP
TCP
IP
Datalinková vrstva
Fyzická vrstva

Obr. 21 Rozhranie M2

Rozhranie M3 (Obr. 22.) používa web GUI na získanie finálneho „.wav“ súboru. Pri základnej konfigurácii modulov tento súbor vygeneruje modul difónovej, alebo inej syntézy (korpusovej, alebo HMM). Ak je pričlenený aj modul vytvárania prozódie, má vplyv na výslednú reč. Je možné audio aj streamovať, ale my uprednostňujeme tento spôsob, pretože je jednoduchší a spĺňa požiadavky.

FTP
TCP
IP
Datalinková vrstva
Fyzická vrstva

Obr. 22 Rozhranie M3

Rozhranie M4 (Obr. 23.) má rovnaké protokoly ako rozhranie M1, avšak súži na komunikáciu v systéme učenia. Prepájacím prvkom systému učenia ostáva hub, avšak riadenie učenia má na starosti osobitný modul. V praxi množstvo dát spracovávaných v systéme učenia je výrazne menšie ako v samotnej syntéze. Požiadavky učenia majú vyššiu prioritu ako požiadavky syntézy. Počas adaptácie modulov, musí byť syntézy pozastavená. Keď sa modul naučí nové pravidlo, pokračuje spracovávanie dát syntézy podľa pozmenených pravidiel.

XML-RPC*
HTTP
TCP
IP
Datalinková vrstva
Fyzická vrstva

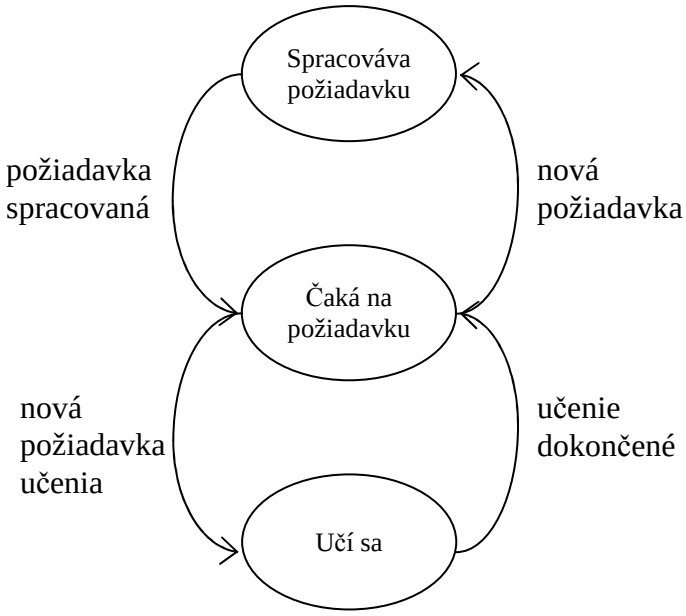
Obr. 23 Rozhranie M4

* s prispôsobením pre funkcie učenia

6.1.3 Stavové diagramy

Jednotlivé moduly môžme charakterizovať z hľadiska stavov v akých sa môžu nachádzať. Nasledovný stavový diagram definuje, čo iniciuje zmenu stavu modulu a aká operácia vedie k akému stavu.

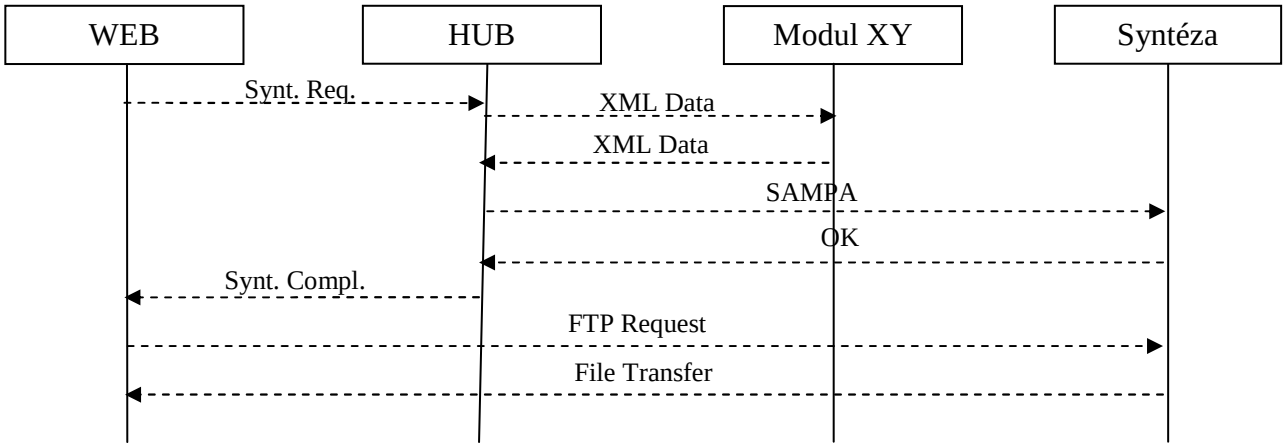
Modul s učením sa nachádza štandardne v stave čakania na požiadavku (Obr. 24.). Ak prijme požiadavku na syntézu, dostane sa do stavu spracovávania tejto požiadavky. V tomto stave nemôže prijať požiadavku na učenie. Keď sa dokončí spracovávanie dát, ktoré sa odošlú do hubu, modul prechádza do stavu počúvania, z ktorého sa môže dostať do stavu učenia ak prijme požiadavku na učenie. Modul bez podpory učenia má iba dva stavy.



Obr. 24 Stavový diagram modulu s učením

6.1.4 Procedúry výmeny správ

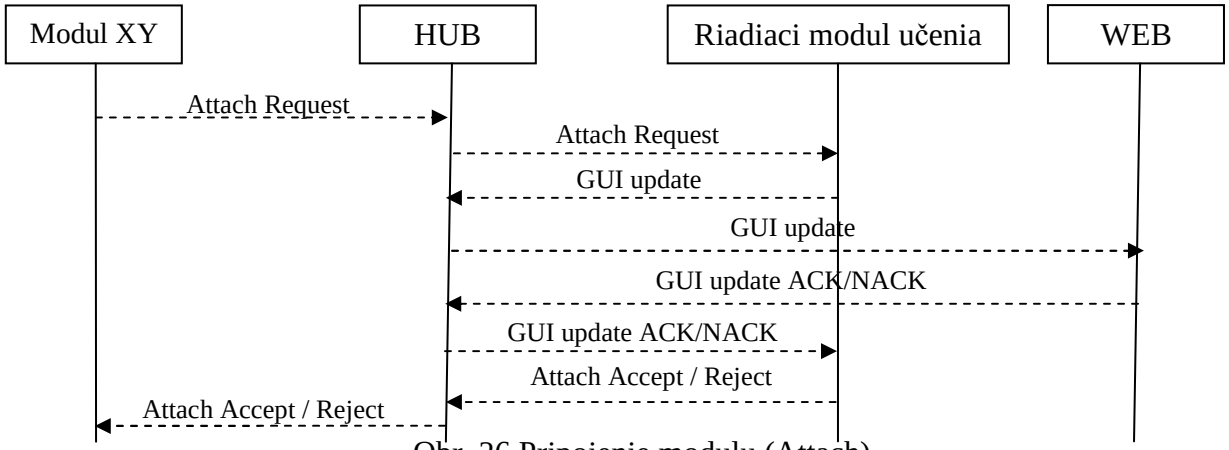
V tejto časti vysvetlíme návrh toku správ pri jednotlivých scenároch. Najprv uvedieme scenár bez učenia (Obr. 25.). Takáto komunikácia je implementovaná v rámci modulárneho syntetizátora. Užívateľ zadá vstupné dáta a čo s nimi chce spraviť. Web GUI vygeneruje a pošle správu do hubu. Hub vyberie z tejto požiadavky dáta a pošle ich na spracovanie sekvenčne do rôznych modulov. Modul fonetickej transkripcie vygeneruje SAMPA prepis. Hub pošle SAMPA prepis spolu s doplňujúcimi informáciami do modulu syntézy. Syntéza vygeneruje reč a odpovie hubu správu s výsledkom, ale nepošle samotný „.wav“ súbor, ten uloží do určeného priečinka. V odpovedi uvedie link na tento adresár. Hub oznámi webu ukončenie procesu syntézy a web si stiahne cez rozhranie M3 „.wav“ súbor a prehrá ho užívateľovi.



Obr. 25 Všeobecný prípad pri modulárnom syntetizátore bez učenia

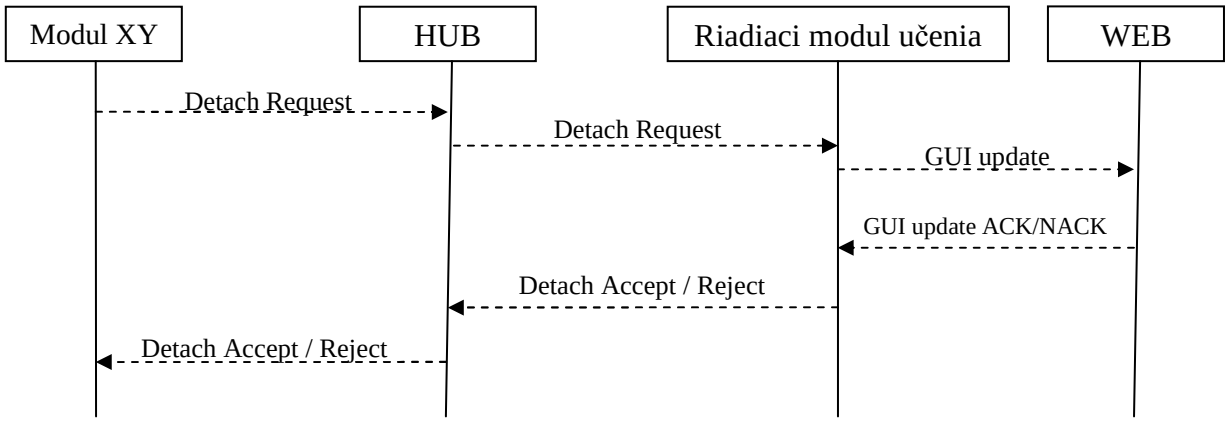
Výhodou modulárneho syntetizátora je možnosť pridať a odobrať, resp. vymeniť nejaký modul. Niektoré moduly sú povinné (GUI, hub, fonetická transkripcia, syntéza) iné sú doplnkové (analýza slovných druhov, kontextová analýza, číslovky, skratky). Doposiaľ fungoval modulárny syntetizátor tak, že akákoľvek zmena v moduloch vyžadovala reštart systému a často aj prispôsobenie vstupov a výstupov incidujúcich modulov. V našom návrhu predpokladáme pripojenie aj odpojenie modulu za behu modulárneho syntetizátora. Aby to bolo možné navrhli sme procedúru „attach“ a „detach“.

Pri pripojení modulu pošle modul správu „Attach Request“ do hubu (Obr. 26.). Ak modul ktorý žiada o pripojenie disponuje rozhraním M4, oznámi hub modulu riadenia učenia prítomnosť nového modulu. Nasleduje update webového GUI. Pridajú sa elementy súvisiace s funkcionalitou daného modulu. Úspešné dokončenie procedúry sa oznámi správou „Attach Accept“, neúspešné správou „Attach Reject“. Odmietnutie modulu môže vykonať riadiaci modul učenia, ak modul nemožno zosúladiť s tým, čo už momentálne v syntetizátore beží.



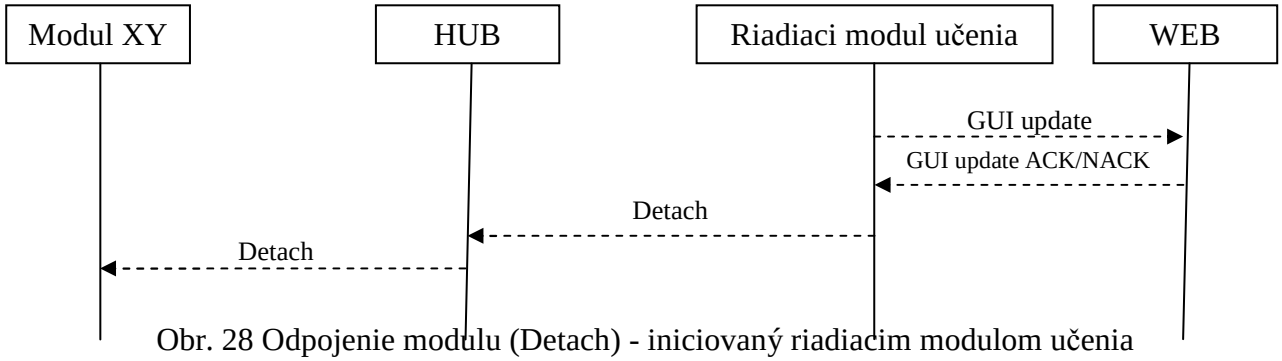
Obr. 26 Pripojenie modulu (Attach)

Korektné odpojenie modulu sa uskutoční vykonaním procedúry „Detach“ (Obr. 27.). Modul oznámi, že sa chce odpojiť, Riadiaci modul iniciuje prispôsobenie webového GUI tak, aby užívateľ už viac nemohol požadovať funkcionalitu zastrešovanú modulom, ktorý sa ide odpojiť. Predíde sa tak havárii systému, ktorý by čakal na odpoveď neexistujúceho modulu.



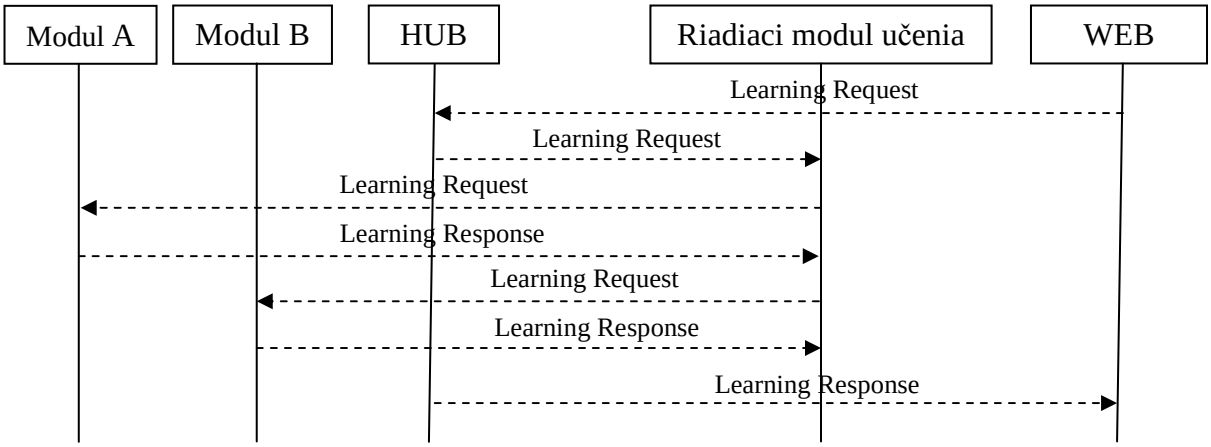
Obr. 27 Odpojenie modulu (Detach) - iniciovaný modulom

V prípade potreby, môže byť modul odmietnutý, odpojený z iniciatívi modulu pre riadenie učenia (Obr. 28.). Napríklad ak pripojíme nový modul, ktorý má lepšie vlastnosti ako iný pripojený, možno pôvodný odpojiť a nahradiť novým. Pri návrhu týchto scenárov sme sa inšpirovali niektorými scenármi z [44].



Obr. 28 Odpojenie modulu (Detach) - iniciovaný riadiacim modulom učenia

Učenie modulu sa týka len modulov s rozhraním M4. Učenie je iniciované požiadavkou užívateľa v GUI (Obr. 29.). Táto požiadavka sa cez hub smeruje do modulu riadenia učenia. Modul riadenia učenia rozhodne aké zásahy a do ktorých modulov je potrebné vykonať, iniciuje ich a čaká na odpoveď modulov. Moduly vykonajú potrebné zmeny vo svojej funkcionalite a odpovedia, či sa podarilo vykonať požadované úpravy. Tieto správy prechádzajú taktiež cez hub.



Obr. 29 Učenie modulu

6.1.5 Smerovanie požiadaviek

Je potrebné aby sa riadiaci modul vedel rozhodnúť, kam sa má príslušná požiadavka na inteligentnú syntézu preposlať, aby bola správne interpretovaná. Smerovanie môže byť vopred definované pre jednotlivé elementy GUI. Reálna prevádzka prináša mnohé úskalia, ktoré sa pri laboratórnom testovaní nemusia ukázať. Vopred sa počíta s tým, že na vstup syntetizátora príde viacero požiadaviek naraz v jednom čase. Tento problém sa rieši viacvláknovým spracovávaním [39]. Pri systéme inteligentného učenia sa predpokladá rádovo desať až stonásobne menšie zaťaženie systému, preto nie je potrebné dimenzovať systém na súbežné spracovávanie viacerých požiadaviek. Vhodné je, aby jednotlivé obslužné moduly akceptovali zmeny inteligentného učenia okamžite a hneď ich aj vykonali a svoju ďalšiu činnosť vykonávali už v súlade s týmito zmenami, bez potreby reštartu celého systému, resp. príslušného modulu.

7 Metóda pre nájdenie optimálnej súčinnosti slovníka fonetickej transkripcie a LTS pravidiel

LTS pravidlá rozhodujú podľa kontextu prepisovaného písmena. Ortografický kontext písmena je chápaný ako príznakový vektor, ktorý sa vyhodnocuje vo vetvách binárnych stromov CART. LTS pravidlá sa trénujú pomocou trénovacích dát. Trénovacie data vzniknú spojením a špecifickou úpravou viacerých slovníkov [32].

Po natrénovaní sú LTS pravidlá schopné nahradiť veľkú časť pôvodných slov zo slovníkov, pretože slová prepísané podľa nových LTS pravidiel sú identické s pôvodným priradením v slovníkoch. Podiel slov z pôvodných dát o ktorom platí spomenutá vlastnosť identickosti môže byť rôzny. Závisí od nastavenia hranice citlivosti trénovania a taktiež od rôznorodosti vstupných trénovacích dát Po trénovaní je vhodné každé slovo z pôvodných slovníkov prepísať pomocou novovzniknutých LTS pravidiel. Následne, daný prepis porovnať s pôvodným prepisom v slovníku. Ak sa prepisy rovnajú, sú identické, slovo sa už naďalej nebude uchovávať v slovníku. Ak sú prepisy rozdielne, slovo v slovníku zachováme. Takéto slovo sa stáva výnimkou z pravidiel. Pravidlá nie sú schopné pokryť všetky slová. Prepis podľa slovníka má vždy vyššiu prioritu. LTS pravidlá sa používajú vždy až na koniec, keď slovo nemožno inak prepísať. Takýmto princípom dosahujeme, že všetky slová, ktoré sme mali k dispozícii na trénovanie, budú fonetickou transkripciou správne prepísané. Okrem nich budú správne prepísané i všetky tie slová, v ktorých sa uplatňujú pravidlá fonetického prepisu zakotvené v LTS pravidlách.

Môže sa stať, že nejaké slovo, ktoré nie je v slovníku sa tiež vymyká LTS pravidlám prepisu. V takomto prípade dochádza k nesprávnemu fonetickému prepisu slova. Nesprávny prepis sa prejaví napríklad nesprávnym spodobovaním, zmäkčovaním, resp. neuplatnením týchto javov tam, kde je to potrebné. V tomto prípade je potrebné, aby sa do procesu prepisu zapojil systém inteligentného učenia rečového syntetizátora. Výsledkom učenia, nech už sa uskutoční akýmkoľvek spôsobom, je nové slovo, ktoré je klasifikované ako výnimka vzhľadom na LTS pravidlá. Toto slovo sa potom zapíše buď do slovníka fonetickej transkripcie, alebo do slovníka výnimiek. Slovník výnimiek je tiež slovníkom fonetickej transkripcie. Po určitom čase fungovania rečového syntetizátora spolu s funkcionalitou inteligentného učenia môže byť do slovníka výnimiek pridaných 10, 100, ale i tisíce slov. Každopádne proces pridávania slov do slovníka výnimiek narúša rovnováhu medzi slovníkom a LTS pravidlami.

Tu sa otvára otázka, ako by bolo možné odmerať, resp. klasifikovať, kedy sa slovník už natoľko rozrástol, že je vhodné pretrénovať LTS pravidlá. Proces pretrénovania je určite z času na čas užitočný. Tento proces môže okrem pridania nových pravidiel, zabezpečiť aj vyčistenie od zastaralých pravidiel, ktoré už stratili svoj význam. Takýto systém je veľmi adaptabilný a učenlivý. Týmto princípom by bolo možné natrénovať syntetizátor s nulovými počiatočnými znalosťami o fonetických vzťahoch prepisu. Tento postup môže nájsť uplatnenie pre nárečia, dialekty, dokonca s istými úpravami i pre cudzí jazyk. Akonáhle sa uzná za vhodné, aby sa LTS pravidlá opäť pretrénovali, musí sa opäť použiť pôvodná kompletná databáza, ku ktorej sa pridá ešte aj novovzniknutý slovník výnimiek. Ak by nastala situácia, že niektoré slovo za slovníka výnimiek stanovuje slovu ktoré je už v kompletnej databáze prítomné s inou výslovnosťou, nahradí sa novou výslovnosťou. Trénovanie nových LTS pravidiel sa uskutočňuje vždy od začiatku nanovo na základe celej dostupnej, čo najbohatšej databázy. Nie je možné robiť dobré dotrénovanie aby sa štruktúra jestvujúcich stromov menila podľa sady výnimiek. Takýto postup by ľahko viedol k znehodnoteniu LTS pravidiel. Predovšetkým preto, že nie je možné spätne zistiť na základe akej váhy bola vetva binárneho stromu vytvorená. Kompletne celá databáza, ktorú máme k dispozícii a na báze ktorej sa trénujú LTS pravidlá môže byť označená ako korpus. Tento korpus sa nepoužíva pri fonetickej transkripcii. Tento korpus je potrebné mať niekde uložený a časom ho obohacovať.

7.1 Princíp pretrénovania LTS pravidiel

Procedúra trénovania je vo všeobecnosti definovaná nasledovnými krokmi:

- Predspracovanie slovníka (databázy) do vhodnej množiny pre trénovanie.
- Definovanie množiny dovolených párov písmeno – fonéma.
- Stanovenie pravdepodobností pre každý pár písmeno – fonéma.
- Zarovnanie písmen do rovnako veľkej množiny foném. Podľa potreby použitie prázdnej fonémy „epsilon“.
- Extrahovanie vhodných dát pre trénovanie jednotlivých písmen.
- Výstavba CART modelov pre analyzované písmeno a jeho kontext [40].

Optimalizačný algoritmus bude mať za úlohu z času na čas overiť proporcionalitu nástrojov fonetickej transkripcie a vykonať potrebné úpravy.

7.2 Metódy hodnotenia

Proces hodnotenia môže vychádzať z rozličných aspektov. V prvom rade musí byť splnená požiadavka, že vo vytvorených nástrojoch bude obsiahnutá všetká informácia obsiahnutá v originálnom korpuse. To v praxi znamená, že ak vezmeme ľubovoľné slovo z korpusu a necháme ho prepísať fonetickou transkripciou ktorá používa LTS pravidlá natrénované z tohto korpusu a slovník výnimiek zostavený podľa týchto pravidiel a výsledok transkripcie porovnáme so SAMPA reťazcom z originálneho korpusu, tieto reťazce sa musia zhodovať. Takéto riešenie nazývame aj bezstratovou transkripciou. Pomocou štatistík je možné navrhnúť systém, ktorý bude pokrývať napríklad len 90%

najčastejších slov a pod. Takéto riešenie by sme uplatnili ak by sme mali veľmi obmedzené možnosti na pamäť.

Základná metóda je založená na porovnaní počtu elementov slovníka a LTS pravidiel. Cieľom je minimalizovať celkový rozsah týchto nástrojov. Elementom slovníka je slovo a jeho SAMPA prepis. Elementom LTS pravidiel môže byť jeden riadok pravidiel, jeden uzol, alebo jeden list. My budeme považovať za optimálne tie pravidlá, ktoré majú minimálnu sumu podľa vzťahu (14). My sme vygenerovali N rôznych LTS pravidiel, ktoré sa od seba líšili napríklad parametrom stop value, metódou trénovania alebo korpusom. Aj keď náročnosť prepisu pomocou slovníka a pomocou pravidiel je mierne odlišná. Preto zavádzame špecifické váhy pre zohľadnenie týchto rozdielností. E_D je počet elementov slovníka, W_D je špecifická váha pre slovník, E_R je počet riadkov LTS pravidiel a W_R je špecifická váha LTS pravidiel. Váhovacie konštanty nastavujú požadovanú proporciu medzi slovníkom a pravidlami.

$$\min\{E_{D_i} \cdot W_D + E_{R_i} \cdot W_R\}_{\forall i \in N} \tag{14}$$

Porovnanie pravidiel a slovníka môžme realizovať okrem základnej metódy aj nasledovnými metódami:

- Vyvážená
- S použitím testovacej databázy
- Realistická
- Štatistická

Vyvážená metóda je podobná základnej. Rozdiel tkvie v tom, že pri tejto metóde nahradíme počet riadkov LTS pravidiel počtom uzlov. Jeden uzol je jedna „if“ podmienka. Metóda, ktorá pracuje s testovacou databázou vyžaduje textové dáta. Pri tejto metóde počítame koľko slov bolo prepísaných pomocou slovníka v porovnaní k počtu slov prepísaných pravidlami.

E_D je počet slov spracovaných slovníkom, W_D je váhovacia konštanta slovníka, E_R je počet slov prepísaných pomocou LTS pravidiel. W_R je váhovacia konštanta LTS pravidiel. $Corr_{Di}$ je počet slov prepísaných pomocou i-tej iterácie slovníka. Analogicky $Corr_{Ri}$ je počet slov správne prepísaných pomocou i-tej iterácie LTS pravidiel. Incorri je počet slov nesprávne prepísaných pomocou nástrojov i-tej iterácie. W_{ERR} je penalizačná váha chybného prepisu. V tomto kroku zachovávame konvenciu: $W_R < W_D < W_{ERR}$ (15).

$$\min\{E_{D_i} \cdot W_D \cdot Corr_{D_i} + E_{R_i} \cdot W_R \cdot Corr_{R_i} + Incorr_i \cdot W_{ERR}\}_{\forall i \in N} \tag{15}$$

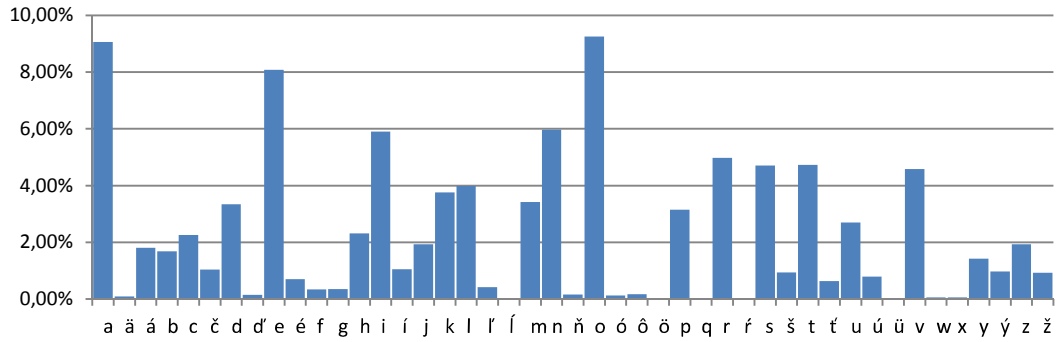
Ak chceme porovnávať ešte podrobnejšie, vezmime do úvahy že $G=(g_1,g_2 \ldots g_n)$ je postupnosť grafém ktorá reprezentuje ortografické dáta jedného slova. Potom $P=(p_1,p_2 \ldots p_n)$ je postupnosť foném, ktorá vyjadruje ortoepickú formu slova. Môžme definovať vzťah $B(g_i \rightarrow p_i)$ ako mieru rozvetvenosti binárneho klasifikačného stromu sledujúc cestu pri prepise grafémy gi na fonému p_i. Potom $P(g_i \rightarrow p_i)$ je pravdepodobnosť prepisu grafémy gi na fonému pi (16).

$$\begin{gathered} G = (g_1, g_2 \ldots g_i \ldots g_n) \\ P = (p_1, p_2 \ldots p_i \ldots p_n) \\ g_i \leftrightarrow p_i \end{gathered} \tag{16}$$

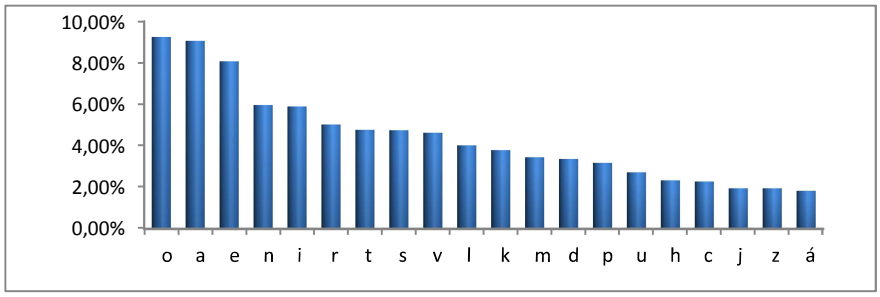
Realistická metóda sa líši od predošlej len použitím väčšieho množstva textových dát. Texty môžu byť cielene zamerané na jednu oblasť, alebo sa môžu týkať všeobecne rôznych žánrov a tém. Čím viac textových dát použijeme, tým lepšie výsledky môžme očakávať.

Napokon štatistická metóda berie do úvahy zákonitosti jazyka. Táto metóda používa aj informáciu o priemernej dĺžke slova v jazyku, pravdepodobnosť slov a pravdepodobnosť grafémy v slove.

Priemerná dĺžka slova v slovenčine je podľa nášho skúmania 4.343103910869559 znakov. Okrem toho sme vypočítali pravdepodobnosť každého písmena v slovách všeobecne (Obr. 30).



Obr. 30 Pravdepodobnosť výskytu písmena v slove



Obr. 31 20 najčastejšie sa vyskytujúcich písmen v slovenských textoch

20 najčastejšie sa vyskytujúcich písmen v slovenských textoch pre prehľadnosť sumarizujeme na (Obr. 31.). Pravdepodobnosť výskytu grafémy nám ukazuje napríklad aj to, ktoré binárne stromy sú nakoľko často používané. Napríklad ak je písmeno „n“ 20 krát častejšie používané ako písmeno „f“, je jeho strom 20 krát viac dôležitý ako strom písmena „f“. Všimli sme si že medzi 5 najčastejšími písmenami sú 4 samohlásky a iba 1 spoluhláska. To nám potvrdzuje ak logická úvaha, že je ich v jazyku menej ako spoluhlások a navyše poskytujú reči harmonickú zložku ktorá je dôležitá pre striedanie znelých a neznelých prvkov reči.

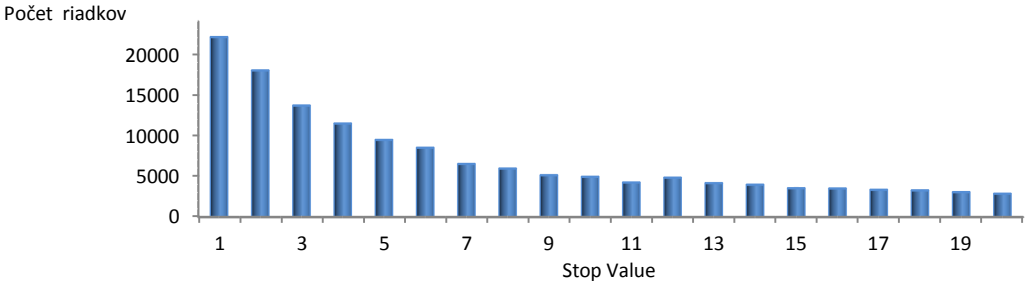
7.2.1 Slovník výnimiek

Pre vyhodnotenie kvality a charakteru získaných pravidiel sme použili metódu ktorá predpokladá vytvorenie slovníka výnimiek pre každú iteráciu vygenerovaných LTS pravidiel. Proces vyhodnotenia začína ihneď po dokončení LTS pravidiel. Slovník výnimiek vytvárame tak, že každé slovo z korpusu, na ktorom sa trénovalo, necháme prepísať pomocou novovytvorených LTS pravidiel. Každý SAMPA prepis porovnáme s pôvodným prepisom, ktorý sa nachádza v korpuse. Ak sa prepis v niečom odlišuje, slovo zaradíme do slovníka výnimiek. Teda takto vytvorený slovník sa stáva slovníkom výnimiek z tejto konkrétnej verzie LTS pravidiel a bude používaný len pre túto verziu pravidiel.

7.2.2 Overenie navrhnutej metódy

Navrhnutú metódu pre optimalizáciu slovníka a LTS pravidiel sme aj implementovali a otestovali. Testovanie tejto metódy nám prinieslo predovšetkým dve podstatné informácie. Prvou bolo potvrdenie že charakter závislosti rozsahu nástrojov je skutočne taký ako sme očakávali. Druhou

bolo zistenie, že uvedená teória je istým spôsobom zjednodušením reality. Dosvedčuje nám to napríklad skutočnosť, že sa nám nepodarilo natrénovať také pravidlá, ktoré by mali slovník výnimiek s nulovým počtom výnimiek. Niektoré charakteristiky by bolo možné zlepšiť použitím lepšieho korpusu, boli sme však značne limitovaný obmedzenosťou našich zdrojov. Osobitnú pozornosť sme venovali príprave databázy. Tým sa nám podarilo získať korpus s takmer 80000 slovami vstupujúcimi do trénovania. Nasledujúci graf (Obr. 32.) znázorňuje rozsah LTS pravidiel. Tieto pravidlá boli trénované "Stepwise" metódou. Stop value bola volená z intervalu 1 až 20. Môžme vidieť jasne exponenciálne klesajúcu charakteristiku. Čím viac vzorov pre dané pravidlo požadujeme, tým menej pravidiel vznikne. Z tejto charakteristiky môžme vyčítať, že základné pravidlá sú vždy pomerne bohato zastúpené v korpuse. Preto v rozsahu 9 až 19 sme nepostrehli veľký nárast pravidiel. Špecifiká fonetického prepisu väčšinou disponujú podporou menšou ako 10 vzorov v trénovacej databáze. Môže sa nám to javiť pomerne malá podpora pri počte trénovacích vektorov, ktorý sa pohybuje rádovo v stovkách tisíc. Musíme však brať do úvahy že ide o kontextovo závislé javy, pričom väčšia časť kontextov je spätá s bežnými vzťahmi.

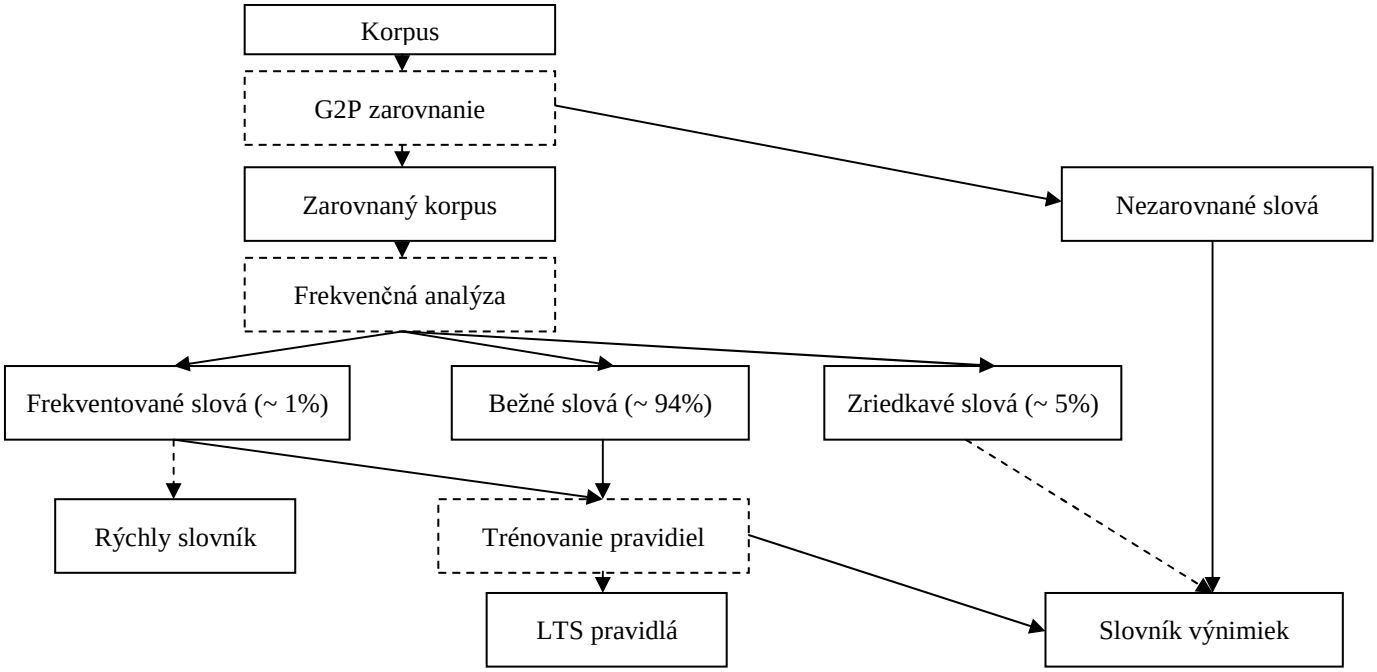


Obr. 32 Rozsah LTS pravidiel pri zdokonalenom G2P zarovnaní a použití stepwise metódy

V nasledujúcej časti vezmeme do úvahy, že slová sa nevyskytujú rovnako frekventovane v jazyku, ale rôzne slová sa vyskytujú rôzne často. Rozhodli sme sa vykonať frekvenčnú analýzu slov z hľadiska frekvencie ich výskytu v jazyku.

7.2.3 Frekvenčná analýza slov

V nasledujúcom kroku sme sa rozhodli zlepšiť výsledky použitím ešte inej metódy. Vychádza z pozorovania, že nie všetky slová sa v jazyku vyskytujú rovnako často.

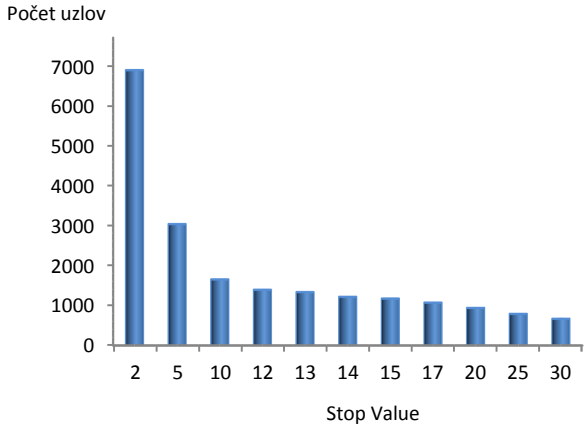


Obr. 33 Trénovanie LTS pravidiel štatistickou metódou

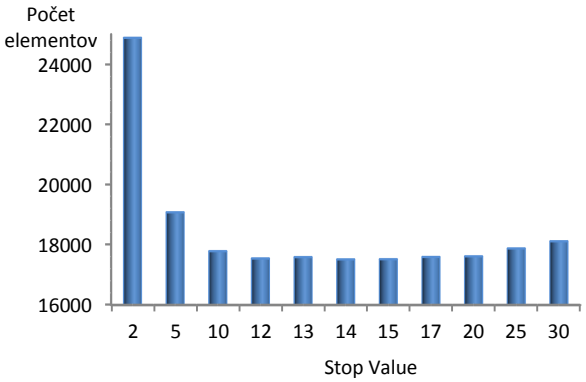
Celý proces analýzy, filtrovania a spracovania korpusu vrátane vytvorenia databázy a LTS pravidiel je zobrazený na nasledujúcej blokovej schéme (Obr. 33.).

Zamerali sme sa na podrobné vyhodnotenie v rámci intervalu 1 až 30. V nasledovných grafoch sú znázornené výsledky. Závislosť počtu uzlov pravidiel (podľa vyváženej metódy) sme vyniesli do grafu (Obr. 34).

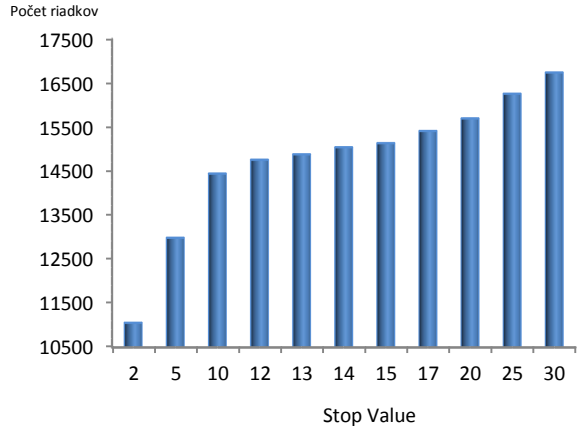
Nasleduje rozsah slovníka výnimiek (Obr. 35.) a celkový rozsah aby bolo možné lokalizovať minimum (Obr. 36). Graf (Obr. 37) je zameraný na detail diferencie najbližších iterácii s optimálnou iteráciou.



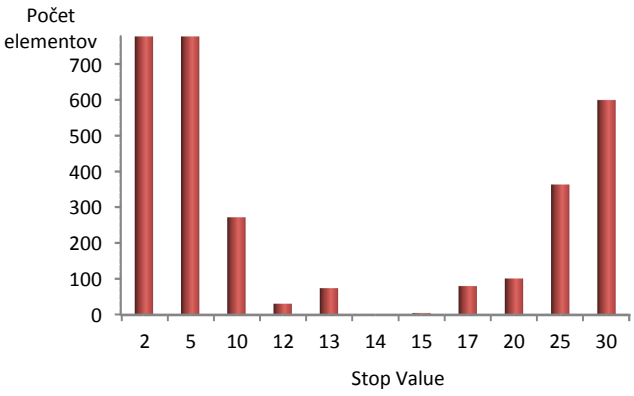
Obr. 34 Počet uzlov LTS pravidiel



Obr. 36 Celkový rozsah nástrojov fonetickej transkripcie



Obr. 35 Rozsah slovníka výnimiek v intervale 1 až 30



Obr. 37 Detail rozdielu voči optimálnej iterácii (stop value = 14)

Na základe uvedených výsledkov môžme uzavrieť, že pre náš korpus sa optimálne pravidlá so slovníkom výnimiek vygenerovali pri použití metódy stepwise pri minimálnej podpore 14 vzorov.

8 Metóda G2P zarovnania

V rámci G2P zarovnania sa musíme vysporiadať s viacerými jazykovými javmi. Prvým je prípad keď 2 grafémy zodpovedajú 1 foném. Takýto prípad sa rieši vkladaním prázdnej fonémy (epsilón) do SAMPA reťazca. Takýmto spôsobom sa riešia aj podobné prípady (Tab. 10.).

Tab. 10. G2P vzťah 2 ku 1

príklad	vstup	výstup	počet
únia	I_^a	I_^a _epsilon_	2=1+(1)
úsilie	I_^E	I_^E _epsilon_	2=1+(1)
ďalšiu	I_^U	I_^U _epsilon_	2=1+(1)
archív	x	x _epsilon_	2=1+(1)

Tab. 11. G2P vzťah 1 ku 2

príklad	vstup	výstup	počet
boxer	k s	k-s	1:2→1:1
exil	g z	g-z	1:2→1:1

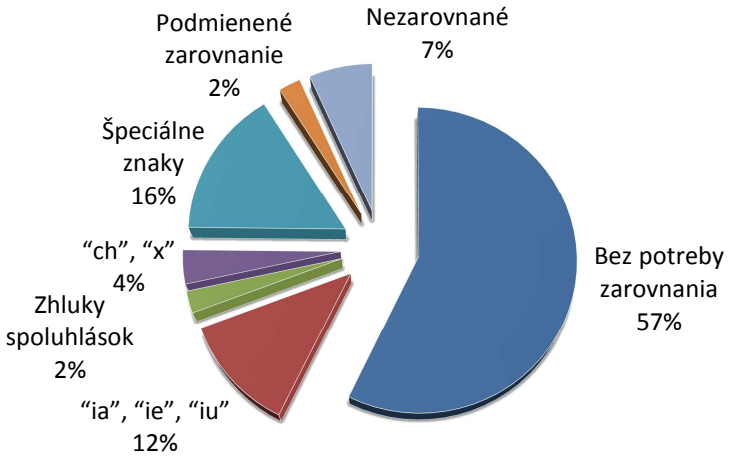
Druhým prípadom je je graféma „x“. Táto graféma sa obvykle vyslovuje dvoma zvukmi v reči, konkrétne „k“ a „s“ alebo „g“ a „z“. Vkladania epsilónu nám v tomto prípade nepomôže, preto že potrebujeme redukovať dĺžku SAMPA reťazca. Naša metóda tento problém rieši fúziou SAMPA znakov (Tab. 11.). Táto úprava sa pri neskoršom spracovaní vráti do pôvodného stavu. Pri dvoch po sebe nasledujúcich spoluhláskach môže nastávať nasledovný jav. Je to prirodzený dôsledok povahy vyslovovania a tvorby reči. Príklady sú uvedené v nasledujúcej tabuľke (Tab. 12.).

Tab. 12. G2P vzťah 2 ku 1 pri spoluhláskových zhlukoch

príklad	vstup	výstup	počet
účinný	[n_>]	[n_>] _epsilon_	2=1+(1)
cenník	[J_>]	[J_>] _epsilon_	2=1+(1)
choďte	[c_>]	[c_>] _epsilon_	2=1+(1)
bezzubý	[z_>]	[z_>] _epsilon_	2=1+(1)
škodca	[ts_>]	[ts_>] _epsilon_	2=1+(1)
vyšší	[s_>]	[s_>] _epsilon_	2=1+(1)
interrupcia	[r_>]	[r_>] _epsilon_	2=1+(1)
predtlač	[t_>]	[t_>] _epsilon_	2=1+(1)
predsudok	[tS_>]	[tS_>] _epsilon_	2=1+(1)
rozsúdiť	[S_>]	[S_>] _epsilon_	2=1+(1)
oddávna	[d_>]	[d_>] _epsilon_	2=1+(1)

príklad	vstup	výstup	počet
subbas	[b_>]	[b_>] _epsilon_	2=1+(1)
polliter	[l_>]	[l_>] _epsilon_	2=1+(1)
subpolárny	[p_>]	[p_>] _epsilon_	2=1+(1)
bohchráň	[x_>]	[x_>] _epsilon_	2=1+(1)
buffo	[f_>]	[f_>] _epsilon_	2=1+(1)
objímme	[m_>]	[m_>] _epsilon_	2=1+(1)
oddťaľovať	[J_>]	[J_>] _epsilon_	2=1+(1)
keďže	[dZ_>]	[dZ_>] _epsilon_	2=1+(1)
odzadu	[dz_>]	[dz_>] _epsilon_	2=1+(1)
mäkkýš	[k_>]	[k_>] _epsilon_	2=1+(1)
markgróf	[g_>]	[g_>] _epsilon_	2=1+(1)

Ak by sme chceli zhrnúť a kvantifikovať zastúpenie jednotlivých javov v jazyku, môžeme použiť korpus. Nasledujúci graf zobrazuje, ako riešenie príslušných fenoménov prispelo k celkovému zarovnaniu slov (Obr. 38.).



Obr. 38. Zarovnanie korpusu

Riešenie zarovnania špeciálnych znakov zahŕňa vysporiadať sa s niektorými prvkami, ktoré sa v našom korpuse nachádzajú a týkajú predovšetkým výrazu slov. Podmienené zarovnanie je proces pri

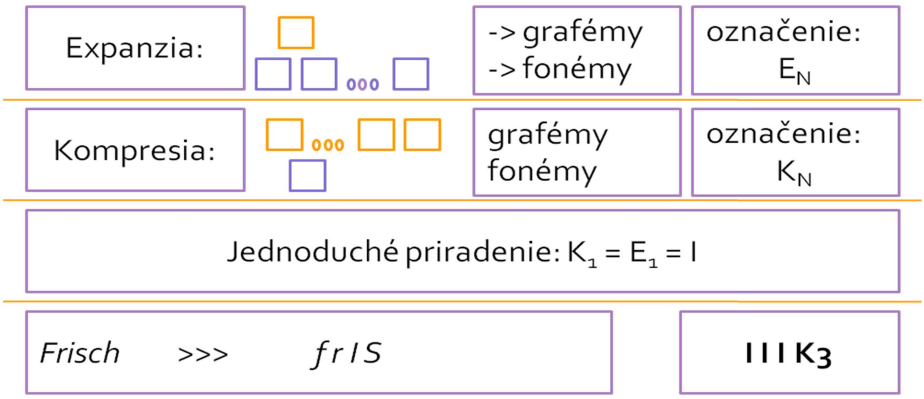
ktorom najprv testujeme prítomnosť istých reťazcov v slove a v SAMPA reťazci a až na základe vyhodnotenia prítomnosti alebo neprítomnosti robíme následnú úpravu. Je dôležité poznamenať, že záleží na poradí v ktorom sa pravidlá G2P zarovnania na text aplikujú.

8.1 Spracovanie výnimiek G2P zarovnania

Na to aby sme vedeli prispôsobiť zarovnanie výnimiek G2P zarovnania potrebujeme vždy keď nastane nejaký nový prípad použiť metódu pre automatické nájdenie nových G2P pravidiel zarovnania. Príkladom sú slová: „frichs, school, packet“. Metóda funguje nasledovne:

- G2P validátor zistí nezarovanané slovo
- vygenerujú sa všetky možnosti zarovnania
- určia sa priority jednotlivých možností
- vygenerujú sa všetky kombinácie každej možnosti
- určí sa súbor štandardných pravidiel
- pre každú kombináciu každej možnosti sa dosadia konkrétne grafémy a fonémy
- priradenia sa automaticky ohodnotia podľa súboru štandardných pravidiel
- kombinácie sa usporiadajú vzostupne podľa hodnotenia
- vyberie sa možnosť ktorá je na prvom mieste a vytvoria sa vzťahy podľa príslušnej masky danej kombinácie – všetky nie 1 k 1 priradenia sú kandidátmi nových pravidiel
- vytvoria sa pravidlá

Pre zrozumiteľnejší detailnejší popis sme si zaviedli nasledovné pojmy. Expanzia je vzťah, kedy sa jedna graféma prepíše pomocou viacerých foném. Ak na prepis jednej grafémy použijeme napríklad 3 fonémy, označujeme to ako expanziu tretieho rádu (E_3) Kompresia je opačný vzťah, označujeme (K_N). Nakoľko platí že $K_1 = E_1$, používame označenie I čo znamená identický počet, G2P zhodu. Nasledovné vzťahy spolu s príkladom na slove „frisch“ sú na nasledujúcom obrázku (Obr. 39.)



Obr. 39 G2P zarovnanie – expanzia a kompresia

Uvažujme, že robíme prepis M písmen textu do N SAMPA znakov (Obr. 40.).

Obr. 40 G2P zarovnanie – prepis

Môžu nastať dve základné situácie: M>N a M<N. M=N nemôže nastať, lebo to je validný stav slova ktorým sa tu nezaoberáme. Keď hlbšie analyzujeme tieto vzťahy prichádzame k nasledovným uzáverom:

- Max. počet expanzií a ich veľkosť je limitovaná počtom foném získaných expanziou na N-1, pričom posledná fonéma je v krajnom prípade kompresiou všetkých ostatných grafém.
- To isté analogicky platí o kompresii, že musí ostať min. jedna graféma, ktorá sa expanduje do všetkých ostatných foném.
- Potom pre G2P platí: I(M<N) = I(M>N) *(počet priamych priradení)*
- E(k, M<N) = K(k, M>N), E(k, M>N) = K(k, M<N)

V ďalšom budeme označovať kompresiu (K_N), expanziu (E_N) aj priame priradenie (I) všeobecne ako G2P vzťah G2P(g,p), kde „g“ označuje počet grafém a „p“ počet foném. Potom pre všetky možnosti G2P zarovnania platí:

- pre g<p: $G2P(g,p) = \{E_k + G2P(g-1,p-k)\}_{k=1}^{p-1}$
- pre g>p: $G2P(g,p) = \{K_k + G2P(g-k,p-1)\}_{k=1}^{g-1}$
- $G2P(1,p) = E_p$, $G2P(g,1) = K_g$, $G2P(g,p) = g*I$ vtedy a len vtedy ak g=p

Keď máme vygenerované všetky možnosti G2P zarovnania, potrebujeme určiť prioritu jednotlivých možností. Určenie priorít je založené na jednoduchej súvahe vychádzajúcej z pozorovania jazykových javov. Uvážme napríklad, že “I I I K2” má väčšiu prioritu ako “E3K4”. Preto kategoricky určíme váhy, napríklad nasledovne: I ... 100; E2/K2 ... 90; E3/K3 ... 80; atď.. V postupe tejto metódy sme spomenuli, že určíme súbor štandardných pravidiel. Takéto pravidlá určíme pomocou metódy pre nájdenie kontextovo nezávislých alternatívnych výslovností. Tejto problematike sa venuje osobitná časť práce. Výsledok môže vyzerať nasledovne (Tab. 13.). Vyberáme len niektoré písmená pre ilustráciu. Hviezdičkou sú označené tie SAMPA znaky, ktorých pravdepodobnosť je menšia ako 1%.

Tab. 13. Súbor štandardných pravidiel – ukážka

Písmeno	SAMPA zank (pravdepodobnosť)
č	tS (0.9334) dZ (0.0375) ε (0.0249) [tS_>]* ts*
d	d (0.6173) J\ (0.1657) t (0.0895) ts (0.0435) dz (0.0288) [ts_>] (0.0123) [tS_>]* ε* [t_>]* [d_>]* J-* [c_>]* [dz_>]* [J_>]* [dZ_>]*
ď	J\ (0.6999) [c_>] (0.2559) c (0.0194) J- (0.0138) ε* [dZ_>]* ts* [t_>]* t*
Í	I=: (1.0000)
ô	U_^O (0.9980) O*
ž	Z (0.8484) S (0.1369) [S_>]* ε* z* s*

Ukážme si teraz niektoré kroky tejto metódy na príklade slova “frisch”. Pre každú kombináciu každej možnosti dosadíme konkrétne grafémy a fonémy. Teda napr.: 6 -> 4 (Obr. 41.). Následne usporiadame vzostupne kombinácie podľa hodnotenia (obr. 42.).

K2 I I K2 = fr-f, i-r, s-i, ch-S
K3 E2 K2 = fri-f, s-ri, ch-S
atď.

Obr. 41 G2P zarovnanie – príklad prepisu

I I I K3 = f-f, r-r, i-i, sch-S = 85+98+70+2 = 255
I I K2 K2 = f-f, r-r, is-i, ch-S = 85+98+0+2 = 185
K2 I I K2 = fr-f, i-r, s-i, ch-S = 0+0+0+2 = 2
K3 E2 K2 = fri-f, s-ri, ch-S = 0+0+2 = 2
atď.

Obr. 42 Vzostupne usporiadané kombinácie

Napokon vezmeme možnosť ktorá je na prvom mieste a vytvoríme vzťahy podľa príslušnej masky danej kombinácie. Všetky nie 1 k 1 priradenia definujeme ako kandidátov nových pravidiel. Teda v našom prípade “I I I K3 = 255” zodpovedá “f-f OK, r-r OK, i-i OK”. Na základe toho určíme “sch –>S“ ako nové pravidlo.

9 Pôvodné vedecké prínosy

Práca Inteligentné učenie výslovnosti rečového syntetizátora prináša viaceré vlastné vedecké prínosy, ktoré je možné stručne zhrnúť v nasledujúcich bodoch:

- **Navrhnutá metóda kontextovo závislého predspracovania textu pre syntézu reči** (navrhnutá a implementovaná kontextová analýza textu, tvorba databázy pre kontextovú analýzu textu, automatické určenie pravdepodobnosti gramatickej správnosti slova, určenie miery špecifickosti slova, určenie dôležitosti slova v tematickej doméne, vytvorenie spektra tém textu) - *Touto problematikou sa zaoberala aj práca [42]. Prínosom tejto práce je návrh novej metódy, ktorá rozširuje počtet tematických domén z 3 na 55 a dosahuje lepšiu úspešnosť* - 83,33%. *Okrem toho práca popisuje postup, ako možno na základe zistenia témy riadiť prepis čísloviek, skratiek. Práca prináša aj návrh a implementáciu zdokonalenia algoritmu fonetickej transkripcie pomocou detekcie prípon, predpon a cudzích slov. - Výsledky a prínos týkajúci sa tohto bodu boli publikované v [ADE1, AFD2 a AFD3].*
- **Navrhnutá a implementovaná metóda pre nájdenie LTS alternatív a ich pravdepodobností** (príprava text-to-SAMPA korpusu, určenie fonémovej intenzity písmen podľa korpusu, určenie distribúcie grafemickej reprezentácie podľa korpusu) - *LTS resp. G2P alternatívy boli doposiaľ určované lingvistami. Prínosom práce je automatizácia procesu, ďalej odhalenie podstatne väčšieho počtu alternatív. (Napr. v prípade písmena „d“ o 11 viac.) Napokon váhovanie alternatív ich pravdepodobnosťami. - Výsledky a prínos týkajúci sa tohto bodu bol poslaný do časopisu Speech communication. Z dôvodu dlhotrvajúceho recenzného procesu a výhradám jedného z recenzentov sa nepodarilo článok publikovať pred odovzdaním tejto práce.*
- **Navrhnutá a implementovaná metóda korekcie výslovnosti** (pomocou interaktívnych SAMPA znakov, usporiadanie alternatívnych výslovností slov podľa ich pravdepodobností, grafické rozhranie pre inteligentné učenie výslovnosti, generovanie alternatívnych výslovností slova podľa masky) - *Téma opravy výslovnosti syntetizátora bola už rozoberaná v niektorých prácach, napr. [45] a [46]. Návrh v teoretickej rovine predostieral všeobecné princípy. Prínosom tejto práce je konkretizácia viacerých postupov, napríklad aj spôsobu, ako možno určiť pravdepodobnosť alternatívnych výslovností. - Výsledky a prínos týkajúci sa tohto bodu bol publikovaný v [ADE1 a AFC1].*
- **Navrhnutá sieťová architektúra modulárneho syntetizátora s podporou učenia** (architektúra, rozhrania, procedúry, správy) - *Návrh architektúry modulárneho syntetizátora nie je úplne nová vec. Prínosom práce je zakomponovanie učenia a konkretizovaný návrh architektúry, rozhraní, procedúr a správ. - Výsledky a prínos týkajúci sa tohto bodu bol publikovaný v [AFC2, AFD1]. Viedol k tomu aj výskum publikovaný v [AFD7 a AFD8].*
- **Navrhnutá, implementovaná a otestovaná metóda pre nájdenie optimálnej súčinnosti slovníka fonetickej transkripcie a LTS pravidiel** (hodnotenie rozsahu slovníka a pravidiel fonetického prepisu, definovanie klasifikačného priestoru fonetickej transkripcie, frekvenčná analýza slov, tematicky optimalizované LTS pravidlá) - *Hľadanie optimálnej súčinnosti slovníka fonetickej transkripcie a LTS pravidiel je pomerne nová problematika, nebolo o nej doposiaľ publikované. - Výsledky a prínos týkajúci sa tohto bodu bol publikovaný v [AFC3, AFC4 a AFD5].*
- **Navrhnutá, implementovaná a otestovaná metóda G2P zarovnania** (spracovanie výnimiek G2P zarovnania, automatické nájdenie nových G2P pravidiel zarovnania) - *Rôzne metódy G2P zarovnania boli navrhnuté už pred touto prácou napr. [28] a [34]. Prínosom tejto práce je návrh metódy G2P zarovnania pre slovenčinu, vytvorenie súboru pravidiel a postup ako možno stanoviť doplnkové pravidlá pre výnimky. - Výsledky a prínos týkajúci sa tohto bodu bol publikovaný v [AFD6].*

Referencie

[1.] CERŇÁK, M. *Využitie objektívnych meraní kvality pri korpusovej syntéze reči.* Bratislava 2004. Dizertačná práca. FEI STU.

[2.] MICHALKO, O. *Difónový syntetizátor slovenskej reči.* Bratislava, 2007. Diplomová práca. FEI STU.

[3.] LEMMETTY, S. History and Development of Speech Synthesis. *Review of Speech Synthesis Technology* [online]. 1999 [cit. 2014-04-20]. Accessible: http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/chap2.html

[4.] TRAUNMÜLLER, H. Wolfgang von Kempelen's speaking machine and its successors. *Wolfgang von Kempelen on the Web* [online]. 2000 [cit. 2014-04-20]. Accessible: http://www2.ling.su.se/staff/hartmut/kemplne.htm

[5.] MANNELL, R. A Brief Historical Introduction to Speech Synthesis: A Macquarie Perspective. *Speech Synthesis* [online]. 2010 [cit. 2014-04-20]. Accessible: http://clas.mq.edu.au/speech/synthesis/history_synthesis/index.html

[6.] National Museum of American History. *Archives Center* [online]. 2013 [cit. 2014-05-15]. Dostupné z: http://americanhistory.si.edu/archives/speechsynthesis/ss_ibm5.htm

[7.] History. *HMM-based Speech Synthesis System (HTS)* [online]. 2013 [cit. 2014-05-15]. Dostupné z: http://hts.sp.nitech.ac.jp/?History

[8.] KONDELOVÁ, A. *Modifikácia prozódie v syntéze reči.* Bratislava, 2013. Dizertačná práca. FEI STU.

[9.] HUDEC, M. *Information technology applications in software compensation.* Banská Bystrica: Science and Research Institute of Matej Bel University, 2006. ISBN 80-8083-196-3.

[10.] PARALIČ, J., FURDÍK, K., TUTOKY, G., BEDNÁR, P., SARNOVSKÝ, M., BUTKA, P., BABIČ, F., *Dolovanie znalostí z textov.* Košice, 2010. TU KE. ISBN 978-80-89284-62-7

[11.] Fonetika. *Fonetika* [online]. 2013 [cit. 2014-05-15]. Dostupné z: http://sk.wikipedia.org/wiki/Fonetika

[12.] VASEK, M. *Transkripcia vstupného textu pre rečový syntetizátor.* Bratislava, 2009. Bakalárska práca. FEI STU.

[13.] TALAFOVÁ, R. *Syntéza reči v mobilnom telefóne*. Bratislava, 2007. Diplomová práca. FEI STU.

[14.] R. RASIPURAM, M. MAGIMAI.-DOSS. *Acoustic data-driven grapheme-to-phoneme conversion using KL-HMM*. ICASSP 2012 vol. 12, pp. 4841- 4844.

[15.] Extensible Markup Language (XML). *W3C* [online]. 2013 [cit. 2014-05-15]. Dostupné z: http://www.w3.org/XML/

[16.] BRAY, T. PAOLI, J. SPERBERG-MCQUEEN, C.M. MALER, E. YERGEAU, F. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* W3C Recommendation 26 November 2008.

[17.] OGBUJI, U. Principal Consultant, Fourthought, Inc., *Principles of XML design: When to use elements versus attributes.* Dostupné z: http://www.ibm.com/developerworks/xml/library/x-eleatt.html

[18.] HAJEK, M. *Python*. Košice, 2007. Univerzita Pavla Jozefa Šafárika. Dostupné z: http://s.ics.upjs.sk/~hajek/sps1/python/

[19.] StatSoft, Inc.. *Electronic Statistics Textbook.* Tulsa, OK: StatSoft. Dostupné z: http://www.statsoft.com/textbook/, 2010.

[20.] M.MAGIMAI.-DOSS, R. RASIPURAM, G. ARADILLA, AND H. BOURLARD. *Grapheme-based automatic speech recognition using KL-HMM*. Proc. of Interspeech, 2011.

[21.] G. ARADILLA, H. BOURLARD, AND M.MAGIMAI-DOSS. *Using KL-Based AcousticModels in a Large Vocabulary Recognition Task.* Proc. of Interspeech, 2008.

[22.] J. LAFFERTY, A. MCCALLUM, F. C. N. PEREIRA. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* Proceedings of the 18th International Conference on Machine Learning (ICML), pp. 282-289, 2001.

[23.] ŠEF, T.. *Slovenian text-to-speech system.* "Joief Stefan" Institute Jamova 39, SI-1000 Ljubljana, Slovenia, ISCAS 2000 - IEEE International Symposium on Circuits and Systems, 2000, Geneva, Switzerland.

[24.] DORFFNER, G., KOMMENDA, M., KUBIN, G., GRAPHON. *The Vienna speech synthesis system for arbitrary German text*. Institut für Nachrichtentechnik, Technische Universität Wien, Austria.

[25.] FARLEX, Inc., *The Free Dictionary*. 2012 Dostupné z: http://www.thefreedictionary.com/grapheme

[26.] D. BRAGA, L. COELHO, F. G. V. RESENDE JR. *A Rule-Based Grapheme-to-Phone Converter for TTS Systems in European Portuguese*. ITS, vol. 6, pp.238 - 333, 2006.

[27.] RICHMOND, K. CLARK, R.A.J. FITT, S.. *Robust LTS rules with the Combilex speech technology lexicon*. Centre for Speech Technology Research, University of Edinburgh, UK, 2009. Dostupné z: http://www.cstr.ed.ac.uk/downloads/publications/2009/IS090308.pdf.

[28.] M. BOHAC, J. MALEK, K. BLAVKA. *Iterative Grapheme-to-Phoneme Alignment for the Training of WFST-based Phonetic Conversion*. TSP, vol. 13, pp. 474 - 478, 2013.

[29.] ALTMANN, G., FENGXIANG, F. (EDS.). *Analyses of Script. Properties of Characters and Writing Systems.* Berlin, New York: de Gruyter, 2008.

[30.] M. CERŇAK, M. RUSKO, M. TRNKA, S. DARŽÁGÍN. *Data-driven versus knowledge-based approaches to orthoepic transcription in Slovak.* Inštitút informatiky, Slovenská akadémia vied.

[31.] HUANG, X. ACERO, A. HON, H.. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, New Jersey, 2001.

[32.] BREIMAN, L. FRIEDMAN, J.H. OLSHEN, R. AND STONE, C.J. 1984. *Classification and Regression Tree*. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific California. Dostupné z: http://support.spss.com/ProductsExt/SPSS/Documentation/Statistics/algorithms/14.0/TREE-CART.pdf

[33.] BEST, K.-H., ALTMANN, G.. *Some properties of graphemic systems.* Glottometrics 9, 29-39, 2005.

[34.] P. LEHNEN, S. HAHN, A. GUTA, H. NEY. *Incorporating alignments into conditional random fields for grapheme to phoneme conversion*. ICASSP, vol. 11, pp. 4916 - 4919, 2011.

[35.] HILL, T. AND LEWICKI, P. (2007). STATISTICS Methods and Applications. StatSoft, Tulsa,OK.

[36.] SIVANANDAM, S. N. *Introduction to Fuzzy Logic Using MATLAB.* Berlin, 2007.

[37.] VASEK, M., RYBÁROVÁ, R., ROZINAJ, G.. *Training database preparation for LTS rules training process with Wagon*. 53nd International Symposium ELMAR-2011, Croatia, September 2011.

[38.] BUK, S., MACUTEK, J., ROVENCHAK, A.. *Some properties of the Ukrainian writing system*. Glottometrics 16, 63-79, 2008

[39.] DROZD, I., VASEK, M.. *Modulárny syntetizátor*. Bratislava 2012, ŠVOČ. FEI STU.
[40.] BLACK, A.W., LENZO, K.A.. *Building Synthetic Voices.* Language Technologies Institute, Carnegie Mellon University, 2007. Lexicons. Dostupné z: http://festvox.org/bsv/x1469.html
[41.] 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, J. KUNZMANN. *Current Status of the IBM Trainable Speech Synthesis System.* Dostupné z: www.research.ibm.com/tts/pubs/esca01a.pdf
[42.] TÓTH, J., KONDELOVÁ, A., ROZINAJ, G.. *N-gram-Based Text Categorization.* PROCEEDINGS Redžúr 2013. Smolenice: STU, 2013. ISBN 978-80-227-3921-4, 23-26 p.
[43.] ŠIMKOVÁ M. *Slovenský jazyk v digitálnom veku.* Berlín: Springer, 2012, 30-36 p. ISBN 987-3-642-30370-8
[44.] 3GPP TS 23.060. *General Packet Radio Service (GPRS); Service description; Stage 2.* Rel-12, 12.4.0. Dostupné z: http://www.3gpp.org/DynaReport/23060.htm
[45.] RYBÁROVÁ, R.. *Metódy učenia pre syntézu reči*. Dizertačná práca, Katedra telekomunikácií FEI STU Bratislava, 2010
[46.] GONŠOR, J.. *Metodika opráv chybnej syntézy reči.* Diplomová práca, Katedra telekomunikácií, FEI STU, Bratislava, 2010
[47.] SOCHER, G., VISHWANATH, M., MENDHEKAR, A.. YAHOO! INC. *Intelligent text-to-speech synthesis* [patent]. USA. Grant, US 6446040 B1. Zapísaný: Sep 3, 2002. Dostupné z: http://www.google.com/patents/US6446040
[48.] OANCEA E., BADULESCU A., *Stressed Syllable Determination for Romanian Words within Speech Synthesis Applications*.

Publikačná činnosť autora

ADE Vedecké práce v zahraničných nekarentovaných časopisoch

ADE1 VASEK, MATÚŠ: *Introduction to Modular Smart Speech Synthesizer*. In: Elektrorevue [eletronický zdroj]. - ISSN 1213-1539. - Vol. 5. No. 1 (2014), online, p. 12-19

AFC Publikované príspevky na zahraničných vedeckých konferenciách

AFC1 KONDELOVÁ, ANNA - TÓTH, JÁN - VASEK, MATÚŠ - ROZINAJ, GREGOR: *Introduction to Speech Synthesis Management Tools.* In: IWSSIP 2012 : 19th International Conference on Systems, Signals & Image Processing. Vienna, Austria, April 11-13, 2012. - Vienna : Technical University, 2012. - ISBN 978-3-200-02588-2. - S. 643-646

AFC2 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Modular speech synthesizer in Python*. In: Redžúr 2012 : proceedings; 6th International Workshop on Multimedia and Signal Processing. April 11, 2012, Vienna, Austria. - Bratislava : Nakladateľstvo STU, 2012. - ISBN 978-80-227-3686-2. - S. 47-50

AFC3 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Optimization of Letter to Sound Rules Construction.* In: SPS 2013 : Signal Processing Symposium. Jachranka Village, Poland, June 5-7, 2013. - Piscataway : IEEE, 2013. - ISBN 978-1-4673-6318-1. - CD-ROM, [6] p.

AFC4 VASEK, MATÚŠ - ROZINAJ, GREGOR - RYBÁROVÁ, RENÁTA: Training *Database Preparation for LTS Rules Training Process with Wagon*. In: Proceedings ELMAR-2011 : 53rd International Symposium ELMAR-2011,14-16 September 2011, Zadar, Croatia. - Zadar : Croatian Society Electronics in Marine, 2011. - ISBN 978-953-7044-12-1. - S. 221-224

AFD Publikované príspevky na domácich vedeckých konferenciách

AFD1 DROZD, IVAN - VASEK, MATÚŠ: *Modulárny syntetizátor*. In: ŠVOČ 2012 [elektronický zdroj] : Zborník vybraných prác, Bratislava, 25. apríl 2012. - Bratislava : FEI STU, 2012. - ISBN 978-80-227-3697-8. - CD-ROM, s. 506-511

AFD2 ŠPILKA, MARIÁN - VASEK, MATÚŠ: *Kontextová analýza*. In: ŠVOČ 2013 [elektronický zdroj] : Zborník vybraných prác, Bratislava, 23. apríl 2013. - Bratislava : FEI STU, 2013. - ISBN 978-80-227-3909-2. - S. 384-389

AFD3 ŠPILKA, MARIÁN - VASEK, MATÚŠ: *Weighting of Words in Context Analysis*. In: Redžúr 2013 : proceedings; 7th International Workshop on Multimedia and Signal Processing; Smolenice, Slovakia, 1. Máj 2013. - Bratislava : Nakladateľstvo STU, 2013. - ISBN 978-80-227-3921-4. - S. 27-30

AFD4 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Komplexný systém pre fonetický prepis textu*. In: ŠVOČ 2011 : Študentská vedecká a odborná činnosť. Zborník vybraných prác. Bratislava, Slovak Republic, 4.5.2011. - Bratislava : STU v Bratislave FEI, 2011. - ISBN 978-80-227-3508-7. - S. 652-657

AFD5 VASEK, MATÚŠ - ROZINAJ, GREGOR: *LTS Letter-Specific Tree Rules*. In: Redžúr 2011 : proceedings; 5th International Workshop on Multimedia and Signal Processing. May 12, 2011, Bratislava, Slovak Republic. - Bratislava : Nakladateľstvo STU, 2011. - ISBN 978-80-227-3506-3. - S. 85-88

AFD6 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Relations between Graphemes and Phonemes in Slovak Language*. In: Redžúr 2013 : proceedings; 7th International Workshop on Multimedia and Signal Processing; Smolenice, Slovakia, 1. Máj 2013. - Bratislava : Nakladateľstvo STU, 2013. - ISBN 978-80-227-3921-4. - S. 19-22

AFD7 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Transcription of Text for the Speech Synthesizer*. In: Redžúr 2009 : proceedings; 3rd International Workshop on Speech and Signal Processing. Bratislava, Slovak Republic, 24.9.2009. - Bratislava : STU v Bratislave FEI, 2009. - ISBN 978-80-227-3137-9. - S. 43-46

AFD8 VASEK, MATÚŠ - ROZINAJ, GREGOR: *XML in Use for Phonetic Transcription*. In: Redžúr 2010 : 4th International Workshop on Speech and Signal Processing, Bratislava, Slovak Republic, May 14, 2010. - Bratislava : Nakladateľstvo STU, 2010. - ISBN 978-80-227-3296-3. - S. 87-90

AFD9 VASEK, MATÚŠ - ROZINAJ, GREGOR: *Transkripcia textu pre rečový syntetizátor*. In: ŠVOČ 2009 : Študentská vedecká a odborná činnosť. Zborník vybraných prác. Bratislava, Slovak Republic, 29.4.2009. - Bratislava : STU v Bratislave FEI, 2009.

Účasť autora na projektoch

1. Program na podporu mladých výskumníkov. Názov projektu: „*Optimalizácia LTS pravidiel fonetického prepisu textu”*, OPTLTS
2. HBB-Next - Next-Generation Hybrid Broadcast Broadband (http://www.hbb-next.eu) - Rozinaj Gregor (STU technical coordinator), Small or medium-scale focused research project (STREP) proposal ICT Call 7, FP7-ICT-2011-7 - 287848, (FEI No: 5828) (2011-2014)
3. IMUROSA - Integration of Multimedia Signal Processing Methods into Multimodal Interface and Network Applications (Integrácia metód spracovania MUltimediálnych signálov do multimodálneho ROzhrania a Sieťových Aplikácií) - Rozinaj Gregor, VEGA 1/0708/13, (FEI No: 1494/115722)(2013-2015)
4. MUFLON - Advanced Multimedia Services in the Environment of ICT Future Networks (Progresívne multimediálne služby v prostredí IKT sietí budúcnosti (future networks)) - Rozinaj Gregor, APVV-0258-12, (FEI No: AK17)(2013-2016)
5. ASIMD - Audio-Speech Interface for Mobile Devices - Rozinaj Gregor, DAAD, (2010-2011)
6. Algorithms and Methods of Multimedia Signal Processing for Human Machine Interface - Rozinaj Gregor, VEGA 1/0718/09, (2009-2011)
7. MENIS - Measurement and Processing of Low-level ECG Signals by Non-invasive Method - Rozinaj Gregor, APVV LPP-065-06, (2007-2010)
8. IQ Kiosk - Intelligent Terminal - Rozinaj Gregor, AV 4/0020/07, (2007-2009)

Resume

The work dealt with intelligent learning of the speech synthesizer pronunciation. Objective of this work was to comprehensively analyze the possibility of intelligent learning for speech synthesizer and design the concept of the modular synthesizer. The main objective of the work was the improvement of the phonetic transcription and proposal of methods for intelligent pronunciation learning. The work includes methods for user interaction that can be used also in the multi modal interface framework. This work researched the context dependent as well as context independent pronunciation relationships. Pronunciations were examined in terms of frequency, probability and character. The work proposes methods for transcription of new words, suffixes, prefixes and foreign words. Especially words with pronunciation out of scope of Slovak language rules. Concerned this, several methods have been proposed, implemented and tested: creation of corpus, over training of pronunciation rules and optimization of rules and dictionary size. Finally, this work dealt with the grapheme to phoneme alignment of trainings vectors for pronunciation rules training as well as the method for learning new alignment rules and their use in additional alignment of exceptions.

The work „Intelligent learning of the pronunciation of speech synthesizer“ delivered following original scientific results:

- **Context-sensitive text pre-processing for speech synthesizer method** (incl. contextual analysis of the text, creating a database for contextual text analysis, determining the likelihood of grammatical correctness of the words, determining the degree of word specificity, determining the importance of words in the thematic domain, creating scope of the themes of the text) - Expanded number of thematic domains from 3 to 55 and achieved better results and method for controlling transcription of numerals and abbreviations by identifying themes. Work brings **enhancement of the algorithm of phonetic transcription** by detecting suffixes, prefixes and foreign words.
- **Method for finding of LTS alternatives and their probabilities** (incl. preparation of text-to-SAMPA corpus, determining the intensity of phoneme letters according to corpus, determine the distribution grafemickej representation according to corpus).
- **Pronunciation correction method** (Using interactive SAMPA characters, arrangement of alternative pronunciations of words according to their probabilities, graphical interface for intelligent learning pronunciation, generating alternative pronunciations of words according to mask)
- **Design of modular speech synthesizer network architecture with support for learning** (architecture, interfaces, procedures, messages)
- **The method for finding the optimal synergy of phonetic transcription dictionary and LTS rules** (assessing the extent of the dictionary and phonetic transcription rules, defining the classification space of phonetic transcription, frequency analysis of words and thematically optimized LTS rules)
- **G2P alignment method** (exception handling G2P alignment, automatic finding new G2P rules alignment) - Proposal of G2P alignment methods for Slovak, a set of rules creation, procedures and additional rules for exceptions.