VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VLIV AKUSTIKY PROSTŘEDÍ NA ÚSPĚŠNOST ROZPOZNÁVAČE ŘEČI

IMPACT OF ENVIRONMENT ACOUSTICS ON SPEECH RECOGNITION ACCURACY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. JAKUB PALIESEK**
AUTHOR

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

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání diplomové práce

Student: **Paliesek Jakub, Bc.**
Program: Informační technologie
Obor: Bezpečnost informačních technologií
Název: **Vliv akustiky prostředí na úspěšnost rozpoznávače řeči**
Impact of Environment Acoustics on Speech Recognition Accuracy
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Nastudujte základy rozpoznávaní řeči a akustiky prostředí (reverberace, filtrace).
2. Nastudujte Kaldi toolkit. Zvolte datovou sadu pro vyhodnocení experimentů. Natrénujte základní (baseline) rozpoznávač řeči.
3. Navrhněte a realizujte experimenty, ve kterých využijete informace o akustice prostředí (impulzní odezva, velikost místnosti, atd.) ke zlepšení úspěšnosti rozpoznávače. Vyhodnoťte úspěšnost rozpoznávačů.
4. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
5. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Szőke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 28. dubna 2021

Abstrakt

Táto diplomová práca sa venuje vplyvom akustiky miestnosti na úspešnosť rozpoznávania reči. Na vyhodnotenie experimentov bol použitý rečový korpus LibriSpeech a databáza impulzných odoziev a šumu ReverbDB. Skúmané rozpoznávače reči boli založené na Kaldi recepte Mini LibriSpeech. Najskôr bolo zmerané, ako sa rozpoznávač dokáže naučiť rozpoznávať vo vybraných prostediach použitím rovnakých akustických podmienok pri trénovaní aj testovaní. Následne bolo experimentované s architektúrou systému s cieľom dosiahnuť čo najlepšiu robustnosť voči rôznym novým podmienkam za použitia metód pre adaptáciu na prostredie pomocou r-vektorov a i-vektorov. Bol ukázaný prínos nedávno predstavenej techniky r-vektorov aj pri použití augmentácie dát pomocou reálnych impulných odoziev.

Abstract

This diploma thesis deals with impact of room acoustics on automatic speech recognition (ASR) accuracy. Experiments were evaluated on speech corpus LibriSpeech and database of impulse responses and noise called ReverbDB. Used ASRs were based on Mini LibriSpeech recipe for Kaldi. First it was examined how well can ASR learn to transcribe in selected environments by using the same acoustic conditions during training and testing. Next, experiments were carried out with modifications of ASR architecture in order to achieve better robustness against new conditions by using methods for adapation to room acoustics – r-vectors and i-vectors. It was shown that recently proposed method of r-vectors is beneficial even when using real impulse responses for data augmentation.

Kľúčové slová

rozpoznávanie reči, akustika miestnosti, adaptácia, impulzná odozva

Keywords

speech recognition, room acoustics, adaptation, impulse response

Citácia

PALIESEK, Jakub. *Vliv akustiky prostředí na úspěšnost rozpoznávače řeči.* Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Vliv akustiky prostředí na úspěšnost rozpoznávače řeči

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Igora Szőkeho, Ph.D. Ďalšie materiály mi poskytli pán Ing. Martin Karafiát, Ph.D. a pán Ing. Martin Kocour. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Jakub Paliesek
16. mája 2021

Poďakovanie

Ďakujem pánovi Ing. Igorovi Szőkemu, Ph.D. za vedenie práce, pravidelné konzultácie a poskytnutie výpočtového prostredia pre experimenty.

Obsah

Kapitola 1

Úvod

Cieľom tejto práce je sledovať vplyv rôznych akustických parametrov na úspešnosť rozpoznávania reči a popritom využiť informáciu o akustike k prispôsobeniu zvoleného rozpoznávacieho systému a zvýšiť jeho úspešnosť. K prvému z cieľov sa dá pristupovať z viac hľadísk. Jedným z nich je práca s hotovým rozpoznávačom reči, ktorému môžeme predkladať nahrávky reči z rôznych prostredí, získané prepisy porovnávať s referenčným prepisom a chybovosť vyjadriť ako podiel chybne rozpoznaných slov (WER – word error rate). „Nahrávky z prostredí“ mohli vzniknúť nahratím skutočného rečníka, prehraním a zaznamenaním nahrávky v danej miestnosti, alebo simulovaním pomocou impulzných odoziev, pričom posledné dva uvedené spôsoby boli porovnané v bakalárskej práci [19], na ktorú táto diplomová práca nadväzuje.

Iným prístupom pre vyhodnotenie rozpoznávania reči je práca s nie hotovým rozpoznávačom, kedy môžeme ovplyvniť aj jeho architektúru a trénovací proces vrátane nahrávok použitých na natrénovanie. V tomto prípade nemusíme sledovať iba ako úspešne rozpoznávač rozpoznáva, ale aj ako sa *naučí rozpoznávať* vo vybraných podmienkach. Touto úrovňou problematiky sa zaoberá aj táto diplomová práca.

Zvolený rozpoznávací systém je najskôr opakovane trénovaný s rôznymi akustickými podmienkami a vyhodnocovaný na odpovedajúcich aj rozdielnych podmienkach. Tým sú zmerané limity, ktoré dokážeme so zvolenou architektúrou pre vybrané testovacie prostredia dosiahnuť. Zároveň je skúmané, ako vplýva charakter nahrávok (množstvo šumu, sila reverberácie) na pokles úspešnosti rozpoznávania zachytenej reči. Následne je experimentované s architektúrou rozpoznávača a výberom trénovacích dát s cieľom zvýšiť úspešnosť v praktickom využití, kedy by sa mal rozpoznávací systém adaptovať na rôzne podmienky a tým zvýšiť svoju robustnosť bez potreby pretrénovania na nové podmienky.

Trénovacie dáta pre skúmané systémy sú tvorené simuláciou akustiky miestnosti na pôvodne nekontaminovanom rečovom korpuse pomocou impulzných odoziev a primiešavaním vzoriek šumu. Tento spôsob tvorby dát je predpokladom pre modifikácie rozpoznávača, s ktorými sa v tejto práci experimentuje – dostupnosť pôvodného čistého datasetu pre vytvorenie kvalitnejšej supervízie pre trénovanie a znalosť impulznej odozvy využitej pri kontaminácii pre trénovanie podstystému pre adaptáciu na akustiku miestnosti.

V druhej kapitole sú predstavené javy prispievajúce k degradácii kvality rozpoznávania a metriky vyjadrujúce silu týchto javov. Tretia kapitola opisuje použité dátové sety s rečou, impulznými odozvami a nahrávkami šumu. Štvrtá kapitola predstavuje systém Kaldi použitý pre trénovanie a vyhodnocovanie rozpoznávačov a recept Mini LibriSpeech, z ktorého vychádzajú experimenty v tejto práci. Piata kapitola obsahuje zhrnutie implementácie umelej kontaminácie dát, nasledované popisom vykonaných experimentov a ich vyhodnotením, zoskupené do podsekcií.

Kapitola 2

Kontaminácia rečového signálu

V nahrávkach zhotovených mikrofónmi rôzne umiestnenými v miestnostiach sa stretávame s dvoma prvkami, ktoré spôsobujú, že rozpoznávanie reči z nich je zložitejšie oproti nahrávkam zhotoveným s mikrofónom umiestneným priamo pred rečníkom v krátkej vzdialenosti – reverberáciou a aditívnym šumom.

Reverberácia je fenomén, ktorý vzniká v dôsledku odrazu zvuku od rôzne vzdialených povrchov v miestnosti (napríklad stien, nábytku). Tieto odrazy vníma prijímač ako rôzne oneskorené kópie pôvodného zvuku [37]. Zmeny zvuku medzi zdrojom a prijímačom v dôsledku reverberácie je možné popísať *impulznou odozvou* (IR – impulse response, RIR – room impulse response). Impulzná odozva závisí od vzájomnej polohy zdroja a prijímača.

Druhým rušivým prvkom je šum, ktorý je v nahrávke prítomný z rôznych dôvodov. Spôsobovať ho môže prítomnosť prístrojov, napríklad klimatizácia, prehrávanie hudby, ale aj samotná nahrávacia technika.

Matematicky môžeme vyjadriť vzťah medzi pôvodným zvukom, zaznamenanou nahrávkou, aditívnym šumom a impulznou odozvou označenými $x(t)$, $y(t)$, $n(t)$, resp. $h(t)$ v rovnici 2.1. [37]. Symbol $*$ predstavuje operáciu lineárnej konvolúcie.

$$y(t) = x(t) * h(t) + n(t) \tag{2.1}$$

Zložku šumu $n(t)$ je možné ďalej rozpísať na súčet izotropného šumu (nezávislého na smere) a sumy šumov z bodových zdrojov konvoluovaných s odpovedajúcimi impulznými odozvami [13]. Impulznú odozvu $h(t)$ môžeme podľa [4] rozdeliť na 3 zložky:

$$h(t) = h_d(t) + h_e(t) + h_l(t) \tag{2.2}$$

V rovnici 2.2 predstavuje symbol $h_d(t)$ prvotný impulz – najkratšiu cestu medzi zdrojom a prijímačom, $h_e(t)$ sú tzv. skoré odrazy (v literatúre označované early reflections) prijaté v rádoch prvých desiatok milisekúnd od prvého impulzu a $h_l(t)$ sú neskoré odrazy (late reflections), ktoré poslucháč vníma ako ozvenu a sú škodlivé z hľadiska schopnosti porozumenia reči pre človeka [15].

2.1 Zvyšovanie úspešnosti rozpoznávania kontaminovanej reči

Za účelom zvýšenia úspešnosti rozpoznávania reči v akusticky náročnejších prostrediach sa vrámci rozpoznávačov realizujú rôzne opatrenia. Používané sú metódy zamerané na vylepšenie zvuku pred klasifikáciou akustickým modelom. Patrí medzi ne napríklad metóda lineárneho filtrovania, ktorá sa snaží v časovej doméne odstrániť reverberáciu. Iným prístupom je

vylepšovanie spektra, kde sa odpovedajúce metódy snažia porušený modul spektra signálu priblížiť čistej reči, nehľadiac pritom na fázu spektra. Existujú aj metódy, ktoré sa snažia o vylepšovanie vektorov príznakov po tom, čo boli extrahované [37]. Nevýhodou týchto prístupov je nevyhnutné zavedenie odhadovacích chýb [21].

Alternatívou k technikám vylepšovania signálu je použitie kontaminovaných dát pre natrénovanie akustického modelu v kombinácii s vhodnou architektúrou rozpoznávacieho systému. Na túto tému existuje viacero prác, napríklad [13, 28, 21] a ďalšie. Trénovaním rozpoznávačov na kontaminovaných dátach sa zaoberá aj táto práca.

Pokiaľ máme k dispozícii korpus nahrávok ľudskej reči vo vysokej kvalite (v literatúre často označované ako close-talk speech), môžeme na princípe zhrnutom rovnicou 2.1 umelo vytvoriť verziu tohto korpusu s akustickými vlastnosťami prostredia, pre ktoré chceme rozpoznávač adaptovať už v čase trénovania alebo vytvoriť dataset pozostávajúci z nahrávok s rôznymi akustickými vlastnosťami pre natrénovanie robustného rozpoznávača voči rozličným podmienkam. Na získanie alebo vygenerovanie impulznej odozvy miestnosti existujú rôzne metódy a šum je možné jednoducho zaznamenať.

2.2 Získavanie impulznej odozvy

Impulznú odozvu je možné vygenerovať pomocou počítača z informácií ako rozmery miestnosti, poloha zdroja a prijímača, prípadne ich natočenie. Známou metódou je ISM (image source model) [1], ktorá pracuje s predpokladom odrazivosti zvuku od stien miestnosti podobne ako u zrkadla. Nová metóda opísaná v [35] sa snaží kompenzovať rozdiely v spektre medzi syntetickými a reálnymi impulznými odozvami pomocou filtrov a vykazuje zlepšenia v úspešnosti rozpoznávania porovnateľné s reálnymi IR.

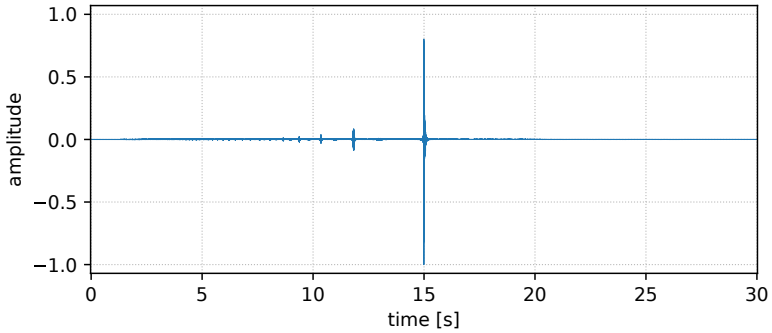
Ako vhodnejšie v oblasti rozpoznávania reči sa ukazujú reálne impulzné odozvy. Najznámejšie metódy pre ich estimáciu sú predstavené v [33]. Spoločnou vlastnosťou týchto techník je prehratie a záznam excitačného signálu v danom prostredí a následná dekonvolúcia impulznej odozvy z tejto nahrávky.

Dve z metód pre meranie reálnej impulznej odozvy, ESS (exponential sine sweep) a MLS (maximum length sequence) boli podrobne porovnané v mojej bakalárskej práci [19] na úlohe vyhodnotenia vopred natrénovaného rozpoznávača reči pri rozpoznávaní dát kontaminovaných IR získanými týmito dvomi technikami. V rozličných akustických podmienkach zahŕňajúcich rôzne vzdialenosti medzi zdrojom a prijímačom, natočenia reproduktora a (ne)prítomnosť prekážok po ceste bola nameraná veľmi podobná úspešnosť rozpoznávania pre odpovedajúce konfigurácie.

2.2.1 Exponential sine sweep

Metóda ESS okrem podobnej úspešnosti kontaminácie reči vykazuje ďalšie vlastnosti, vďaka ktorým je vhodnejšia pre získanie reálnych IR oproti MLS. Jej výhodou je, že dokáže v časovej doméne separovať lineárnu IR od harmonických skreslení spôsobených nelinearitami v prostredí [9]. Okrem toho, ako bolo overené v [19] je robustná voči nezrovnalostiam v synchronizácii hodín prehrávača a nahrávacej techniky a tiež prítomnosti praskania v nahrávke excitačného signálu.

Excitačný signál je v tvare sínusoidy s exponenciálne sa zvyšujúcou frekvenciou. Parametre pre jeho generovanie sú dĺžka a počiatočná a koncová frekvencia. Lineárnou konvolúciou nahrávky s časovo prevrátenou verziou excitačného signálu s určitou kompenzáciou spektra získame signál, ktorý je dvakrát dlhší ako excitačný signál. Želaná impulzná odozva začína



Obr. 2.1: Extrahovaný signál metódou ESS. Impulzná odozva sa začína v polovici časovej osi, pred ňou sa nachádzajú harmonické skreslenia. Prevzaté z [19].

v polovici časovej osi, pred ňou sa nachádzajú separované harmonické skreslenia. Príklad extrahovaného signálu je na Obr. 2.1.

Dostatočná dĺžka excitačného signálu je dôležitá pre správnu separáciu. Príliš krátky signál spôsobí prekrývanie lineárnej IR so skresleniami. V [19] bola otestovaná kontaminácia týmito IR dekonvoluovanými z ESS dĺžok od 15 do 240 sekúnd, pričom boli zistené rozdiely v chybovosti na úrovni štatistickej chyby. Ďalej pracujeme s databázou obsahujúcou impulzné odozvy extrahované z ESS dĺžky 15 sekúnd.

2.3 Akustické vlastnosti impulznej odozvy

V tejto sekcii predstavíme niektoré číselné metriky vypočítateľné z impulznej odozvy, ktoré môžu ovplyvňovať úspešnosť rozpoznávania reči.

2.3.1 Doba dozvuku

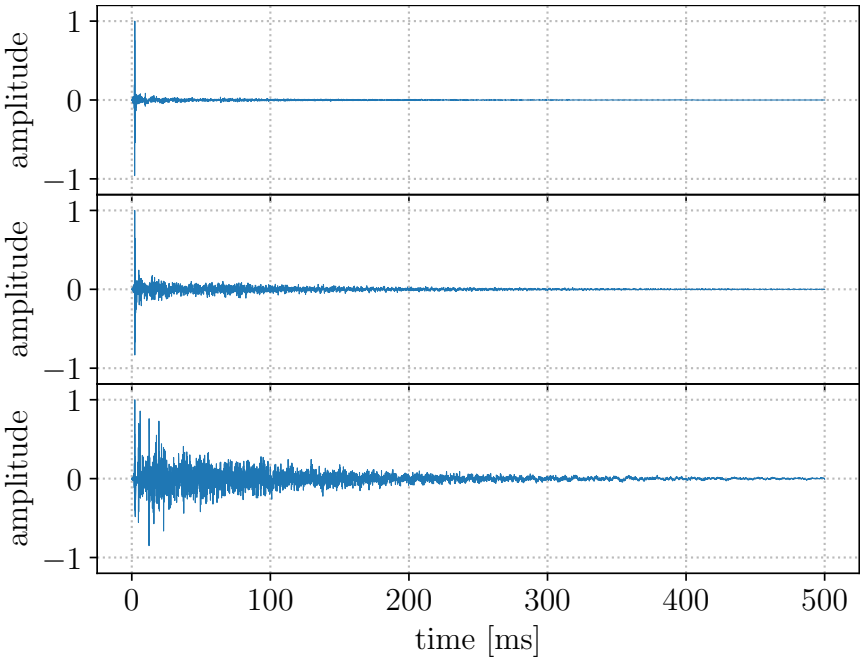
Metrika nazývaná doba dozvuku (reverberation time) je známym a často používaným parametrom v oblasti akustiky miestností. Vyjadruje čas, za ktorý klesne hustota zvukovej energie o 60 dB po vypnutí zdroja zvuku alebo po vybudení impulzom. V literatúre je označovaná ako RT_{60}. Závisí na rozmeroch miestnosti, materiáli stien, podlahy a stropov a tiež množstve nábytku. Existujú postupy pre jej výpočet pre danú miestnosť zo znalosti týchto údajov [15]. Namerané hodnoty sú približne konštantné v rámci jednej miestnosti [18]. Pre odhad z impulznej odozvy existujú rôzne implementácie porovnané v [5]. Pre nedostatok dynamického rozsahu sa zvykne odhadovať z analogicky definovaných parametrov RT_{20} alebo RT_{30} násobených číslom 3, resp. 2.

2.3.2 Pomer priamej cesty a reverberácie

Tento akustický parameter predstavuje pomer medzi zvukovou energiou v nahrávke v dôsledku priamej cesty medzi zdrojom a prijímačom a energiou spôsobenou reverberáciou. Udáva sa v dB. Pre odhad z impulznej odozvy je možné nájsť vzorku s najvyššou amplitúdou a okolité vzorky v časovom okne dĺžky 2.5 ms ako v [8]. Výpočet môžeme vyjadriť vzorcom 2.3, kde t_d je čas, v ktorom sa nachádza vzorka s najvyššou amplitúdou v impulznej odozve

$h(t)$. Príklady impulzných odoziev s rôznymi hodnotami DRR sú na Obr. 2.2.

$$\mathrm{DRR} = 10 \log_{10} \frac{\int_{t_d-1.25}^{t_d+1.25} h^2(t)\,\mathrm{d}t}{\int_{t_d+1.25}^{\infty} h^2(t)\,\mathrm{d}t} \tag{2.3}$$



Obr. 2.2: Zhora: impulzná odozva pomerom energií priamej cesty a odrazov (DRR) s hodnotami 10 dB, 0 dB resp. −10 dB. Jedná sa o reálne impulzné odozvy z databáze ReverbDB [34].

2.3.3 Pomer skorých a neskorých odrazov

Pomer skorých a neskorých odrazov vyjadruje podobne ako DRR pomer energií dvoch časových úsekov vrámci impulznej odozvy. Pre podobný účel sa v literatúre pre tento údaj používajú rôzne označenia, napríklad C_{80} v [15] ako *clarity* – jasnosť. Číslo v dolnom indexe vyjadruje hranicu medzi úsekmi skorých a neskorých odrazov, v tomto prípade 80 ms od začiatku impulznej odozvy. V [4] je táto hranica experimentálne stanovaná na 110 ms a rovnaká metrika je tam označovaná ako ELR_T (early-to-late reverberation ratio). V zmysle vzorca 2.2 sa jedná o výpočet:

$$\mathrm{ELR} = 10 \log_{10} \frac{\int h_d^2(t) + h_e^2(t)\,\mathrm{d}t}{\int h_l^2(t)\,\mathrm{d}t} \tag{2.4}$$

2.4 Prehľad súvisiacich prác

Rozpoznávacie systémy odovzdané v rámci výzvy ASpIRE (Automatic speech recognition in reverberant enviroments) boli v [16] použité na analýzu vplyvu vzájomnej pozície mikrofónu

a rečníka na úspešnosť rozpoznávania. Bolo zistené, že metrika tvorená súčtom útlmov spôsobených vzdialenosťou a odklonom pozície mikrofónu od smeru rečníka je silnejšie korelovaná s chybovosťou ako jednoduchšia metrika založená na samotnej vzdialenosti.

Článok [4] sa zaoberá vplyvom metriky ELR (2.4) na úspešnosť rozpoznávacieho systému pre sekvencie vyslovených číslic. Pracuje sa tu s hranicou medzi skorými a neskorými odrazmi experimentálne stanovenou na 110 ms. Bola ukázaná silná korelácia ELR s úspešnosťou rozpoznávača trénovaného na čistých dátach. Okrem toho sa ukázalo, že systém trénovaný na jednej IR s určitou hodnotou ELR je efektívny pri rozpoznávaní dát vytvorených pomocou IR z iných prostredí, ale s podobnou hodnotou ELR. Iná práca [30] experimentuje s umelým znižovaním energie neskorých častí reálnych IR a následnou konvolúciou testovacích dát, čím simuluje dereverberáciu. Hranica stanovená na 50 ms sa ukázala ako optimum, za ktorým sa postupne znižovala úspešnosť rozpoznávania systémom trénovaným na nekontaminovaných dátach.

Štúdia [13] obsahuje formuláciu komplexného algoritmu pre augmentáciu trénovacích dát konvolúciou s IR a primiešavaním šumu, ktorý rozdeľuje podľa toho, či pochádza z bodového zdroja alebo je izotropný. Bodový šum je ďalej primiešavaný po konvolúcii s ďalšou IR. Experimenty v tomto článku ukazujú výrazné zlepšenie úspešnosti rozpoznávania pri použití augmentovaných dát (relatívne zlepšenie 31.4 %). Okrem toho sa ukázalo ako výhodné použitie kombinácie simulovaných a reálnych IR oproti prístupu s použitím iba reálnych IR (1.7 % relatívne zlepšenie). Tiež bolo prezentované mierne zlepšenie rozpoznávania pri trénovaní vrátane šumu z bodových zdrojov oproti samotnému izotropnému šumu (3.3 % relatívne zlepšenie). S niektorými z týchto techník budeme experimentovať aj v tejto práci.

Práce [12, 3, 28] predstavujú a vyhodnocujú rôzne metódy pre učenie rozpoznávačov zameraných na dáta kontaminované reverberáciou, šumom alebo oboch javov naraz – adaptácia na prostredie pomocou r-vektorov (podobne ako adaptácia na rečníka pomocou i-vektorov) s relatívnym zlepšením do 14 %, predtrénovanie neurónovej siete pomocou čistej verzie datasetu a postupné pridávanie kontaminovaných dát v neskorších fázach alebo využitie zarovnaní čistej verzie datasetu na fonémy ako učiteľa pre trénovanie neurónovej siete s kontaminovanými dátami (relatívne zlepšenie do 12 %).

Kapitola 3

Dátové sety

V tejto kapitole predstavíme dva datasety, ktoré sa ďalej v práci používajú – rečový korpus LibriSpeech a databázu impulzných odoziev a šumu ReverbDB.

3.1 LibriSpeech

V tejto práci používame voľne dostupný korpus LibriSpeech v anglickom jazyku ako základ pre trénovanie aj vyhodnocovanie rozpoznávacích systémov. LibriSpeech je založený na projekte LibriVox[1], databáze voľne dostupných audiokníh, ktorých texty sú väčšinou získané z projektu Gutenberg[2], databáze taktiež voľne dostupných e-kníh. Celková dĺžka nahrávok je 1 000 hodín [20].

Každý audio súbor v korpuse má nanajvýš 35 sekúnd. Tieto úryvky vznikli pre trénovaciu časť delením v úsekoch ticha dlhších ako 300 ms. V testovacích častiach okrem dostatočne dlhého ticha muselo dané miesto byť zároveň koncom vety. Adresárová štruktúra korpusu má dve úrovne: prvá delí nahrávky podľa rečníka, ktorého adresár ďalej obsahuje jeden alebo viac podadresárov odpovedajúcich kapitolám kníh. Kapitoly nemusia byť kompletné, pretože autori korpusu kládli dôraz na odfiltrovanie častí s vysokou pravdepodobnosťou nesúladu s prepisom [20].

Okrem audio súborov v bezstratovom formáte FLAC a odpovedajúcich prepisov sú v balíkoch dostupné metadáta o rečníkoch v súbore `SPEAKERS.TXT`, ktoré sa používajú v prípravnom skripte na vytvorenie súboru s mapovaním medzi identifikátorom nahrávky a pohlavím rečníka:

```
;ID  |SEX| SUBSET           |MINUTES| NAME
14   | F | train-clean-360  | 25.03 | Kristin LeMoine
16   | F | train-clean-360  | 25.11 | Alys AtteWater
17   | M | train-clean-360  | 25.04 | Gord Mackenzie
...
```

Bližšie informácie o knihách a kapitolách nájdeme v súbore `CHAPTERS.TXT` (skrátené):

```
;ID    |READER| SUBSET           | PROJ.|BOOK ID| CH. TITLE | PROJECT TITLE
1      | 110  | train-other-500  | 53   | 1023  | In Chancery | Bleak House
2      | 110  | train-other-500  | 53   | 1023  | In Fashion | Bleak House
159    | 4174 | train-other-500  | 68   | 2184  | Letter XXV | Unbeaten Tracks
```

[1] https://librivox.org/
[2] http://www.gutenberg.org

Korpus LibriSpeech je rozdelený na niekoľko rôzne dlhých častí určených na trénovanie alebo testovanie. Každá časť obsahuje množinu rečníkov neprekrývajúcu sa s inou časťou. Kompletné delenie je v tabuľke 3.1.

časť	dlžka (hodiny)	minút na rečníka	ženy	muži	spolu rečníkov
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Tabuľka 3.1: Delenie korpusu LibriSpeech na samostatne stiahnuteľné časti. V každej časti je približne rovnaké zastúpenie pohlaví rečníkov. Prevzaté z [20]

Kategórie *clean* a *other* boli vytvorené autormi na základe úspešnosti rozpoznávania nahrávok na rozpoznávači s akustickým modelom trénovaným na inom korpuse, WSJ. Nahrávky s lepšou úspešnosťou sú v *clean* verziách, ostatné nahrávky boli zaradené do častí s príponou *other* [20].

Pre účely experimentov s trénovaním pracujeme najmä s časťou *train-clean-100* a pre vyhodnocovanie s *test-clean*. *Clean* verzie boli vybrané, pretože chceme vrámci vyhodnocovania minimalizovať vplyvy spôsobené samotným korpusom. Naopak, sústredíme sa na degradáciu úspešnosti rozpoznávania spôsobenú vytváraním kontaminovaných verzií týchto častí. Pre trénovacie účely bola vybratá kratšia časť predovšetkým kvôli časovej úspore.

K tomuto korpusu sú dostupné 4 verzie natrénovaného jazykového modelu (LM – language model), jeden 4-gramový a 3 rôzne veľké 3-gramové vo formáte ARPA[3]. Základom pre ich tvorbu bolo 14 500 kníh z projektu Gutenberg z pôvodného počtu 22 000, kde niektoré boli vyradené, aby sa predišlo zhode s časťami rečového korpusu *dev* a *test*, vrátane citácií a možnej prítomnosti rovnakých rozprávok v rôznych knihách. V knihách, ktoré prešli filtrovaním bolo spolu 803 miliónov tokenov a 900 000 unikátnych slov. Najčastejšie sa vyskytujúcich 200 000 slov bolo aj s ich výslovnosťami zaradených do slovníka, ktorý je možné stiahnuť z rovnakej stránky ako ARPA jazykové modely [20].

V toolkite Kaldi, ktorý používame aj v tejto práci je zároveň vydaný recept LibriSpeech, s ktorým je možné reprodukovať výsledky prezentované v [20]. Pre *test-clean* dosiahli autori s akustickým modelom založenom na hlbokej neurónovej sieti (DNN) trénovanej na všetkých 960 hodinách trénovacích dát WER na úrovni 8.02, 7.21, 5.74 a 5.51 pre trigramové LM (od najmenšieho po najvačší), resp. 4-gramový LM. V tejto práci pracujeme so zjednodušenou verziou Mini LibriSpeech popísanou v sekcii 4.2. K tomuto receptu sú existujú aj podmnožiny rečových dát[4] z LibriSpeech, *train-clean-5* a *dev-clean-2*, nebudeme s nimi však pracovať a budú nahradené časťami z pôvodného LibriSpeech.

[3] http://www.openslr.org/11/
[4] http://www.openslr.org/31/

3.2 BUT ReverbDB

ReverbDB je voľne dostupná databáza impulzných odoziev, retransmitov[5] čistých nahrávok ľudskej reči a tichých nahrávok obsahujúcich šum na pozadí. Vznikla na fakulte FIT VUT [34]. Materiál z tejto databáze je vhodný na tvorbu kontaminovaných dát ako bolo predstavené v kapitole 2.

Databáza obsahuje impulzné odozvy zmerané metódou ESS v celkovo 9 miestnostiach rôznych typov: kancelária, meetingová miestnosť, prednášková aula a schody. V každej miestnosti autori najskôr rozmiestnili 31 mikrofónov pripojených k nahrávaciemu zariadeniu, ktoré zaznamenávalo zvuk zo všetkých mikrofónov naraz. Postupne s rôznymi pozíciami štúdiového monitora boli v prostredí prehrávané excitačné signály pre extrakciu IR. Pre podmnožinu pozícií reproduktora a miestností boli zároveň prehrávané čisté (close-talk) verzie rečových korpusov. Medzi reálnymi retransmitmi je dostupná aj časť korpusu LibriSpeech *test-clean*. Výhodou tohto datasetu je, že ku každej pozícii mikrofón-reproduktor obsahuje podrobné metadáta zahŕňajúce údaje ako kartézske a polárne súradnice a orientácia oboch zariadení, vzdialenosť, či prítomnosť prekážok medzi reproduktorom a mikrofónom.

Okrem tradičných umiestnení mikrofónov pred reproduktorom v malej vzdialenosti môžeme pracovať aj s umiestneniami viac ako 10 metrov od zdroja. Nechýbajú pozície so mikrofónmi skrytými vo vnútri uzavretého nábytku, takmer uzavretého odpadkového koša alebo v kvetináči. Objem miestností sa pohybuje od 40 do 2 000 m^3. Informácie o miestnostiach a dostupnosti retransmitov sú zhrnuté v tabuľke 3.2.

Názov	Objem [m^3]	Typ miestnosti	RIR	LibriSpeech retransmit
Q301	192	Kancelária	Áno	Áno
L207	98	Kancelária	Áno	Áno
L212	107	Kancelária	Áno	Áno
L227	229	Schody	Áno	Áno
R112	∼40	Hotelová izba	Áno	Nie
CR2	1033	Konferenčná miestnosť	Áno	Nie
E112	∼900	Prednášková aula	Áno	Nie
D105	∼2000	Prednášková aula	Áno	Áno
C236	102	Meetingová miestnosť	Áno	Nie

Tabuľka 3.2: Prehľad miestností dostupných v ReverbDB. Prevzaté z [34], upravené.

Reálny retransmit LibriSpeech *test-clean* budeme používať na vyhodnotenie úspešnosti rozpoznávania. Dôvodom je, že spôsob tvorby týchto dát je najbližšie reálnym podmienkam pre rozpoznávače reči, teda rečník fyzicky prítomný v miestnosti. V našom prípade je rečník nahradený reproduktorom schopným vernej reprodukcie na štúdiovej úrovni. Hlavné experimenty v tejto práci sa týkajú miestností L207 a D105. Dôvodom je dostupnosť testovacích dát a kontrast vo veľkosti týchto miestností.

3.2.1 Prednášková miestnosť D105

Ako reprezentant veľkých miestností bola pre experimenty vybraná prednášková miestnosť D105. Predná a bočné steny sú betónové, na bočných stenách sa nachádzajú okná. Zadná

[5] retransmit – zvuk prehraný a znovu zaznamenaný v danej miestnosti obsahujúc akustické vlastnosti tohto prostredia

stena a strop sú zo sadrokartónu. Na podlahe sú drevené parkety. Uvádzané sú rozmery: výška 6.9 m, šírka 22.8 m a hĺbka 17.2 m, ale celkový uvedený objem je menší (približne 2000 m^3) vďaka poschodovému hľadisku. Na Obr. 3.1 je fotografia miestnosti.

V D105 bol zo 7 rôznych pozícií reproduktora zaznamenaný reálny retransmit LibriSpeech iba pre pozíciu v prednej časti hľadiska smerovaného približne k strednej časti katedry. V datasete je identifikovaná číslom 7. Ak nebude uvedené inak, v experimentoch sa pracuje práve s touto konfiguráciou.

Obr. 3.1: Panoramatická snímka miestnosti D105 s viditeľným rozmiestnením nahrávacej techniky pri pohľade od vstupných dverí. Vpredu miestnosti sú mimo snímky v oboch rohoch umiestnené balkóny.

3.2.2 Kancelária L207

Akustiku malej miestnosti reprezentuje kancelária L207. Materiál stien je podľa metadát betón a sadrokartón. Strop je betónový a na podlahe je linoleum. Vo vnútri je pomerne veľké množstvo nábytku a iných predmetov z rôznych materiálov. Rozmery sú: výška 3.1 m, šírka 6.9 m a hĺbka 3.1 m. Fotografia je na Obr. 3.2.

Z celkových 6 pozícií reproduktora boli pre identifikátory č. 1, 2 a 6 zaznamenané aj reálne retransmity. Číslo 6 predstavuje polohu na kraji miestnosti (vidno na obrázku), čo umožňuje skúmať vplyv rôznej vzdialenosti mikrofónov umiestnených priamo pred reproduktorom (t. j. s minimálnou odchýlkou ich pozície od osi membrány reproduktora). Preto budeme v práci ďalej pracovať primárne s touto polohou.

Obr. 3.2: Panoramatický záber miestnosti L207 s viditeľným rozmiestnením nahrávacej techniky pri pohľade od vstupných dverí. Na snímke je vidno reproduktor umiestnený na poličke vľavo.

Kapitola 4

Rozpoznávanie reči

V tejto kapitole bližšie rozoberieme princípy a technológie použité v experimentálnej časti, v ktorej trénujeme rozpoznávače reči a vyhodnocujeme úspešnosť dekódovania.

4.1 Kaldi toolkit

Kaldi[1] je open-source súbor nástrojov používaný na výskum rozpoznávania reči. Je založený na konečných stavových prevodníkoch. Z toho dôvodu závisí na knižnici OpenFST[2]. Druhou závislosťou je knižnica na lineárnu algebru, kde je možné si pri kompilácii toolkitu vybrať z viacero alternatív (Intel MKL, OpenBLAS, CLAPACK), ktoré implementujú rozhrania BLAS a LAPACK. Hlavný účel tohto toolkitu je akustické modelovanie. Niektoré výpočty, ako napríklad trénovanie neurónových sietí, je možné akcelerovať na GPU, pokiaľ je toolkit kompilovaný s podporou Nvidia CUDA. [26]

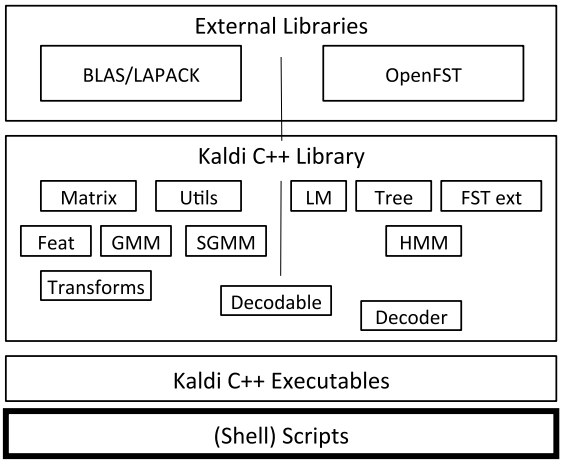
Skompilovaný toolkit obsahuje niekoľko programov napísaných v C++ so špecifickou funkcionalitou a typicky nízkym počtom argumentov, napríklad pre extrakciu príznakov, lineárne transformácie, spájanie vektorov a podobne. Vstupy a výstupy týchto programov sú prepájané v rámci Shell skriptov, čím sa tvorí rozpoznávací systém. Samotné dáta prúdia medzi programami buď pomocou rúr (pipes) alebo pomocou medzivýsledkov ukladaných na disk. Repozitár obsahuje adresár `egs`, kde sú umiestnené publikované príklady kompletných rozpoznávacích systémov, nazývané recepty. Architektúra toolkitu je znázornená na 4.1.

Kaldi recept predpokladá na vstupe určitý formát dát a prítomnosť súborov, s ktorými bude pracovať, napríklad zoznam nahrávok určených na trénovanie a rôzne mapovania (rečníka na nahrávku, nahrávku na prepis a podobne). Ku niektorým dostupným korpusom sú zároveň vydávané odpovedajúce recepty pre Kaldi, ktoré obsahujú skripty pre prípravu dát, čo je využité ako základ aj v tejto práci. Pokiaľ vytvoríme inú verziu vydaného korpusu so zachovaním rovnakej štruktúry, môžeme použiť existujúce prípravné skripty.

Spúšťanie jednotlivých výpočtov vrámci Kaldi receptov je abstrahované pomocou Perl skriptov `run.pl` a `queue.pl`. To umožňuje kroky spustiť na lokálnom počítači, alebo zaradiť ako úlohy do fronty vo výpočtovom klastri riadenom Oracle Grid Engine. Pri oboch prístupoch je možné úlohy rozdeliť na niekoľko častí a spúšťať ich paralelne. Okrem spomenutých dvoch skriptov existujú aj `slurm.pl` a `ssh.pl` [25].

[1] http://kaldi-asr.org/

[2] http://www.openfst.org/twiki/bin/view/FST/WebHome



Obr. 4.1: Architektúra rozpoznávacieho systému založeného na Kaldi. Na zvýraznenej úrovni sú vrámci tejto práce robené určité úpravy. Prevzaté z [26], upravené.

4.2 Recept Mini LibriSpeech

Táto práca vychádza z publikovaného receptu `mini_librispeech` na opakované trénovanie akustického modelu a následné vyhodnotenie úspešnosti rozpoznávania dát. Účelom tejto sekcie je predstaviť, čo sa deje v pôvodnej verzii tohto receptu dostupnej v Kaldi. Recept je ďalej postupne modifikovaný pre účely experimentov a tieto zmeny sú pre prehľadnosť uvádzané spolu s výsledkami experimentov v Kapitole 5. Bloková schéma systému je zobrazená na Obr. 4.2.

Adresár s receptom pred jeho spustením obsahuje:

- `run.sh` – spúšťací skript receptu, ktorý stiahne potrebné dáta a jazykový model, extrahuje príznaky, natrénuje modely a dekóduje testovacie dáta,
- `cmd.sh` – obsahuje nastavenia pre spúšťanie jednotlivých krokov v recepte. Sú tu odkazované spomínané skripty `run.pl` alebo `queue.pl`, od ktorých závisí, kde a ako budú programy spustené,
- `path.sh` – modifikuje premennú prostredia `PATH`, kde je nastavená cesta k spustiteľným súborom, ktoré budú spustené na výpočtových uzloch,
- `RESULTS` – skript, ktorý vyhodnotí výsledky dekódovania testovacích setov,
- `utils` – adresár spoločný s inými receptami, obsahuje pomocné skripty spúšťané aj počas behu `run.sh`,
- `steps` – adresár podobne ako `utils` spoločný s inými receptami, obsahuje podkroky spúšťané v `run.sh`,
- `local` – adresár so skriptami špecifickými pre tento recept,
- `conf` – obsahuje konfiguračné súbory obsahujúce prepínače, s ktorými sú spúšťané skripty z ostatných adresárov.

4.2.1 Príprava dát pre trénovanie

V pôvodnej verzii recept vždy stiahne a použije pre trénovanie `test-clean-5` a na vyhodnotenie `dev-clean-2`, podmnožiny dát z pôvodného LibriSpeech datasetu. Pred extrakciou príznakov a procesom trénovania skript `local/data_prep.sh`, špecifický pre formát dát tohto korpusu, pripraví niekoľko štandardných súborov, s ktorými nástroje Kaldi ďalej pracujú, napríklad (zoznam nie je kompletný):

- `text` – pre každý preslov jeho transkripcia, príklad (výstup dekódovania nahrávok rozpoznávačom je tiež v tomto formáte):

  ```
  103-1240-0000 CHAPTER ONE MISSUS RACHEL LYNDE IS SURPRISED
  103-1240-0001 THAT HAD ITS SOURCE AWAY BACK IN THE WOODS
  103-1240-0002 FOR NOT EVEN A BROOK COULD RUN PAST MISSUS RACHEL
  ```

- `utt2spk` – mapovanie medzi rečníkmi a preslovmi. Skript z adresára `utils` vytvorí aj súbor pre opačné mapovanie `spk2utt`. Tieto mapovania používajú programy, ktoré pracujú s adaptáciou na rečníka. Príklad:

  ```
  103-1240-0000 103-1240
  103-1240-0001 103-1240
  103-1240-0002 103-1240
  ...
  ```

- `spk2gender` – mapovanie rečníka na pohlavie. Príklad:

  ```
  103-1241 f
  1034-121119 m
  1069-133709 f
  ...
  ```

- `wav.scp` – mapovanie identifikátorov zvukových súborov na cesty k nim. Pokiaľ sa nejedná o jednokanálové (mono) wav, obsahuje príkaz, ktorý daný formát prekonvertuje do požadovanej formy. Príklad:

  ```
  103-1240-0000 flac -c -d -s ./train-D105-17/103/1240/103-1240-0000.flac |
  ```

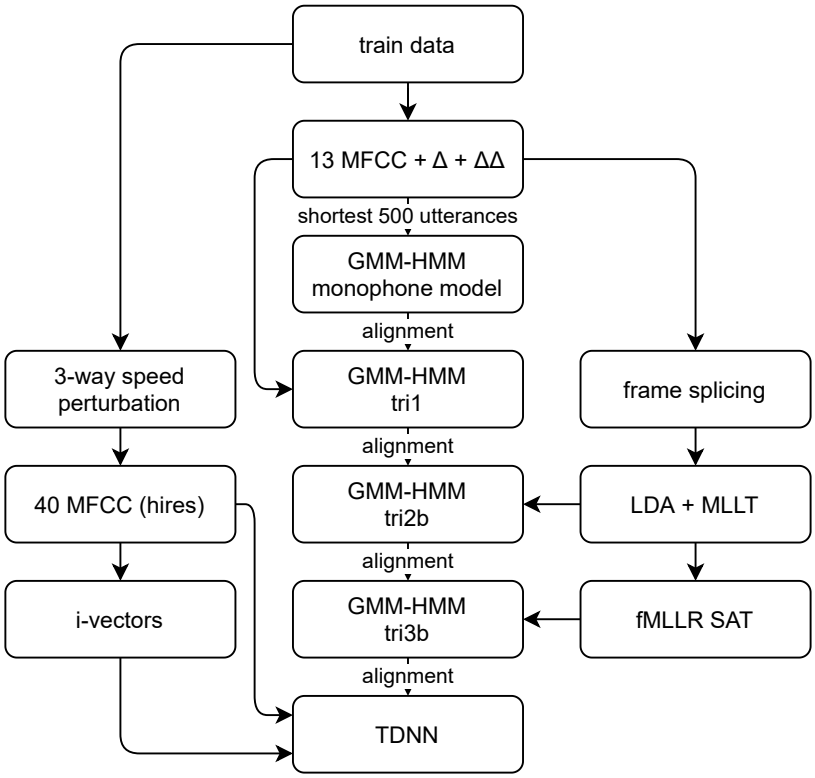
- `lexicon.txt` – zoznam slov a ich výslovností. Dôležité pre tvorbu akustického modelu založeného na skrytých Markovových modeloch (HMM). Príklad:

  ```
  ABDUCT  AE0 B D AH1 K T
  ABDUCTED  AE0 B D AH1 K T IH0 D
  ABDUCTED  AH0 B D AH1 K T IH0 D
  ...
  ```

- súbory `silence_phones.txt`, `nonsilence_phones.txt` a ďalšie s informáciami, ktoré „hlásky“ predstavujú skutočné hlásky, resp. ticho, ktoré je tiež modelované v akustickom modeli.

4.2.2 Trénovanie GMM-HMM modelov

Počas behu trénovania je postupne natrénovaných niekoľko modelov založených na GMM (Gaussian mixture model), ktorých komplexnosť sa s každým krokom zvyšuje. Okrem prvého modelu každý ďalší na začiatku dostáva tzv. zarovnania vytvorené predchádzajúcim modelom. Zarovnanie (alignment) je v terminológii Kaldi postupnosťou prechodov medzi stavmi HMM po najlepšej Viterbiho ceste referenčným prepisom preslovu [24].



Obr. 4.2: Bloková schéma receptu Mini Librispeech použitého v tejto práci.

Pred trénovaním je z nahrávok extrahovaných po 13 príznakov MFCC (Mel-frequency cepstral coefficients) na rámcoch s dĺžkou 25 ms a posunom 10 ms pre výpočet nasledujúceho rámca, čo sú predvolené hodnoty programu `compute-mfcc-feats`. Okrem toho sú pre každého rečníka spočítané z MFCC príznakov štatistiky pre CMVN (Cepstral mean and variance normalization). Za spomenutie stojí, že v tomto recepte je ako rečník myslená jedna prečítaná kapitola daným rečníkom (odpovedá jednému adresáru druhej úrovne ako bolo opísané v sekcii 3.1), nie všetky nahrávky od jednej osoby. Jedná sa o formu umelého zvýšenia variability trénovacích dát.

Ako prvý je natrénovaný monofónny model (kde jedna hláska odpovedá jednému stavu HMM) na podmnožine dát vybranej ako 500 najkratších nahrávok. Dôvodom je, že tieto nahrávky majú väčšiu šancu byť zarovnané správne, ako sa uvádza v komentári skriptu `run.sh`. Okrem MFCC príznakov sú použité aj delta a delta-delta, ktoré aproximujú prvú, resp. druhú deriváciu MFCC. Pomocou tohto modelu sú vytvorené zarovnania pre celú množinu trénovacích dát.

Na zarovnaniach z predchádzajúceho modelu je natrénovaný prvý trifónový model (kontextovo závislé hlásky sú modelované troma stavmi HMM) s príznakmi okrem MFCC aj delta a delta-delta.

Druhý trifónový model nepracuje s delta a delta-delta príznakmi, ale vstupujú do neho rámce vzniknuté spojením siedmich po sebe idúcich vektorov MFCC (pravý aj ľavý kontext aktuálneho rámca veľkosti 3). Dimenzionalita je redukovaná na 40 pomocou transformácie LDA (linear discriminant analysis) [2], ktorá sa snaží o maximalizáciu separability medzi jednotlivými triedami. Na týchto 40 príznakov je ďalej aplikovaná MLLT (maximum likelihood linear transform) [11], po ktorej majú byť príznaky lepšie modelovateľné pomocou Gaussových funkcií s diagonálnymi kovariančnymi maticami [27].

Posledný GMM-HMM model je založený na transformáciách LDA a MLLT, ale naviac sa tu pracuje s adaptáciou na rečníka (SAT – speaker adaptive training) pomocou fMLLR [10]. Zarovnania z tohto modelu sú ďalej použité pre účely trénovania neurónovej siete. Program `show-alignments` nám umožňuje náhľad na tieto zarovnania:

```
103-12-000  [2 8 18 17] [7526 7525] [1928 1927] [19182 19181 19181] [20816]
103-12-000  SIL         CH_B        AE1_I       P_I                 T_I
```

Na prvom riadku sa nachádzajú identifikátory stavov HMM, pod ním je odpovedajúca hláska zo súboru `phones.txt`. Prepis odpovedá začiatku slova „chapter“ s úsekom ticha pred ním.

4.2.3 TDNN model

Model založený na TDNN [36] (time-delay neural network) je finálny model receptu Mini LibriSpeech. Tento typ neurónovej siete sa dokáže naučiť pracovať s dlhými časovými závislosťami, ako napríklad reverberácia [21].

Najskôr sa zvýši objem datasetu pomocou trojcestnej rýchlostnej perturbácie (3-way speed perturbation) [14] – z existujúcich dát sú vytvorené ďalšie dve kópie, z ktorých je jedna spomalená na 0.9-násobok pôvodnej rýchlosti a druhá zrýchlená na 1.1-násobok. Z rozšírených dát sú extrahované tzv. MFCC príznaky s vysokým rozlíšením, čo znamená, že ich je 40 na každý rámec. Na týchto príznakoch sa najskôr natrénuje extraktor i-vektorov, ktoré poskytujú adaptáciu na rečníka. Ich výhodou oproti fMLLR je, že pri dekódovaní nie sú potrebné dva priechody, môžu byť extrahované online [23]. Do TDNN potom spoločne vstupuje 40 príznakov MFCC s i-vektormi o veľkosti 100.

4.3 Adaptácia v DNN

Adaptáciu rozpoznávača na rečníka a/alebo prostredie je možné u neurónových sietí zabezpečiť napríklad pripojením extra vektorov ku klasickým vektorom príznakov na vstupe. Príkladom sú techniky používané pôvodne v problematike verifikácie rečníka, i-vektory a x-vektory, ktorých využitie v rozpoznávačoch reči je analyzované napríklad v [21, 17, 12]. Obe metódy z nahrávok všeobecne ľubovoľnej dĺžky extrahujú nízkodimenzionálny vektor fixnej dĺžky, tzv. *embedding*. V systémoch pre rozpoznávanie rečníka je po extrakcii vektorov na skórovanie a následné rozhodnutie, či dvojica vektorov patrí jednému rečníkovi používaný skórovací model (back-end), napríklad PLDA (probabilistic linear discriminant analysis) [17]. Pre adaptáciu v DNN nie je potrebný, použité sú práve extrahované vektory.

4.3.1 i-vektory

Pôvodne predstavené v [6], i-vektory sú technika vychádzajúca z JFA (joint factor analysis). Je to estimácia vektora w_u podľa rovnice 4.1, kde m_0 je supervektor priemerov GMM[3] modelu UBM[4] (universal background model), s_u je supervektor adaptovaného GMM pre nahrávku u a $\mathbf{T}$ je matica úplnej variability s nízkou hodnosťou [7].

$$s_u = m_0 + \mathbf{T}w_u \tag{4.1}$$

Na rozdiel od JFA, kde je variabilita rečníkov a prostredí modelovaná dvoma separátnymi maticami, u i-vektorov je celá variabilita modelovaná $\mathbf{T}$. Ako bolo overené napríklad v [29],

[3] Supervektor priemerov GMM vzniká konkatenáciou priemerných vektorov jednotlivých zložiek GMM
[4] UBM je univerzálny model trénovaný na nahrávkach všetkých rečníkov

i-vektory poskytujú adaptáciu nie iba na rečníka, ale zároveň aj na prostredie. Proces trénovania matice **T** je bez supervízie, nepracuje sa s informáciou o príslušnosti nahrávok ku konkrétnym rečníkom. Každá nahrávka je braná ako keby pochádzala od iného rečníka [6].

Ako bolo skôr spomenuté, implementáciu i-vektorov používa aj recept Mini LibriSpeech. Najskôr je natrénovaný UBM na štvrtine dostupných dát po aplikovaní CMN (cepstral mean normalisation) s kĺzavým oknom. Následne je trénovaný extraktor i-vektorov s maticou **T** na vektoroch príznakov bez aplikovanej CMN. Dôvodom je, že samotný extrahovaný vektor má niesť informáciu o posunoch pre neurónovú sieť, aby mohol byť následne na vstupe s MFCC príznakmi, tiež bez aplikovanej normalizácie [21]. Posteriórne pravdepodobnosti Gaussových zložiek sú počítané z príznakov s aplikovanou CMN.

4.3.2 x-vektory

Druhou metódou tvorby adaptačných príznakov, s ktorými bude v tejto práci experimentované, sú x-vektory. Recept LibriSpeech ich použitie neobsahuje. Pôvodne boli sformulované v [31] ako náhrada i-vektorov pre rozpoznávanie rečníka. Na rozdiel od i-vektorov je v procese trénovania extraktora použitá supervízia vo forme značenia nahrávok identifikáciou príslušných rečníkov.

Podobne ako rozpoznávač v LibriSpeech, sú x-vektory založené na neurónovej sieti s time-delay architektúrou. Počiatočné vrstvy, tzv. *frame-level*, pracujú s jednotlivými rečovými rámcami, s postupne sa rozširujúcim časovým kontextom (existuje viac implementácií, v ktorých sa presná podoba vrstiev líši), nasledujú 2 „obyčajné vrstvy“, za nimi je tzv. *pooling* vrstva, ktorá počíta priemer a smerodajnú odchýlku z výstupu frame-level časti. Za pooling vrstvou nasledujú dve *segment-level* vrstvy pracujúce so štatistikami získanými z celej nahrávky. Výstupnou vrstvou je softmax o veľkosti rovnej počtu rečníkov. Aktivačná funkcia používaná v sieti je ReLU (rectified linear unit) [31].

V procese trénovania je pre príslušnosť nahrávky n-tému rečníkovi nastavený n-tý výstup softmax vrstvy na 1, ostatné na 0. Neurónová sieť sa teda učí klasifikovať nahrávky k rečníkom. Pri extrakcii je výsledný embedding získaný ako výstup jednej z dvoch segment-level vrstiev. Požadovaná veľkosť x-vektora sa nastaví ako šírka odpovedajúcej segment-level vrstvy v neurónovej sieti. Pre rozpoznávanie reči môže byť užitočné pri učení vymeniť klasifikáciu rečníkov za vlastnosti prostredia, napríklad impulzné odozvy použité pri augmentácii dát, ako bolo navrhnuté v [12].

Kapitola 5

Experimentálna časť

Táto kapitola zhŕňa prípravu experimentov a ich výsledky. Na začiatku je predstavený systém pre tvorbu kontaminovaných dát (reverberácia a pridanie šumu). Následne bola vykonaná krátka analýza dát dostupných v ReverbDB, zameraná na súvis pozície mikrofónu v miestnosti s prítomnosťou šumu v nahrávkach a akustických parametrov impulzných odoziev (ELR, DRR). Pre experimenty s rozpoznávačmi sú najskôr predstavené dáta použité pre vyhodnotenie úspešnosti. Nasledujú samotné experimenty, ktoré najskôr skúmajú limity dosiahnuteľné pôvodným receptom Mini LibriSpeech pre zvolené testovacie dáta. V záverečných sekciách sa snažíme modifikovať recept pre systém robustný voči rôznym akustickým podmienkam, ktorý by sa čo najviac priblížil skôr zmeraným limitom. Prínos rôznych modifikácií oproti stanovenému referenčnému systému bude percentuálne vyjadrený ako relatívny rozdiel alebo relatívne zlepšenie WER (po vzore z [12]):

$$\Delta\text{WER}, rel. = \frac{\text{WER}_{ref} - \text{WER}_{mod}}{\text{WER}_{ref}} \tag{5.1}$$

5.1 Implementácia procesu kontaminácie dát

Za účelom rýchlej a paralelnej kontaminácie veľkých datasetov bolo napísaných v jazykoch Python 3 a BASH niekoľko skriptov: `collect_vad`, `calculate_snr`, `prepare_lists`, `process_list` a `submit_reverb`.

Program `submit_reverb` použije najskôr prvé dva skripty na získanie potrebných informácií z čistých a reálne retransmitovaných nahrávok. Adresárové štruktúry rečového aj kontaminačného korpusu sú následne prejdené prípravným skriptom `prepare_lists`, ktorý vygeneruje inštrukcie pre skript `process_list`. Ten je potom spúšťaný v jednotlivých podúlohách v SGE. V rámci optimalizácie sú nahrávky pre spracovanie zoskupené do podúloh podľa použitého prostredia pri kontaminácii, takže program nemusí pre každú rečovú nahrávku znova načítavať nové impulzné odozvy a šum. Inštrukcie pre `process_list` sú predané v textovom súbore, kde na prvom riadku je uvedená cesta k impulzným odozvám a šumu, za ňou nasledujú trojice s cestami k zdrojovej a cieľovej nahrávke a hodnotou SNR, s ktorou je primiešaný šum. Tieto hodnoty sú generované buď rovnomerne v určitom rozsahu alebo podľa rozloženia pravdepodobnosti daného štatistikami z nahrávok reálneho retransmitu (podrobnejšie vysvetlené v sekcii 5.1.1).

Jednotlivé nahrávky z čistého datasetu sa všeobecne môžu vo výsledných kontaminovaných dátach nachádzať niekoľko krát, zakaždým v inom simulovanom prostredí. V Kaldi recepte musí mať ale každý preslov jednoznačný identifikátor, nie je preto možné vždy

použiť samotný skript `data_prep` pre LibriSpeech distribuovaný s toolkitom. Do prípravy potrebných súborov vstupuje aj skript `prepare_lists`. Aby sa zamedzilo duplicitám, sú v identifikátoroch nahrávok zahrnuté informácie o použitom „prostredí“ z ReverbDB, ktoré je jednoznačne určené miestnosťou, pozíciou reproduktora a číslom kanálu. Okrem toho je typicky pre jedno prostredie v ReverbDB zachytených niekoľko verzií impulzných odoziev a nahrávok šumu, ale napísané skripty neumožňujú kontamináciu jednej nahrávky dvoma verziami tej istej impulznej odozvy. `prepare_lists` potom pripraví mapovanie medzi pôvodnými a novými identifikátormi nahrávok, ktoré je možné použiť napr. pre odvodenie prepisu textu z pôvodného súboru `text` (viď sekciu 4.2.1).

5.1.1 Odhad odstupu signálu od šumu

Dôležitým faktorom z hľadiska porozumenia reči na nahrávke je odstup signálu od šumu (SNR – signal-to-noise ratio), ktorý sa bežne meria v dB a je možné ho vypočítať pomocou rovnice 5.2, kde P_s je výkon užitočného signálu a P_n je výkon šumu.

$$\mathrm{SNR} = \log_{10} \frac{P_s}{P_n} \tag{5.2}$$

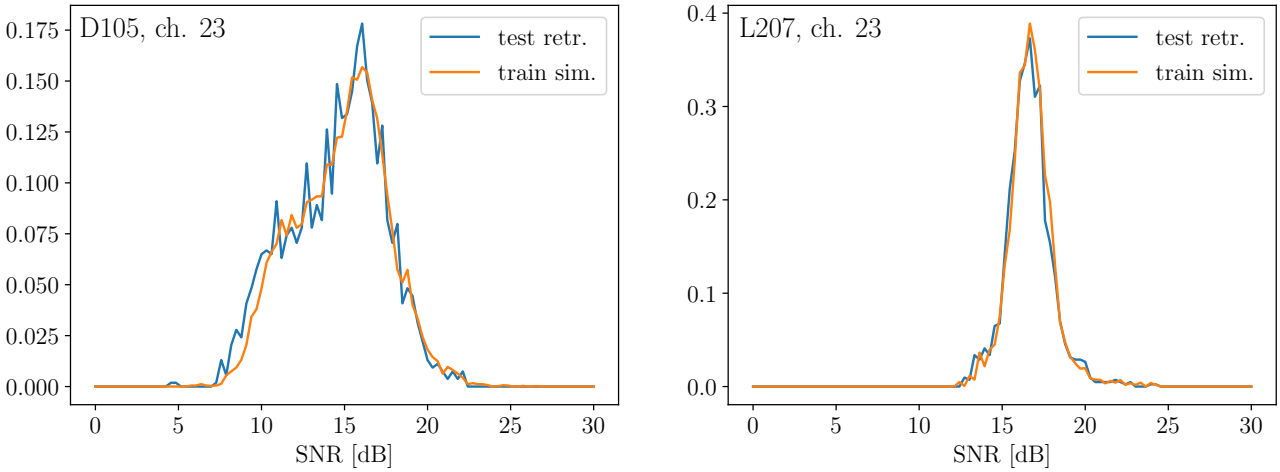
V databáze ReverbDB máme k dispozícii nahrávky šumu na pozadí bez reči a nahrávky testovacej časti LibriSpeech vrátane šumu, ale nie iba zreverberované verzie bez šumu, takže SNR musíme odhadovať. V [16] sa pracuje s pomerom výkonov rečového signálu vrátane šumu (P_{s+n}) a šumu bez reči (P_n), tzv. SNRp (5.3), ktorý budeme ďalej používať aj v tejto práci.

$$\mathrm{SNRp} = \log_{10} \frac{P_{s+n}}{P_n} \tag{5.3}$$

Pre výpočet SNRp pracujeme iba s úsekmi nahrávky, kde sa nachádza reč. Skript `collect_vad` je určený na extrakciu informácií o umiestnení týchto segmentov v čase z čistej verzie datasetu, u ktorého predpokladáme vyššiu presnosť oproti priamej extrakcii z kontaminovaných dát. Na detekciu rečovej aktivity (VAD – voice activity detection) je použitá knižnica `webrtcvad` pre Python.

Skript `calculate_snr` pracuje s časovými informáciami z `collect_vad` a pre každú nahrávku v kontaminovaných verziách počíta priemerný výkon rečových segmentov. Výkon šumu je vypočítaný z nahrávok `silence_16kHz_60sec.v00.wav` a pod. Výstupom je SNRp pre každú nahrávku a priemerné SNRp celej verzie datasetu.

V retransmitovaných testovacích dátach je v dôsledku rôznej hlasitosti rečníkov z čistých nahrávok určitá variabilita SNRp naprieč nahrávkami. Hodnoty SNRp jednotlivých nahrávok zhromaždené skriptom `calculate_snr` je možné skúmať pomocou histogramu. Ten môže slúžiť ako základ rozloženia pravdepodobnosti pre generovanie požadovaného SNR do súboru s inštrukciami pre kontamináciu skriptom `process_list`. Tento prístup je užitočný v prípade, ak chceme generovať množstvo trénovacích dát s charakteristikou odpovedajúcou retransmitom z daného kanála v ReverbDB. Príklady rozložení SNRp v retransmitoch a odpovedajúcich umelo kontaminovaných dátach je možné sledovať na Obr. 5.1.



Obr. 5.1: Porovnanie rozloženia pravdepodobnosti SNRp nahrávok z reálnych retransmitov a simulovaných dát vytvorených skriptom u vybraného kanálu z miestností L207 a D105.

5.1.2 Reverberácia a primiešanie šumu

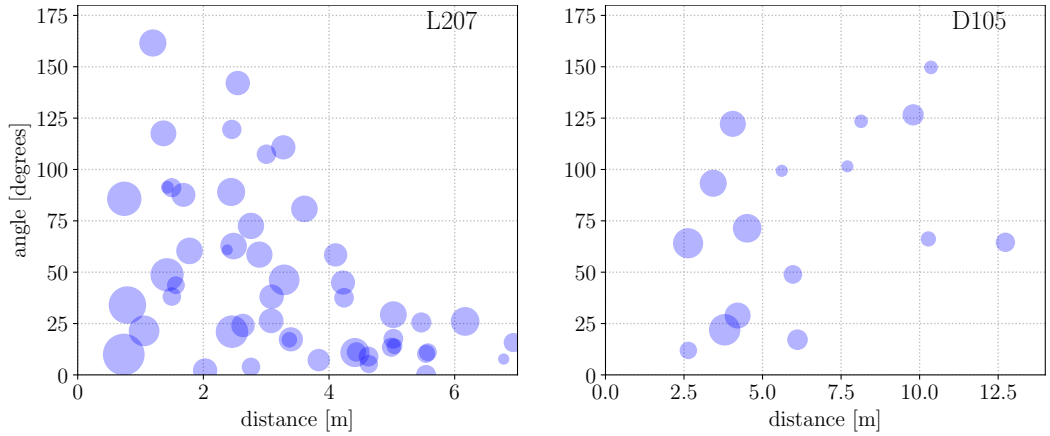
Pre simuláciu šírenia zvuku nahrávky v danom prostredí sa používa konvolúcia s odpovedajúcou impulznou odozvou. Pokiaľ adresár s prostredím obsahuje viac verzií impulznej odozvy, sú pre nahrávky vyberané náhodne s rovnomerným rozložením. V tejto práci používa skript `process_list` implementáciu konvolúcie metódou `convolve` z knižnice `scipy.signal` pre Python. Impulzná odozva zmeraná metódou ESS okrem reverberácie obsahuje aj oneskorenie impulzu závisiace na rýchlosti šírenia zvuku a vzdialenosti zdroja a prijímača. Aby sme mohli pre účely odhadu výkonu rečovej zložky signálu použiť VAD z čistých nahrávok, musí byť toto oneskorenie v IR kompenzované. V ReverbDB boli IR vykompenzované autormi.

Časti nahraného šumu primiešavané k reči sú vyberané sekvenčne, takže výsledné nahrávky vznikajú ako keby boli postupne v prehrávané v danej miestnosti. V inštrukciách z `prepare_lists` sa nachádza požadovaná úroveň šumu pre výslednú nahrávku. Pre získanie signálu s určitým SNR je potrebné vzorky reverberovanej reči deliť koeficientom c (5.4). Význam symbolov P_s a P_n je v zmysle predchádzajúcich rovníc.

$$c = \sqrt{\frac{P_s}{P_n} \cdot 10^{-\frac{\mathrm{SNR}}{10}}} \tag{5.4}$$

5.2 Analýza dát z ReverbDB

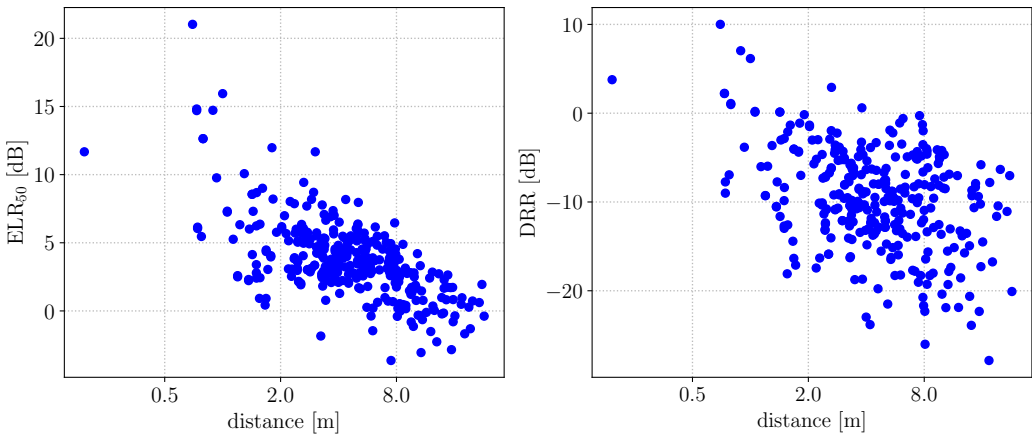
Rôzne predchádzajúce práce identifikovali niekoľko číselných metrík, ktoré majú vplyv na úspešnosť rozpoznávania reči. V [19] bol sledovaný rozdiel medzi úspešnosťou rozpoznávania reverberovaných dát bez primiešaného šumu oproti dátam obsahujúcim šum. Tento rozdiel sa zvyšoval so vzdialenosťou mikrofónu a uhlom v zmysle polárnych súradníc s počiatkom v bode s umiestneným reproduktorom. Okrem toho, v článku [16] bola sledovaná negatívna korelácia medzi odstupom signálu a šumu a úspešnosťou rozpoznávania. ReverbDB obsahuje množstvo nahrávok reči zaznamenanej v rôznych pozíciach mikrofónu. Z nich môžeme sledovať závislosť týchto pozícii na SNRp (vysvetlené v 5.1.1), výsledky z dvoch miestností sú na Obr. 5.2.



Obr. 5.2: Závislosť priemerného SNRp nahrávok jednotlivých kanálov (veľkosť bublín) na vzdialenosti zdroja a príjímača a odklone pozície mikrofónu od osi membrány reproduktora prehrávajúceho nahrávky čistej reči v miestnostiach L207 (obsahuje 3 rôzne pozície reproduktora) a D105 (1 pozícia reproduktora). V štatistike sú zahrnuté iba mikrofóny s výhľadom na reproduktor.

Pri oboch miestnostiach pozorujeme trend v poklese SNRp so vzďaľujúcim sa mikrofónom aj so zvyšujúcim sa uhlom. Tieto dva parametre však nie sú jediným faktorom ovplyvňujúcim odstup signálu od šumu. Niektoré mikrofóny sa nachádzajú v tesnej blízkosti zdrojov šumu, napríklad chladenia počítačov. Tieto informácie však nie sú v metadátach ReverbDB obsiahnuté. To môže vysvetľovať prítomnosť „malých bublín“ v relatívne malej vzdialenosti od reproduktora.

Pokles úspešnosti rozpoznávania bol v [19] pozorovaný aj naprieč reverberovanými testovacími datasetmi bez primiešaného šumu, čo naznačuje prítomnosť iných, čisto akustických vlastností, ktoré môžu mať vplyv na úspešnosť rozpoznávania. Takými môžu byť napríklad doba dozvuku, DRR alebo ELR (opísané v 2.3.1, 2.3.2, resp. 2.3.3). Doba dozvuku je približne konštantná vrámci určitej miestnosti, napriek tomu boli pozorované rozdiely v úspešnosti rozpoznávania naprieč rôznymi pozíciami. Budeme sa preto zaujímať o ELR a DRR, ktoré nie sú v rámci miestnosti konštantné. Závislosť hodnôt na vzdialenosti je vykreslená na Obr. 5.3. Keďže pre tento experiment nepotrebujeme reálne retransmity, bola do dát pridaná aj miestnosť E112 pre pokrytie rôzne objemných miestností.



Obr. 5.3: Závislosť ELR (s hranicou medzi skorými a neskorými odrazmi nastavenou na 50 ms) a DRR na vzdialenosti zdroja a príjímača v logaritmickej mierke. Hodnota Pearsonovho korelačného koeficientu medzi ELR a logaritmom vzdialenosti je približne -0.65, pre DRR je to -0.42. Pre lineárnu vzdialenosť sú hodnoty nižšie: -0.57, resp. -0.33.

Pri logaritmickej škále vzdialenosti vidíme klesajúci trend ELR, teda energia neskorých odrazov rastie v pomere k energii skorých odrazov so vzďaľujúcim sa mikrofónom. Pre DRR je tento vzťah menej badateľný. V [38] sa uvádza, že kým energia odrazov so zvyšujúcou sa vzdialenosťou ostáva približne rovnaká, energia priamej zložky sa s dvojnásobkom vzdialenosti znižuje približne o 6 dB. Vo vzťahu k natočeniu reproduktora nebola viditeľná výraznejšia závislosť. V nasledujúcej sekcii budeme vykresľovať závislosť rozpoznávania aj vzhľadom k SNRp, ELR a DRR.

5.3 Výber testovacích dát

V nasledujúcich experimentoch sú opakovane trénované rozpoznávače, ktoré budeme navzájom porovnávať. Táto sekcia predstavuje zloženie dát, ktoré budú použité na vyhodnotenie úspešnosti. Z dostupných reálnych retransmitov nahraných miestnostiach D105 a L207 (ktoré boli predstavené v sekciách 3.2.1, resp. 3.2.2) bola vybraná podmnožina kanálov so zameraním na variabilitu vzdialenosti mikrofónu od reproduktora. Ku niektorým pozíciám mikrofónov boli vybrané iné kanály v podobnej vzdialenosti, ale skryté za prekážkou, aby bolo možné v experimente v sekcii 5.4 skúmať dopad prekážky na pokles úspešnosti. Zoznám kanálov z D105 spolu so základnými údajmi je v Tabuľke 5.1, z L207 v Tabuľke 5.2.

kanál	vzdialenosť	odklon	viditeľný	SNRp	DRR	ELR_{50}	ELR_{110}
22	2.53 m	19.23°	nie	13.49 dB	−6.58 dB	1.53 dB	6.34 dB
13	2.63 m	64.02°	áno	21.62 dB	−1.96 dB	6.05 dB	10.50 dB
21	3.09 m	48.71°	nie	14.34 dB	−9.52 dB	4.84 dB	9.64 dB
15	4.51 m	71.19°	áno	20.49 dB	−6.44 dB	2.03 dB	7.12 dB
19	5.97 m	48.91°	áno	13.00 dB	−1.10 dB	2.94 dB	8.17 dB
20	7.70 m	101.54°	áno	8.02 dB	−3.91 dB	0.62 dB	6.21 dB
23	10.05 m	35.59°	nie	15.33 dB	−10.02 dB	1.16 dB	6.75 dB
17	10.28 m	66.18°	áno	10.44 dB	−8.50 dB	4.20 dB	8.52 dB
29	12.73 m	63.57°	áno	13.53 dB	−6.99 dB	1.76 dB	5.89 dB

Tabuľka 5.1: Vybrané kanály z miestnosti D105, ktorých reálne retransmity LibriSpeech *test-clean* budú použité na vyhodnotenie úspešnosti rozpoznávania. Všetky kanály sú vybrané z pozície reproduktora SpkID č. 7. Poradie je určené vzdialenosťou mikrofónu od reproduktora.

kanál	vzdialenosť	odklon	viditeľný	SNRp	DRR	ELR_{50}	ELR_{110}
24	1.50 m	38.05°	áno	12.91 dB	−11.66 dB	2.41 dB	10.25 dB
17	1.53 m	23.93°	nie	8.27 dB	−10.17 dB	3.76 dB	11.62 dB
18	2.76 m	0.44°	áno	12.93 dB	−5.23 dB	6.02 dB	13.82 dB
23	3.39 m	17.29°	áno	17.12 dB	−7.71 dB	5.32 dB	12.46 dB
01	4.61 m	2.62°	áno	13.79 dB	−9.52 dB	3.74 dB	11.89 dB
21	5.57 m	10.97°	áno	12.37 dB	−9.36 dB	3.73 dB	11.78 dB
29	6.42 m	2.38°	nie	2.93 dB	−10.33 dB	2.35 dB	8.45 dB
16	6.78 m	1.63°	áno	7.43 dB	−12.03 dB	3.82 dB	10.89 dB

Tabuľka 5.2: Vybrané kanály z miestnosti L207, ktorých reálne retransmity LibriSpeech *test-clean* budú použité na vyhodnotenie úspešnosti rozpoznávania. Všetky kanály sú vybrané z pozície reproduktora SpkID č. 6. Poradie je určené vzdialenosťou mikrofónu od reproduktora.

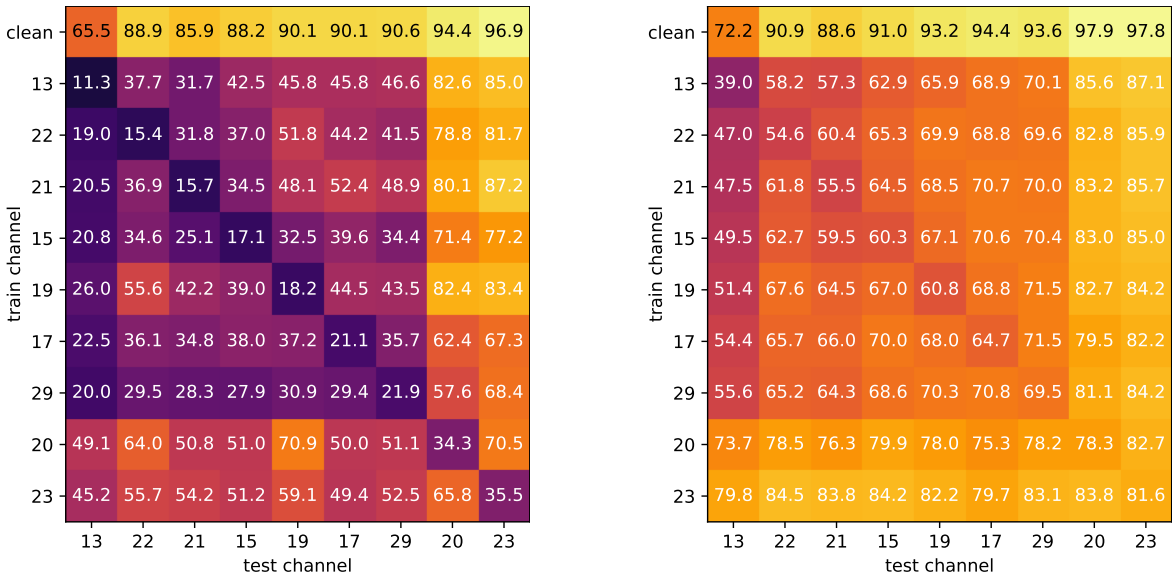
5.4 Trénovanie pôvodného receptu na cieľových podmienkach

V tomto experimente opakovane trénujeme rozpoznávač založený na recepte Mini LibriSpeech na verziách *train-clean-100*, kde každá verzia vznikla konvolúciou všetkých nahrávok s impulznou odozvou prináležiacou jednému konkrétnemu kanálu z testovacích dát predstavených v 5.3. K reverberovaným nahrávkam boli primiešané vzorky šumu z rovnakého kanála s variabilným SNR, kde rozloženie pravdepodobnosti hodnôt SNR odpovedalo štatistikám získaným z reálneho retramsmitu *test-clean*, tiež z daného kanála (podľa postupu v 5.1.2).

Originálna verzia receptu pracujúca s Mini LibriSpeech dátami `train-clean-5` pre trénovanie a `dev-clean-2` pre testovanie bola parametrizovaná, aby mohla používať nami pripravené verzie zvolenej 100 hodinovej podmnožiny LibriSpeech pri trénovaní a 5 hodinové reálne retransmity pre testovanie.

Týmto experimentom sledujeme možnosti zvoleného receptu, ktoré sú dosiahnuteľné v ideálnom prípade, teda ak umožníme systému pri trénovaní sústrediť sa cieľové prostredie. Zároveň sledujeme, do akej miery je rozpoznávač trénovaný s jednou impulznou odozvou z danej miestnosti schopný rozpoznávať reč zachytenú pre iné pozície mikrofónu v rovnakej miestnosti. Najskôr boli natrénované rozpoznávače pre všetky testovacie kanály z D105 (celkovo 9).

Vrámci trénovacieho procesu všetkých verzií akustického modelu, od monofónneho až po neurónovú sieť, mal recept k dispozícii iba kontaminované dáta. S predposlednou verziou, GMM s adaptáciou na rečníka (tzv. *tri3b*) a poslednou TDNN boli pre všetkých 9 rozpoznávačov zároveň dekódované reálne retransmity *test-clean* rovnakej množiny kanálov a pre každý kanál separátne vyhodnotená chybovosť rozpoznávania vyjadrená ako WER. Výsledky sú porovnané s úspešnosťou baseline systému natrénovaného na nekontaminovanej verzii použitých trénovacích dát. Tento systém dosiahol pre TDNN model úspešnosť na úrovni **5.99 %** a **10.54 %** pre GMM pri rozpoznávaní nekontaminovanej verzie destovacích dát. Takto vznikli matice, ktoré môžeme vidieť na Obr. 5.4.



(a) TDNN model

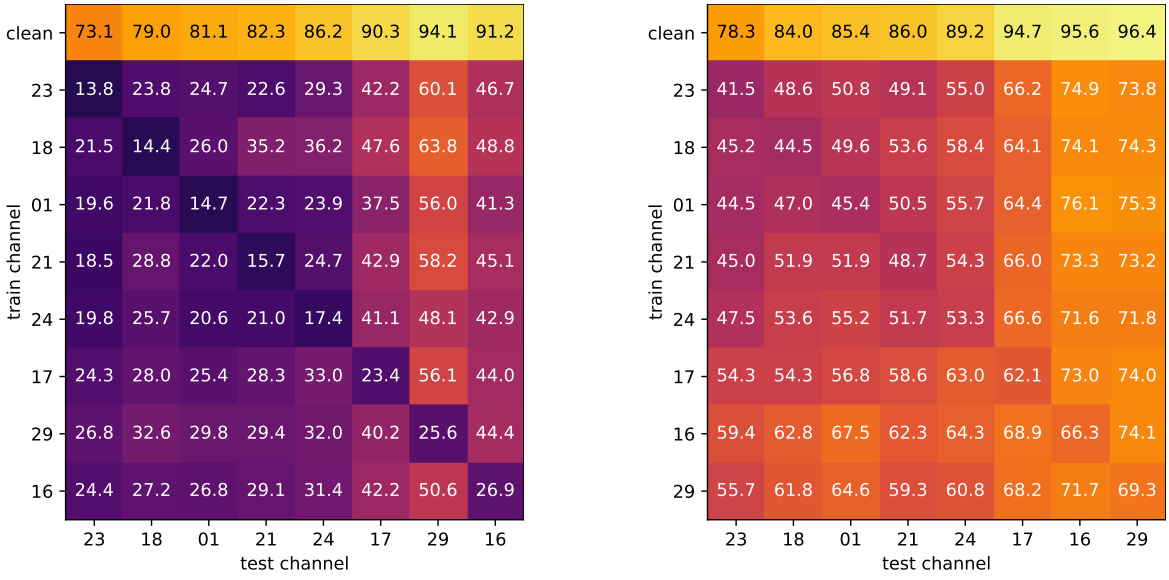
(b) GMM model (*tri3b*)

Obr. 5.4: Prehľad chybovosti rozpoznávania dát (WER) z rôznych kanálov z miestnosti D105 rozpoznávačmi trénovanými na dátach s úzkym zameraním na odpovedajúce kanály. Poradie kanálov je určené vzostupne podľa chybovosti rozpoznávania dát odpovedajúcich trénovaciemu setu. Riadok *clean* predstavuje výsledky rozpoznávania baseline systémom.

Na obidvoch maticiach môžeme pozorovať najnižšie WER na diagonále, čo znamená, že najlepším modelom pre rozpoznávanie v danom prostredí, podľa očakávania, je ten s odpovedajúcimi podmienkami. Tmavšie farby pod diagonálou oproti osovo súmerným hodnotám nad diagonálou značia, že modely so zložitejšími podmienkami[1] majú tendenciu byť relatívne úspešné aj s jednoduchšími testovacími dátami a naopak. Poradie úspešnosti rozpoznávania jednotlivých kanálov baseline systémom neodpovedá poradiu schopnosti kanálov naučiť sa rozpoznávať v odpovedajúcich podmienkach. TDNN aj GMM modely však zoradili kanály podľa zložitosti v rovnakom poradí.

Okrem jednoznačného rozdielu v úspešnosti rozpoznávania vidíme rozdiel vo výraznosti diagonály. To znamená, že TDNN sa dokáže lepšie adaptovať práve na cieľové podmienky. Až na jednu výnimku platí, že TDNN je zakaždým najúspešnejšia práve pri rozpoznávaní odpovedajúcich dát a menej úspešná pri rozpoznávaní iných kanálov. V prípade GMM *tri3b* vidíme, že so zložitosťou podmienok klesá schopnosť adaptácie a pri najzložitejších podmienkach trénovací kanál (na diagonále) nevyčnieva spomedzi ostatných kanálov (mimo diagonály).

Rovnaký experiment bol vykonaný aj so všetkými ôsmimi kanálmi z menšej miestnosti L207 (sekcia 3.2.2). Výsledky sú na Obr. 5.5.



(a) TDNN model

(b) GMM model (*tri3b*)

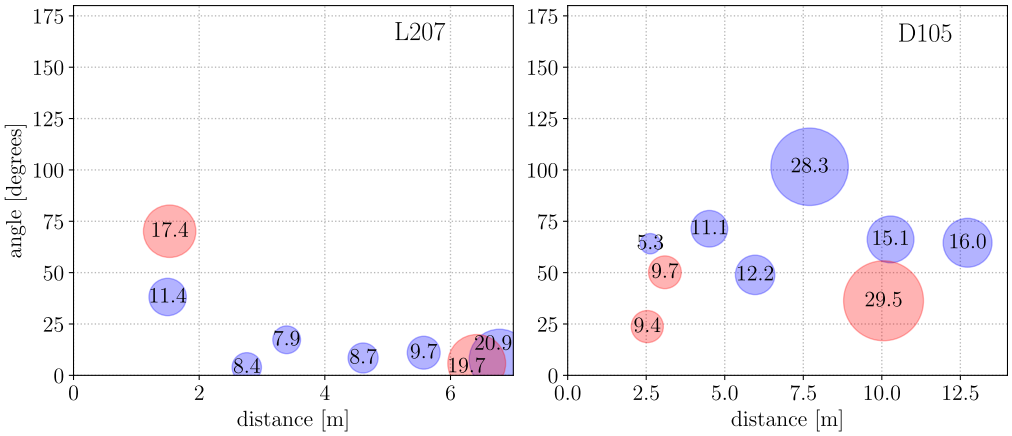
Obr. 5.5: Prehľad chybovosti rozpoznávania dát (WER) z rôznych kanálov z miestnosti L207 rozpoznávačmi trénovanými na dátach s úzkym zameraním na odpovedajúce kanály. Poradie kanálov je určené vzostupne podľa chybovosti rozpoznávania dát odpovedajúcich trénovaciemu setu. Riadok *clean* predstavuje výsledky rozpoznávania baseline systémom.

Z matice úspešnosti rozpoznávania pre L207 môžeme vyvodiť podobné závery ako pri D105. Rozdiel je, že kým pre D105 poradie úspešnosti rozpoznávania kanálov baseline systémom neodpovedalo poradiu schopnosti systému naučiť sa na dané podmienky, v L207 toto poradie u GMM modelu sedí a pre TDNN sú vymenené iba posledné dva kanály.

Na Obr. 5.6 je vykreslená závislosť úspešnosti adaptácie systému na vzájomnej pozícii mikrofónu a reproduktora pre jednotlivé kanály, rozdelené podľa miestností. Hodnoty sú

[1] Ako provizórnu metriku „zložitosti“ podmienok berieme schopnosť receptu Mini LibriSpeech rozpoznávať dáta zo súhlasného prostredia (vyjadrenú ako WER).

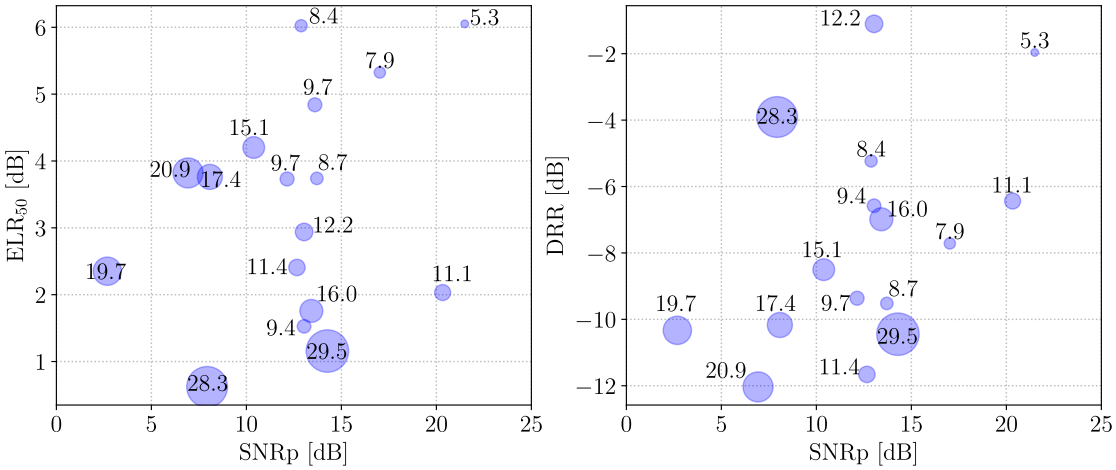
udávané v tzv. ΔWER, čo predstavuje rozdiel vo WER z rozpoznávania dát daného kanála rozpoznávačom z odpovedajúcich podmienok (t. j. hodnôt z diagonál predchádzajúcich grafov) a WER baseline systému TDNN pri rozpoznávaní čistých dát (5.99 %).



Obr. 5.6: Závislosť ΔWER u systémov rozpoznávajúcich dáta zo súhlasných kanálov na vzdialenosti od reproduktora a natočenia v miestnostiach L207 a D105. Modré bubliny predstavujú mikrofóny s výhľadom na reproduktor, červená znázorňuje uzavreté mikrofóny.

Pre D105 pozorujeme mierny pokles úspešnosti so zvyšujúcou vzdialenosťou. Skryté mikrofóny dosahujú nižšiu úspešnosť oproti viditeľným mikrofónom s podobnou vzdialenosťou a uhlom. V miestnosti L207 sú s jednou výnimkou rozdiely pre rôzne vzdialenosti ešte nižšie a takisto platí jav s poklesom úspešnosti v dôsledku uzavretia mikrofónov. Pre porovnanie, v [19] ukázali experimenty so zafixovaným rozpoznávačom pokles úspešnosti na 6 metrov o 40 a viac percent pri rozpoznávaní reálnych retransmitov z tejto databáze u nahrávok z miestnosti L207.

Okrem pozície mikrofónov môžeme úspešnosť skúmať aj v súvislosti so SNR, ELR a DRR. Na Obr. 5.7 je vykreslená závislosť ΔWER na týchto parametroch. Pre neprehľadnosť vizualizácie hodnôt v trojrozmernom priestore sú hodnoty vykreslené v dvoch grafoch po dvojiciach SNRp a ELR, resp. SNRp a DRR.



Obr. 5.7: Naľavo závislosť ΔWER u systémov rozpoznávajúcich dáta zo súhlasných kanálov na SNRp a ELR s hranicou skorých a neskorých obrazov nastavenou na 50 ms, napravo rovnaká veličina v závislosti na SNRp a DRR. Dáta sú v oboch grafoch spoločne z miestností L207 a D105.

V sekcii 5.2 bolo pozorované nižšie SNRp, ELR a DRR so vzďaľujúcim sa mikrofónom, podobne aj pri úspešnosti rozpoznávania v predchádzajúcich grafoch. V súlade s týmito pozorovaniami vidíme menej úspešné rozpoznávače bližšie k ľavému dolnému rohu grafov, naopak najúspešnejší rozpoznávač odpovedá najväčšiemu odstupu signálu od šumu a najsilnejšiemu zastúpeniu skorých odrazov, resp. priamej cesty v energii impulznej odozvy. Pri hlbšej analýze nečakaného umiestnenia rozpoznávača s $\Delta\mathrm{WER} = 29.5\,\%$ bola zistená relatívne vysoká smerodajná odchýlka v hodnotách SNRp naprieč jednotlivými nahrávkami z retransmitu, na úrovni 2.93, pričom obvyklá smerodajná odchýlka u ostatných kanálov je rozmedzí 1 až 2. Na Obr. 5.1 je na ľavej strane histogram nameraných SNRp v nahrávkach z tohto kanála v porovnaní s kanálom s obvyklým rozložením hodnôt SNRp u príslušných nahrávok.

5.5 Využitie čistého datasetu pri trénovaní

Pôvodný recept pozná iba 2 množiny dát – trénovaciu, ktorú systém vidí pri učení a testovaciu, použitú pri vyhodnotení. V prípade predchádzajúceho experimentu videl systém pri trénovaní GMM aj TDNN iba kontaminované dáta. Ako bolo načrtnuté v Kapitole 4.2, TDNN nadväzuje na GMM systémy tak, že použije zarovnanie rečových rámcov na stavy HMM ako supervíziu pri trénovaní neurónovej siete. V receptoch LibriSpeech sa namiesto exaktných zarovnaní používajú tzv. *zväzy* (lattices), ktoré predstavujú viac alternatívnych ciest prechodmi HMM, čím je podľa komentárov v kóde receptu dosiahnutá väčšia voľnosť pri trénovaní.

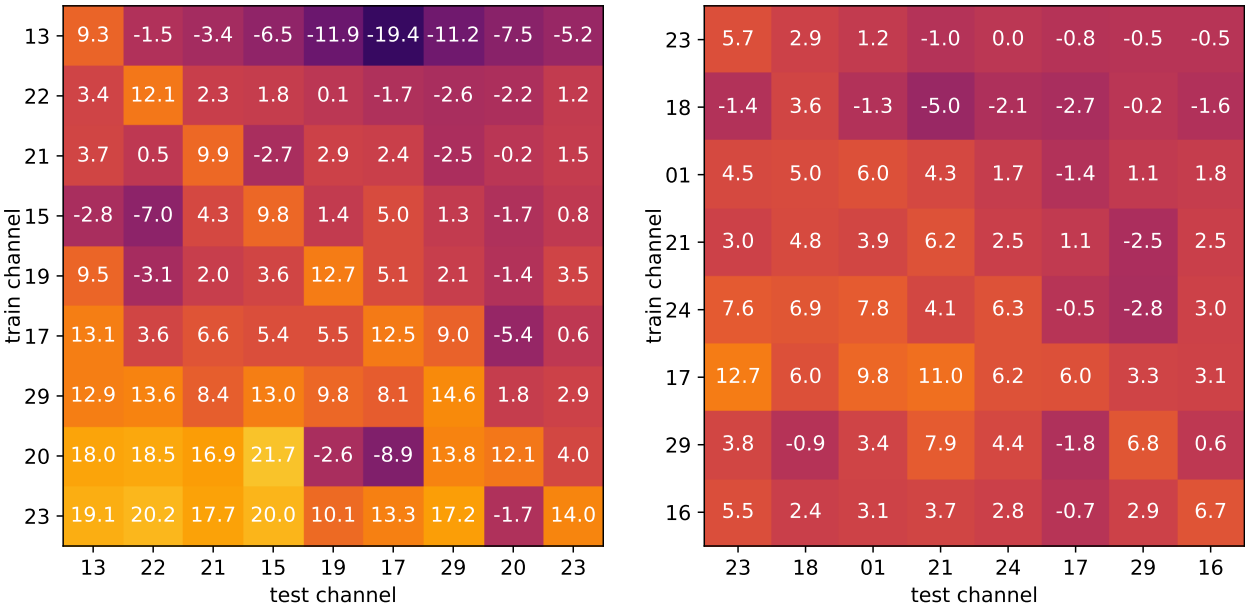
U GMM sme pozorovali relatívne slabú úspešnosť rozpoznávania kontaminovaných dát aj napriek tomu, že bol trénovaný na zhodných akustických podmienkach. Dá sa predpokladať, že takto trénovaný GMM model bude generovať nepresné zarovnanie, pričom sa tieto nepresnosti prejavia pri trénovaní TDNN. Problémy so zarovnaním sú analyzované napríklad v [22].

Článok [28] ako jedno z vylepšení pre trénovanie rozpoznávačov robustných voči reverberácii predstavuje využitie čistej verzie dát pre trénovanie GMM modelu, spolu s generovaním zarovnaní a ich následné využitie pri trénovaní TDNN, tentokrát s príznakmi z reverberovaných dát. Zároveň je prezentované výrazné zvýšenie úspešnosti pri použití tohto postupu. Bez dostupných dvoch paralelných verzií, čistého a kontaminovaného datasetu, nie je možné túto metódu použiť. Spôsob tvorby reverberovaných dát v tejto práci nám však umožňuje s týmto vylepšením experimentovať, čím sa ďalej zaoberá táto sekcia.

Cieľom experimentu bolo natrénovať GMM rozpoznávač na čistom *train-clean-100* a vytvoriť zarovnania týchto dát. Následne boli na 17 vybraných prostrediach z miestností L207 a D105 (detaily týchto kanálov sú predstavené v sekcii 5.3) trénované TDNN rozpoznávače (znova pre každé prostredie osobitne) za využitia týchto zarovnaní.

Existujúce skripty s parametrami `train-set` a `test-sets` boli rozšírené o nový parameter `train-env-set`, ktorý špecifikuje kontaminovanú verziu čistého datasetu `train-set`. Keďže TDNN časť pracuje so strojnásobeným objemom dát pomocou rýchlostnej perturbácie, bolo potrebné strojnásobiť čistú aj kontaminovanú verziu. Z čistých dát sú extrahované klasické 13 dimenzionálne vektory MFCC príznakov použité na vytvorenie zarovnávacích zväzov GMM *tri3b* modelom. Z kontaminovaných dát sú extrahované 40 dimenzionálne vektory MFCC príznakov ako vstup neurónovej siete a tiež pre trénovanie a extrakciu i-vektorov. Pre všetky TDNN rozpoznávače sú však použité rovnaké zarovnania, takže bol zároveň vytvorený skript `retrain_dnn.sh`, ktorý obsahuje iba kroky pre pretrénovanie i-vektorov a neurónovej siete, aby nebolo nutné opakovane trénovať aj GMM systém.

Jednotlivými rozpoznávačmi odpovedajúcimi kanálom boli podobne ako v sekcii 5.4 rozpoznané aj testovacie dáta z ostatných kanálov danej miestnosti, čím vznikli ďalšie štvorcové matice úspešnosti, vykreslené na Obr. 5.8. Hodnoty sú udávané ako relatívne percentuálne zlepšenie WER oproti odpovedajúcim skóre z predchádzajúceho experimentu, t. j. oproti systému bez použitia čistých dát pre zarovnanie.



(a) Miestnosť D105

(b) Miestnosť L207

Obr. 5.8: Matice relatívnych percentuálnych zlepšení úspešnosti rozpoznávania pri použití close-talk dát pre natrénovanie GMM systému a zarovnanie oproti pôvodnému použitiu kontaminovaných dát pre tento účel. Poradie kanálov je rovnako ako na Obr. 5.4 a 5.5, radené vzostupne podľa chybovosti rozpoznávania dát odpovedajúcich trénovaciemu setu.

Použitie čistého datasetu pre zarovnania prináša pri rozpoznávaní kanálov odpovedajúcimi systémami (na diagonálach) vo všetkých 17 prípadoch zníženie chybovosti. Zaujímavé je však pozorovať, že kým u systémov trénovaných na akusticky zložitejších podmienkach táto technika priniesla vo väčšine prípadov zlepšenie rozpoznávania aj ostatných kanálov z danej miestnosti (hodnoty pod diagonálou), systémy trénované na „jednoduchších“ kanáloch prinášajú buď zhoršenie úspešnosti (záporné čísla), alebo len minimálne zlepšenie pri rozpoznávanie dát z neodpovedajúcich kanálov.

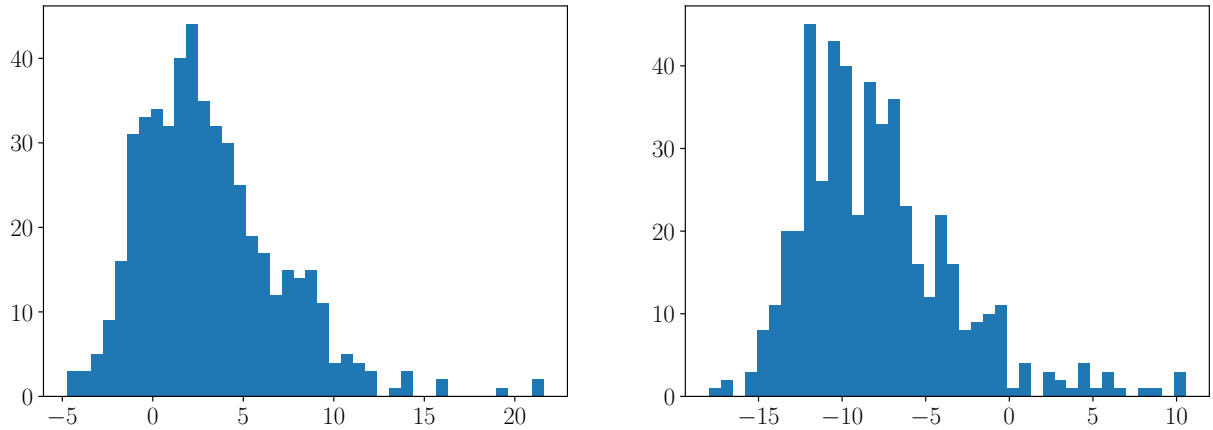
Vidíme aj značný rozdiel v prínose tejto techniky pre rôzne miestnosti, kým pre D105 bolo priemerné zlepšenie na úrovni 12 %, u L207 je priemerné zlepšenie iba polovičné. Jedným z vysvetlení by mohlo byť, že impulzné odozvy z miestnosti L207 boli menej precízne kompenzované ako IR z D105. Podľa manuálnej kontroly sa však prvotný impulz v IR z oboch miestností nachádza približne v čase $t = 2\,\text{ms}$, teda pätina jednej periódy rámcov pre extrakciu príznakov, čo sa dá považovať za dostatočne precízne.

5.6 Zmiešaný trénovací dataset

V predchádzajúcich dvoch experimentoch sme skúmali možnosti vybraného receptu rozpoznávať reč zachytenú v rôznych prostrediach, pokiaľ sa mohol systém „sústrediť“ na konkrétne

podmienky. Úspešnosť rozpoznávania pri použití odpovedajúcej impulznej odozvy a šumu sa síce veľmi výrazne zvýšila oproti baseline systému trénovanému na pôvodných čistých dátach, takéto použitie však nie je veľmi praktické. Pokiaľ by sme chceli rozpoznávač adaptovať na iné prostredie, potrebovali by sme zakaždým vysoko výkonné výpočtové prostredie vrátane grafických kariet na pretrénovanie neurónovej siete. Budeme sa preto zaujímať o využitie zmesi viac impulzných odoziev a šumu z ReverbDB pri kontaminácii jednotlivých nahrávok datasetu pre natrénovanie rozpoznávača robustného voči rôznym novým prostrediam. Tiež bude sledovaný prípadný pokles úspešnosti rozpoznávania dát z vybraných testovacích prostredí oproti systémom z predchádzajúceho experimentu, ktoré budú označované ako „oracle train“.

Zmiešaný dataset vytvorený pre ďalšie experimenty je založený podobne ako v predchádzajúcich sekciách na LibriSpeech *train-clean-100*. Do výberu boli zahrnuté kanály z miestností Q301, L212, R112, CR2, E112 a C236. Miestnosti L207 a D105 nie sú vo výbere, pretože sa snažíme sledovať, ako sa natrénované systémy za behu adaptujú na nevidené prostredia. Schodisko L227 bolo vyňaté tiež, pretože impulzné odozvy v ReverbDB sú časovo ohraničené na dĺžku 1 s, doba dozvuku z tejto miestnosti je ale výrazne dlhšia ako táto hranica, čo viedlo k orezaniu užitočných informácií z ich konca. Po odfiltrovaní miestností ďalej vznikla zo zostávajúceho výberu náhodná podmnožina 500 kanálov, ktorými boli dáta kontaminované. Pre každú nahrávku reči bolo zo spomenutej podmnožiny náhodne vybrané prostredie s rovnomerným rozložením. Hodnoty SNR pre primiešavanie šumu boli vyberané tiež s rovnomerným rozložením a to z rozsahu $\langle 0, 30 \rangle$. Naproti tomu, v predchádzajúcich experimentoch kopírovalo generované rozloženie SNR v trénovacích dátach hodnoty z testovacích retransmitov. Rozloženie hodnôt ELR_{50} a DRR impulzných odoziev z vybraných prostredí je zobrazené na Obr. 5.9. Vidíme, že hodnoty z používaných testovacích prostredí sú pokryté aj v zmiešanom datasete. Priemerné WER pri rozpoznávaní testovacích kanálov pôvodným systémom Mini LibriSpeech trénovaným na takto vytvorenom datasete je 28.75 %.



(a) ELR_{50} [dB] (b) DRR [dB]

Obr. 5.9: Histogramy zastúpenia hodnôt ELR_{50} a DRR v impulzných odozvách z prostredí vybraných do trénovacieho datasetu v nasledujúcich experimentoch.

5.7 Úprava informácie o rečníkoch pre systém i-vektorov

Ako bolo spomenuté v sekcii 4.3.1, použitie i-vektorov na vstupe neurónovej siete rozpoznávača reči poskytuje adaptáciu na rečníka aj prostredie a v článku [21] je prezentovaný ich

výrazný prínos v úspešnosti rozpoznávania. Sú zároveň súčasťou nami využívaného receptu. Aj keď pri trénovaní systému pre ich extrakciu nie je potrebná priamo supervízia v podobe mapovania nahrávok ku konkrétnym rečníkom, v skúmanom recepte i-vektory využívajú informáciu o rečníkoch zo súboru `spk2utt` pri aplikácii CMN na príznaky pri trénovaní UBM a pre posteriórne pravdepodobnosti Gaussových zložiek a tiež v samotnom procese extrakcie.

Podľa dokumentácie a komentárov programov pre extrakciu i-vektorov v použitom recepte sú vektory namiesto jedného pre celú nahrávku alebo rečníka extrahované online. Pre každý desiaty rámec je vytvorený nový i-vektor. V určitom časovom bode je i-vektor extrahovaný z rámcov daného rečníka videných od začiatku po tento bod.

V tejto sekcii budeme experimentovať s definíciou „rečníka“, ďalej uvádzanou ako spk-id, za použitia datasetu popísaného v sekcii 5.6. Pokiaľ chceme v Kaldi manipulovať s identifikátormi nahrávok (utt-id) a rečníkov (spk-id), musíme zaistiť, že poradie nahrávok v súboroch `utt2spk` (podľa utt-id) a `spk2utt` (podľa utt-id a následne podľa spk-id) bude rovnaké. Je to požadované z dôvodu prúdového spracovania dát a ich predávania medzi procesmi pomocou rúr, kedy je dôležité, aby nahrávky z rôznych vstupov za sebou nasledovali v jasne definovanom poradí. Najjednoduchšie je táto podmienka splniteľná, pokiaľ spk-id je prefixom utt-id. To zaistí skript `prepare_lists`, používaný pri kontaminácii dát, bližšie popísaný v 5.1. Tento skript vytvorí pre účel experimentu tri alternatívne verzie `utt2spk` (opačné mapovanie je automaticky odvodené), ktoré z hľadiska dopadu na úspešnosť rozpoznávania systému ďalej porovnáme:

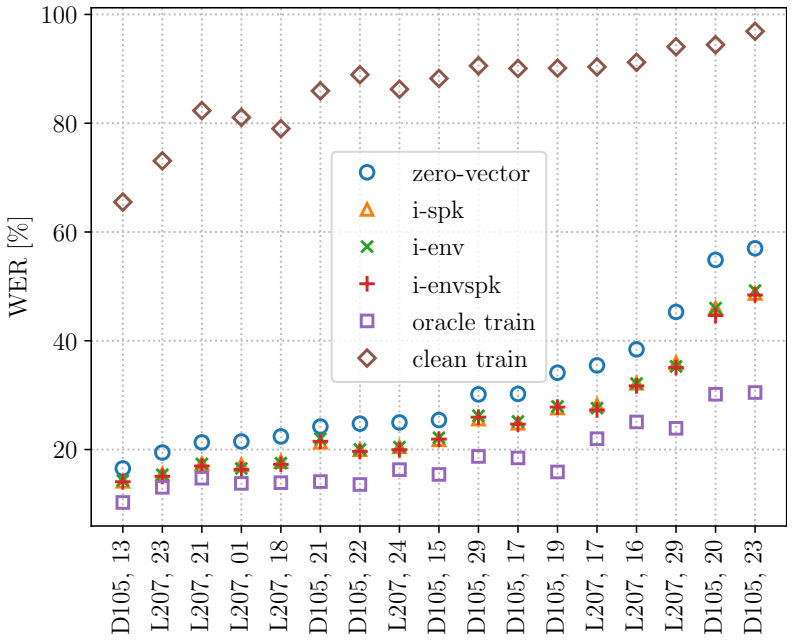
- Mapovanie *i-spk* – spk-id pozostáva z identifikátorov rečníka a kapitoly – toto mapovanie odpovedá pôvodnému prípravnému skriptu z Mini LibriSpeech. Keďže nie je obsiahnutá informácia o prostredí, ktorým bola nahrávka kontaminovaná, nadväzujú na seba nahrávky od jedného rečníka čítajúceho danú knižnú kapitolu, ale zo všeobecne rôznych prostredí.

- Mapovanie *i-env* – spk-id pozostáva z identifikátorov prostredia (miestnosť, pozícia reproduktora, kanál) – pri tomto mapovaní na seba nadväzujú rámce nahrávok kontaminovaných jedným prostredím, ale potenciálne od rôznych rečníkov.

- Mapovanie *i-envspk* – spk-id obsahuje identifikátory prostredia a rečníka. Pre účely CMN a extrakcie vektorov na seba nadväzujú iba nahrávky od daného rečníka z daného prostredia.

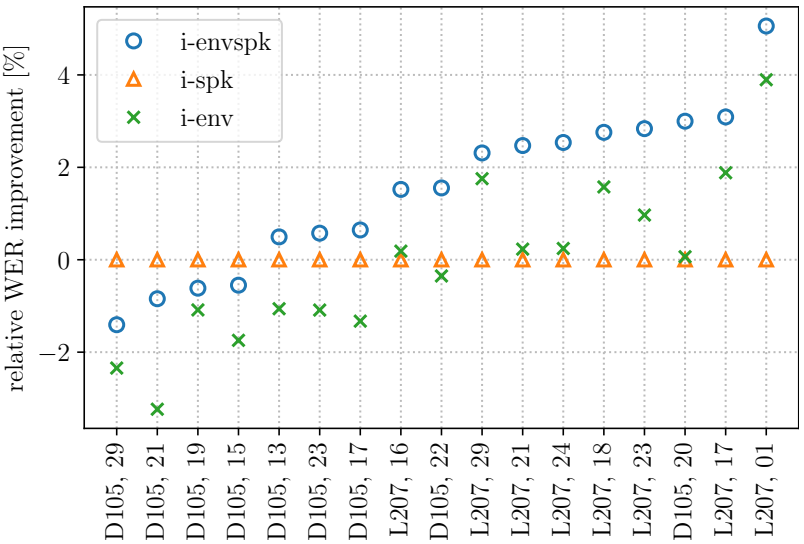
Spomenuté spôsoby boli porovnané v rámci modifikovaného receptu za využitia čistých dát pre zarovania (podľa experimentu v sekcii 5.5). Pre účel trénovania GMM systému čistými dátami bolo použité pôvodné mapovanie `utt2spk`, pretože u nekontaminovaných dát nemá zmysel zavádzať do identifikátorov informáciu o „prostredí“. To však spôsobuje nezhodu medzi identifikátormi nahrávok v archívoch so zarovnávacími zväzmi a kontaminovanými nahrávkami. Bolo potrebné upraviť spôsob výstupu skriptu `align_fmllr_lats.sh` z Kaldi a zároveň vstup `get_raw_egs.sh`, aby bolo medzi týmito dvomi krokmi možné modifikovať utt-id a zároveň meniť poradie nahrávok.

Výsledky experimentu sú porovnané na Obr. 5.10 s WER získanými v predchádzajúcich experimentoch a tiež oproti kontrolnému systému, ktorému na mieste i-vektorov boli na vstupe dodané pri trénovaní aj testovaní iba nulové vektory.



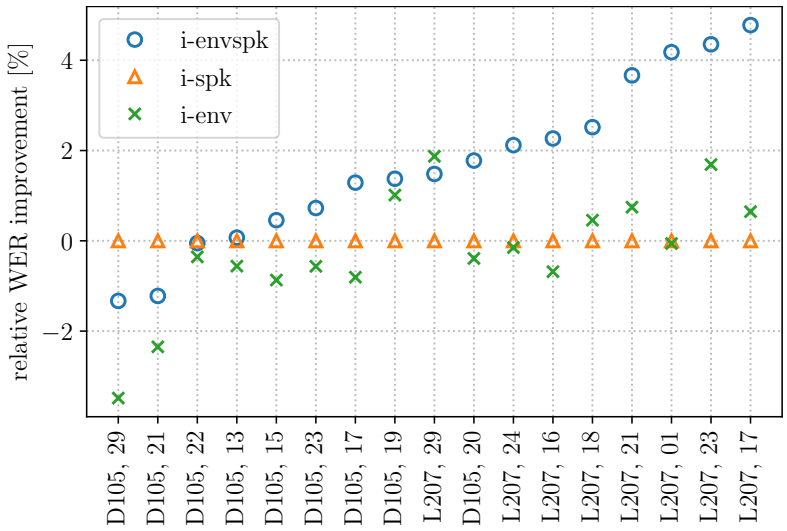
Obr. 5.10: Porovnanie rôznych variánt mapovania nahrávok a rečníkov pre i-vektory oproti baseline systému trénovanému na čistých dátach (clean-train), systému s nulovými „i-vektormi“ (zero-vector) a oproti výsledkom systémov trénovaných na cieľových podmienkach zo sekcie 5.5 (oracle train).

Pri zobrazenej škále na Obr. 5.10 vidno medzi rôznymi variantami trénovania a extrakcie i-vektorov veľmi malé rozdiely. Dá sa konštatovať, že systém trénovaný na zmiešaných podmienkach využívajúci i-vektory má výrazne navrch voči systému bez nich, zároveň však nedosahuje tak dobré výsledky ako systémy osobitne trénované pre cieľové prostredie. Pre podrobnejšie porovnanie i-vektorových variánt výsledky zobrazujeme na Obr. 5.11 aj ako relatívne zlepšenie WER u i-env a i-envspk (nové varianty) oproti i-spk (pôvodné mapovanie).



Obr. 5.11: Relatívne zlepšenie WER systémov s i-env a i-envspk oproti pôvodnému i-spk pre jednotlivé testovacie kanály.

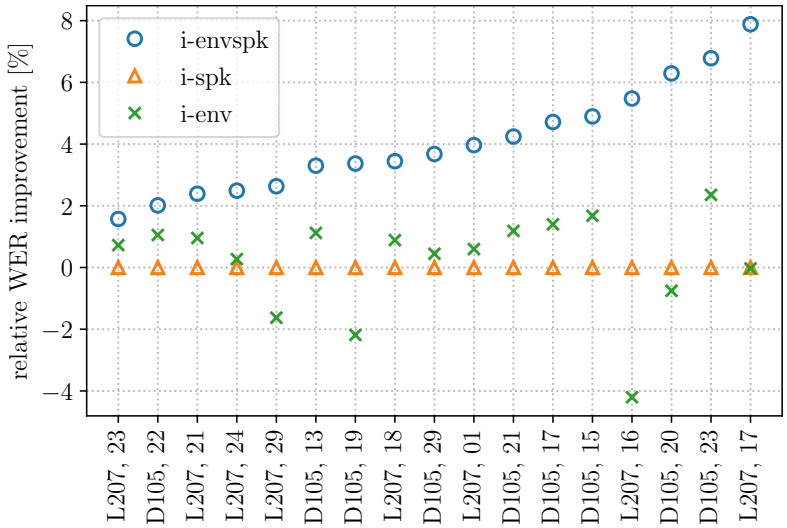
Keďže rozdiely medzi variantami sú relatívne tesné, bola vytvorená ďalšia verzia zmiešaného datasetu (kontaminovaná inou podmnožiou 500 prostredí) a na nej zopakované experimenty s trénovaním týchto troch systémov, ktorých výsledky sú na Obr. 5.12.



Obr. 5.12: Relatívne zlepšenie WER systémov s i-env a i-envspk oproti pôvodnému i-spk pre jednotlivé testovacie kanály. Výsledky sú zo systému s rovnakou architektúrou ako v 5.11, ale použitá bola iná verzia zmiešaného trénovacieho datasetu (iná množina 500 RIR).

Pri oboch opakovaniach pokusu vidíme u väčšiny testovacích kanálov mierne zlepšenie WER za použitia mapovania i-envspk oproti pôvodnému i-spk. Zaujímavé je tiež, že všeobecne väčší prínos tohto mapovania je u kanálov z L207, kým pre D105 táto metóda priniesla v určitých prípadoch mierne zhoršenie. Priemerné zlepšenie i-envspk je pri dvoch behoch pokusu 1.5 % a 1.7 %. Mapovanie i-env prinieslo mierne zhoršenie oproti i-spk (−0.1 % a −0.2 %).

Experiment so zmenou mapovania sme zopakovali aj pre pôvodný recept (použitie kontaminovaných dát pre trénovanie GMM a zarovnanie). Výsledky sú na Obr. 5.13.



Obr. 5.13: Relatívne zlepšenie WER systémov s mapovaním i-env a i-envspk oproti pôvodnému i-spk pri trénovaní nemodifikovaného systému Mini LibriSpeech, kde GMM časť videla iba kontaminovaný dataset (zároveň použitý pre generovanie zarovnania).

U systému bez použitia vylepšených zarovnaní vidíme vyšší prínos mapovania i-envspk ako pri predchádzajúcich systémoch – priemerné zlepšenie je v tomto prípade na úrovni 4.1 %. Mapovanie i-env ani u tohto systému neprináša výraznejšie zlepšenie, je na úrovni 0.2 %.

5.8 Použitie x-vektorov namiesto i-vektorov

Článok [12] pojednáva o možnosti využitia techniky x-vektorov ako alternatívneho a/alebo doplnkového vstupu k i-vektorom v rozpoznávačoch založených na neurónových sieťach. Pracuje s kontamináciou dát na zmiešané podmienky pomocou 30 000 umelo generovaných impulzných odoziev a x-vektorový podsystém sa učí klasifikovať nahrávky podľa IR, ktorou boli reverberované. Autori článku tento systém klasifikujúci IR nazývajú r-vektory. Bola dosiahnutá podobná úspešnosť ako pri použití i-vektorov a ďalšie zlepšenie pri použití oboch vektorov spoločne. V tejto sekcii ďalej experimentujeme s x-vektormi (adaptácia na rečníka) aj r-vektormi (adaptácia na prostredie).

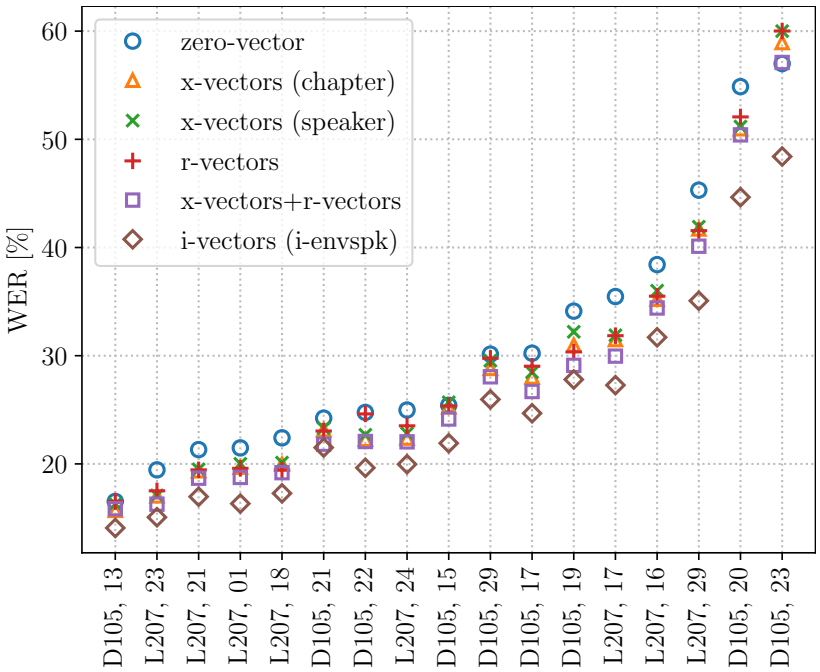
R-vektorový systém z článku [12] pracuje s originálnou architektúrou neurónovej siete extraktora x-vektorov popísanou v [31] – prvé tri frame-level vrstvy postupne rozširujú časový kontext (time-delay), nasledujú dve obyčajné vrstvy, pooling vrstva, za ňou 2 segment-level vrstvy a výstupná softmax vrstva. V tomto experimente pracujeme s modifikovaným systémom dodaným pánom Ing. Martinom Karafiátom, Ph.D., kde namiesto prvých troch time-delay vrstiev sú vo frame-level časti striedavo radené 4 time-delay a 4 obyčajné vrstvy. Výsledný časový kontext, s ktorým sa pracuje vo frame-level časti, je v tejto verzii širší ako v pôvodnej verzii. Táto architektúra bola použitá napríklad v [32].

Dodaný recept pre prácu s x-vektormi pôvodne trénuje neurónovú sieť na klasifikáciu rečníkov z iných rečových korpusov a zahŕňa kontamináciu dát inými impulznými odozvami. Tieto časti boli zmazané a recept bol upravený pre použitie nami vygenerovaných dát podľa sekcie 5.6. Pre zaistenie kompatibility však musel byť tento dataset podvzorkovaný z pôvodných 16 kHz na 8 kHz, z čoho sú ďalej extrahované 23 dimenzionálne MFCC príznaky (i-vektorový systém pracoval s 40 dimenzionálnymi MFCC). Namiesto online extrakcie ako u i-vektorov, je x-vektor vytvorený iba jeden pre celú nahrávku. Interne je do archívov ďalej rozkopírovaný pre každý desiaty rečový rámec nahrávky, aby mohol byť výstup tohto upraveného receptu zameniteľný za i-vektory v TDNN rozpoznávači reči z Mini LibriSpeech.

Na rozdiel od článku [12] pracujeme s kontamináciou pomocou reálnych impulzných odoziev. Použitý dataset je pre spravodlivé porovnanie zhodný s predchádzajúcim experimentom. Ako supervízia pre trénovanie x/r-vektorovej neurónovej siete sú použité mapovania i-spk, a i-env z experimentu v sekcii 5.7. Podobne ako i-vektory použité v recepte Mini LibriSpeech, aj x/r-vektory boli trénované na dátach po aplikovanej trojcestnej rýchlostnej perturbácii. Pridané zrýchlené a spomalené kópie nahrávok zavádzajú do mapovania odpovedajúce kópie spk-id. Výsledné vektory sú extrahované ako výstup prvej z dvoch segment-level vrstiev, ktorej veľkosť bola v zhode s dimenziou i-vektorov nastavená na 100. Natrénované boli 3 extraktory podľa mapovaní:

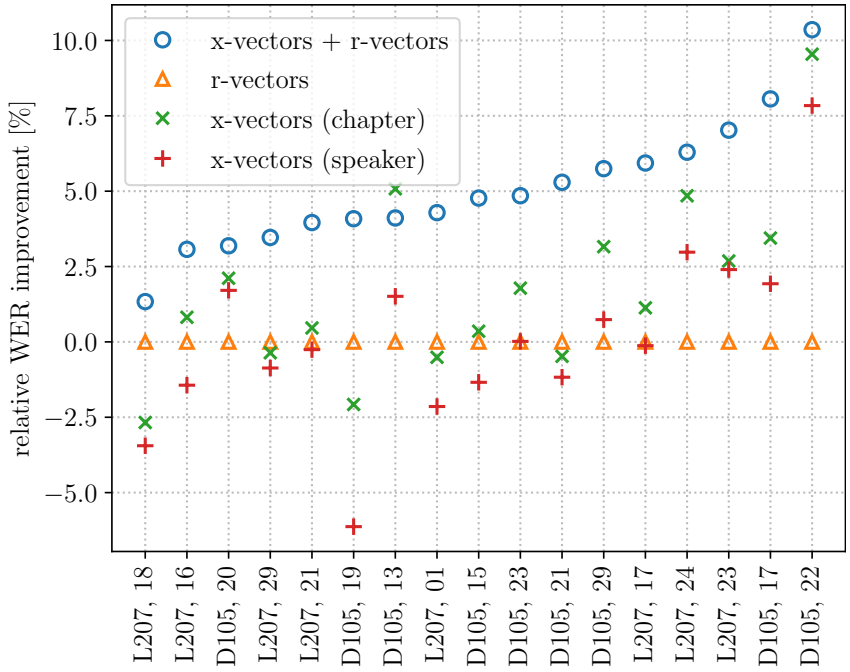
- x-vektory klasifikujúce 3×581 rečníko-kapitol od 251 unikátnych rečníkov – pôvodné mapovanie z receptu Mini LibriSpeech, v predchádzajúcom experimente zvané i-spk,
- x-vektory klasifikujúce 3×251 unikátnych rečníkov – bez informácie o kapitolách (neodpovedá žiadnemu mapovaniu z experimentu s i-vektormi),
- r-vektory klasifikujúce 3×500 prostredí (mapovanie i-env).

Vektory vytvorené troma uvedenými extraktormi boli použité ako vstup TDNN namiesto i-vektorov. Okrem týchto troch systémov bol experiment rozšírený o použitie konkatenovaného r-vektora a úspešnejšieho z dvojice x-vektorov (dokopy 200 dimenzií). Výsledky vyjadrené ako WER dosiahnuté výslednými rozpoznávacími systémami sú zhrnuté na Obr. 5.14. Hodnoty podobne ako na Obr. 5.10 porovnávané s kontrolným systémom s nulovými vektormi a v tomto prípade tiež s variantou i-vektorového systému, ktorá vyšla z predchádzajúceho experimentu najúspešnejšie.



Obr. 5.14: Porovnanie úspešnosti rozpoznávačov používajúcich rôzne varianty x/r-vektorov namiesto i-vektorov oproti systému s nulovými vektormi (zero-vector) a najlepšou i-vektorovou variantou.

Podobne ako u i-vektorov, aj rôzne varianty samostatných x/r-vektorov vykazujú veľmi tesné výsledky pri rozpoznávaní. Kým v článku [12] boli pozorované podobné úspešnosti pri použití samotných r-vektorov a samotných i-vektorov, naše experimenty ukazujú nižšiu úspešnosť r-vektorov. Ani spojenie x-vektorov a r-vektorov neprináša lepšie výsledky oproti samotným i-vektorom. Je však potrebné pripomenúť, že pracujeme s modifikovanou architektúrou DNN a reálnymi IR namiesto umelých. Podrobnejšie porovnanie jednotlivých variánt x/r-vektorov je na Obr. 5.15.



Obr. 5.15: Podrobnejšie porovnanie variánt x/r-vektorov použitých v rozpoznávači reči vyjadrených ako relatívne rozdiel WER oproti r-vektorovej variante.

Z dvojice x-vektorov je mierne úspešnejšia varianta, kde bol extraktor trénovaný na klasifikáciu rečníko-kapitol oproti klasifikátoru unikátnych rečníkov (priemerný relatívny rozdiel rovný 1.6 %). Lepší z x-vektorov vychádza o trochu úspešnejšie aj v porovnaní s r-vektormi (priemerne o 1.8 %), nie však konzistentne naprieč všetkými kanálmi. Najúspešnejšia z porovnaných systémov je kombinácia r-vektorov s x-vektormi (verzia klasifikujúca rečníko-kapitoly).

Autori [12] pracovali s 512-dimenzionálnymi r-vektormi, boli preto zopakované niektoré experimenty za použitia tejto šírky a porovnané so 100-dimenzionálnymi x/r-vektormi. Výsledky agregované ako priemer WER naprieč testovacími kanálmi sú v Tabuľke 5.3. V našom prípade sa javia ako mierne výkonnejšie x-vektory o dimenzii rovnej 100, r-vektory a spojenie oboch vektorov dosahujú približne rovnaké výsledky.

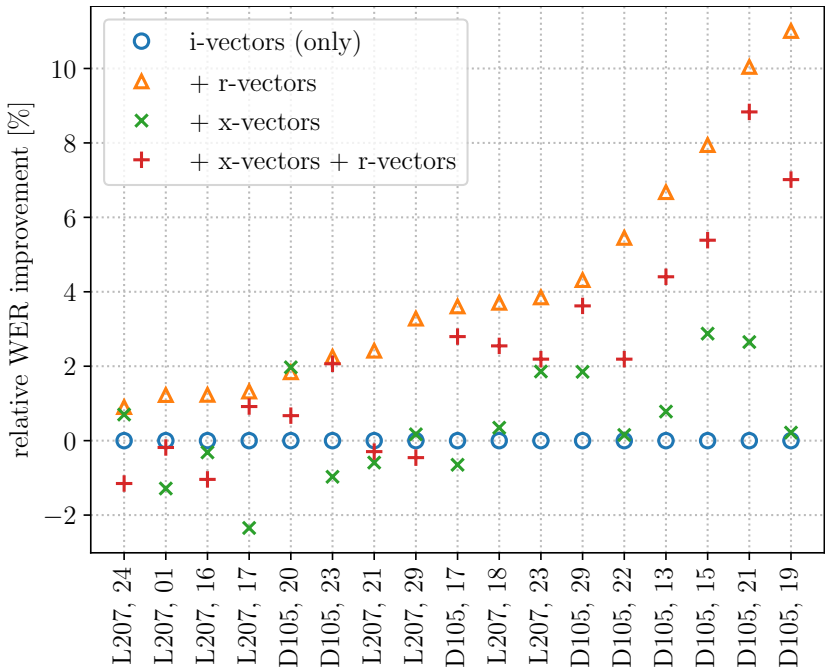
	100 dim.	512 dim.	ΔWER, rel.
x-vectors	27.92 %	28.74 %	−2.83 %
r-vectors	29.37 %	29.33 %	0.12 %
x-vectors + r-vectors	28.88 %	29.00 %	−0.41 %

Tabuľka 5.3: Porovnanie priemerných WER rozpoznávačov využívajúcich 100 a 512 dimenzionálne x-vektory a r-vektory.

5.9 Kombinácia i-vektorov, x-vektorov a r-vektorov

V článku [12] boli prezentované výsledky, podľa ktorých sú i-vektory a r-vektory komplementárne pri využití v rozpoznávači reči, teda ich spoločné využitie prináša ďalšie zlepšenie úspešnosti rozpoznávania oproti použitiu samotnej jednej alebo druhej metódy. V tomto experimente hľadáme najlepšie fungujúcu kombináciu týchto vektorov.

Vychádzame z i-vektorového systému s mapovaním i-envspk, ktoré vyšlo v experimentoch zo sekcie 5.7 ako najúspešnejšie. Z predchádzajúcej sekcie 5.8 využívame extraktor r-vektorov a úspešnejší z dvojice x-vektorových podsystémov (ktorý sa učil klasifikovať rečníko-kapitoly). Postupne boli natrénované rozpoznávače využívajúce r-vektory skonkatenované s i-vektormi, podobne x-vektory s i-vektormi a na koniec bola vyskúšaná konkatenácia všetkých troch typov adaptačných vektorov. Výsledky sú na Obr. 5.16 vyjadrené ako relatívne zlepšenie WER oproti samotným i-vektorom.



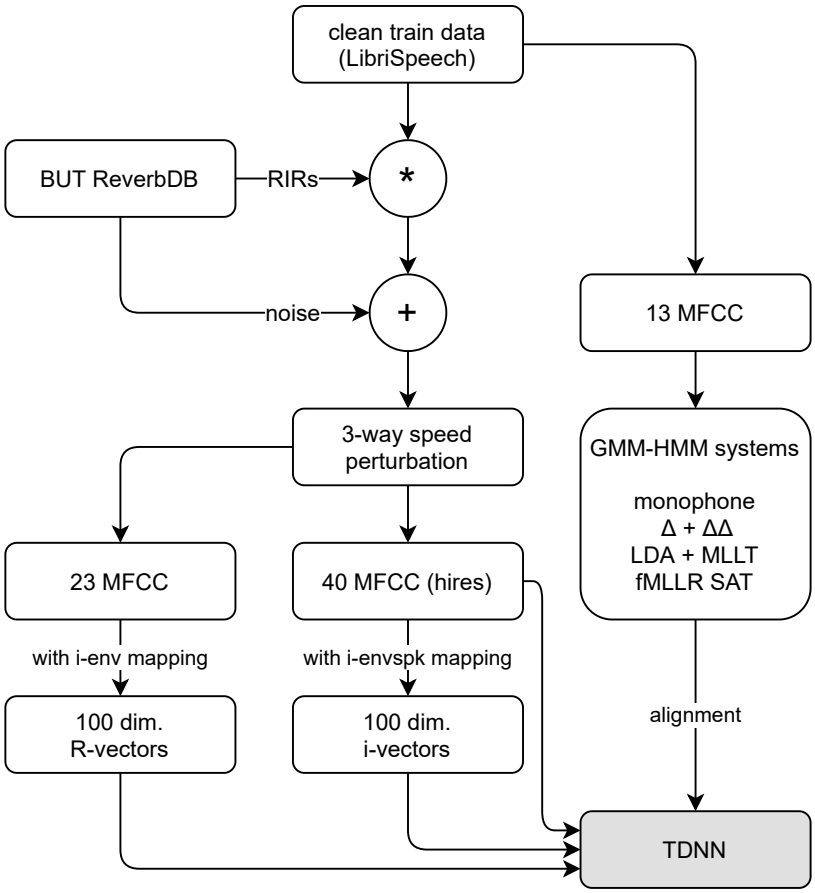
Obr. 5.16: Porovnanie relatívneho rozdielu WER rôznych systémov využívajúcich x-vektorové a r-vektorové extraktory ako doplnok ku i-vektorom oproti základnému systému so samotnými i-vektormi.

Kým v systémoch bez i-vektorov vychádzali r-vektory ako najmenej účinné z hľadiska WER, v kombinácii s i-vektormi naopak ako jediné vykazujú zlepšenie u všetkých kanálov, priemerne na úrovni 4.2 % oproti samotným i-vektorom. Môžeme teda potvrdiť zistenia z [12], že táto technika adaptácie je s i-vektormi komplementárna, a to aj pri inej architektúre extraktora, inej dimenzii r-vektorov a za použitia reálnych prostredí pre augmentáciu dát. Zaujimavé je, že pokiaľ k tejto dvojici ďalej pridáme x-vektory (a neuberieme žiadnu informáciu), dosiahneme u všetkých kanálov horšie výsledky a relatívne zlepšenie oproti samotným i-vektorom je iba 2.3 %. Pridanie x-vektorov bez r-vektorov prináša rozdiel iba 0.4 % a u 6 zo 17 testovacích kanálov prináša zhoršenie.

Pre úplnosť bol natrénovaný aj systém využívajúci i-vektory spolu s 512 dimenzionálnymi r-vektormi. Oproti rovnakej konfigurácii so 100 dimenzionálnymi vektormi vykazuje relatívne zhoršenie 2.4 %.

Najlepší systém z hľadiska robustnosti voči rôznym akustickým podmienkam, ktorý sa v rámci tejto práce podarilo zrealizovať využíva zarovnanie pomocou čistých nahrávok a na vstupe má 40 dimenzionálne MFCC príznaky z kontaminovaných nahrávok, i-vektory s upraveným mapovaním i-envspk a r-vektory o dimenzii 100. Oproti systému s pôvodnou architektúrou (ktorý by bol trénovaný na zmiešaných podmienkach) bolo týmito úpravami

dosiahnuté relatívne zlepšenie WER o 15.79 %. Výsledná schéma tohto systému je na Obr. 5.17. Pre porovnanie uvádzame Tabuľku 5.4 s porovnaním s ostatnými významnými systémami z vykonaných experimentov.



Obr. 5.17: Schéma najúspešnejšieho systému z hľadiska robustnosti voči rôznym prostrediam, ktorý sa podarilo zrealizovať v tejto práci.

systém/modifikácia	WER
originálny recept trénovaný na čistej reči	86.35 %
po kontaminácii trénovacích dát na zmiešané podmienky	28.75 %
naviac po použití zarovnania vytvoreného pre čistú reč	25.20 %
naviac po pridaní r-vektorov	**24.21 %**
po kontaminácii dát na zhodné podmienky, bez r-vektorov (oracle systém)	18.23 %

Tabuľka 5.4: Porovnanie priemerného WER (zo 17 testovacích kanálov) najúspešnejšieho robustného rozpoznávača s významnými systémami z ostatných experimentov.

5.10 Počet replikácií nahrávok a počet prostredí

V predchádzajúcich sekciách sme pracovali vždy so 100 hodinovou podmnožinou korpusu LibriSpeech pre možnosť férového porovnania výsledkov naprieč experimentami. Pokiaľ však kontaminujeme čistú reč pre zmiešané podmienky, je možné každú nahrávku kontaminovať

viackrát (zakaždým inou impulznou odozvou a šumom). Okrem toho sme pracovali so zafixovaným datasetom, kde bola každá nahrávka kontaminovaná jedným z 500 vybraných prostredí. Pokiaľ však máme k dispozícii impulzné odozvy a šum z viac prostredí, môžeme pri kontaminácii umožniť širší výber ako len 500.

V tomto experimente skúšame trénovať rozpoznávač na datasete rozšírenom na 300, 500 a 700 hodín a tiež skúšame rozšíriť výber možných kontaminačných prostredí z 500 na 1 000. Pre datasety generované s 1 000 prostrediami bola množina možných prostredí z ReverbDB rozšírená o korpus ACE[2], pretože v samotnom ReverbDB nám po odfiltrovaní miestností D105, L207, L227 a obmedzení výberu kanálov 1 až 8 z guľového mikrofónneho poľa (ktoré sú si veľmi podobné) ostane iba 696 prostredí. Výsledky pokusov sú zhrnuté v Tabuľke 5.5. Na tento pokus bol vybraný najúspešnejší zo systémov z predchádzajúcich experimentov – používajúci kombináciu i-vektorov a r-vektorov.

počet trénovacích hodín	počet RIR	
	500	1 000
100	24.21 %	24.57 %
300	21.99 %	22.51 %
500	21.62 %	21.79 %
700	21.60 %	21.60 %

Tabuľka 5.5: Porovnanie WER rozpoznávača trénovaného na datasetoch s rôznym počtom replikácií nahrávok a impulzných odoziev na dostupných pri kontaminácii.

Vidíme, že pre rôzne dĺžky datasetov je optimálny počet impulzných odoziev rozdielny. Príliš veľa impulzných odoziev pre málo hodín trénovacích dát vedie k nižšej úspešnosti. Pre 7 replikácií sa výsledky za použitia 1 000 oproti 500 impulzným odozvám vyrovnávajú.

[2] http://www.ee.ic.ac.uk/naylor/ACEweb/index.html

Kapitola 6

Záver

V teoretickej časti boli predstavené javy spôsobujúce degradáciu úspešnosti rozpoznávania reči, spôsoby, akými sa rozpoznávacie systémy s týmito javmi vyrovnávajú, prístupy k získavaniu impulzných odoziev miestností s dôrazom na použitú metódu ESS pre reálne IR, metriky pre vyjadrenie množstva reverberácie v nahrávkach a v neposlednom rade prehľad prác zaoberajúcich sa problematikou skúmania vplyvov akustiky na rozpoznávanie reči a možným zmiernením problémov spôsobených akustickými javmi. Ďalšie kapitoly oboznamujú čitateľa s použitými dátovými setmi, rozpoznávacím systémom a procesom jeho trénovania a spôsobmi využitia technológií z oblasti rozpoznávania *rečníka* pri adaptácii rozpoznávača *reči* na prostredie a/alebo rečníka.

Experimentálna časť najskôr predstavila implementáciu kontaminácie dát, s ktorými bolo následne v práci manipulované. Nultým experimentom bolo overenie tejto implementácie s dôrazom na primiešanie šumu do reverberovaných nahrávok pri správnom odstupe signálu od šumu. Ďalej bola predstavená množina testovacích kanálov pozostávajúca z kancelárie L207 a prednáškovej auly D105 používaná naprieč experimentami pre vyhodnotenie a porovnanie úspešnosti.

Prvý experiment bol vykonaný s pôvodným receptom, kde bol systém opakovane trénovaný na dátach z jediného kanála a vyhodnotený na všetkých testovacích kanáloch jeho miestnosti (vrátane toho, na ktorom bol trénovaný). Počas trénovania GMM aj TDNN časti boli použité iba kontaminované dáta. Bolo zistené, že modely trénované na odpovedajúcich podmienkach sú úspešnejšie ako modely trénované s inými podmienkami. Schopnosť rozpoznávacieho systému naučiť sa na dané podmienky sa znižovala so vzdialenosťou mikrofónu. U dvojíc mikrofónov s podobnou pozíciou v polárnych súradniciach, kde jeden mikrofón bol uzavretý a druhý mal priamu viditeľnosť na reproduktor dosahovali lepšie výsledky viditeľné mikrofóny. Úspešnosť sa znižovala aj s klesajúcim SNR, ELR a DRR, pričom v prípade ELR bol tento vzťah badateľnejší ako pri DRR.

Následne bol recept upravený tak, aby pri trénovaní GMM časti použil čistú verziu datasetu a vytvoril pomocou nej zarovnanie rámcov na stavy HMM ďalej použité ako supervíziu pre trénovanie neurónovej siete na príznakoch z kontaminovanej verzie dát. Rozpoznávače boli znova natrénované a vyhodnotené na všetkých vybraných kanáloch. Tento experiment priniesol badateľne vyššie zlepšenie pre kanály z D105 oproti kanálom z L207, pričom sa presnú príčinu nepodarilo zistiť. Výsledky tohto pokusu vykazujú najnižšiu nami dosiahnutú chybovosť pre všetky testovacie kanály. Systém síce nie je veľmi praktický (pretože pre iné podmienky by bolo potrebné pretrénovať neurónovú sieť), ale poskytoval referenčné hodnoty WER (v práci nazývané aj „oracle train“), ku ktorým sme sa snažili

priblížiť v experimentoch s trénovaním so zmiešanými podmienkami a adaptáciou na prostredie.

V časti, kde sme sa snažili vytvoriť čo najrobustnejší rozpoznávač voči rôznym podmienkam (trénovaním na dátach zo zmesi rôznych prostredí), sme najskôr experimentovali s mapovaním nahrávok na rečníkov pre trénovanie a extrakciu i-vektorov. Ukázalo sa, že rozdelenie nahrávok jedného rečníka podľa prostredia, ktorým bola daná nahrávka kontaminovaná má mierny prínos pre úspešnosť (relatívne zlepšenie o 1.5 %). Ďalej sme skúšali nahradiť i-vektory za x-vektory a/alebo r-vektory (založené na separátnej neurónovej sieti, ktorá sa učí klasifikovať nahrávky podľa rečníkov, resp. prostredí), pričom sa z rôznych variánt ukázala ako najlepšia práve kombinácia x-vektorov a r-vektorov, ale ani spolu nedosahovali takú výkonnosť ako samotné i-vektory. Na koniec bola vyskúšaná kombinácia i-vektorov s x/r-vektormi, pričom najlepšie vyšla dvojica i-vektorov s r-vektormi, vykazujúc relatívne zlepšenie o 4.2 % oproti samotným i-vektorom. Podarilo sa teda ukázať prínos r-vektorov aj v kombinácii s trénovaním reálnymi impulznými odozvami (oproti pôvodnému článku využívajúcemu syntetické IR).

Modifikáciami receptu sa podarilo dosiahnuť zvýšenie úspešnosti rozpoznávania oproti pôvodnej verzii Mini LibriSpeech (ak by bola iba trénovaná na zmiešaných podmienkach, bez iných úprav). Znásobenie pôvodného 100 hodinového trénovacieho datasetu na 500 hodín prinieslo ďalšie zníženie odstupu od úspešnosti oracle systému, stále je však priestor na vylepšenia.

V práci by sa ďalej dalo pokračovať aplikovaním ďalších techník, ktoré sa ukázali v niektorých publikovaných článkoch ako prínosné, napríklad predtrénovanie neurónovej siete na príznakoch z čistých dát a postupné pridávanie kontaminovaných dát (tzv. curricculum learning) alebo použitie kombinácie reálnych a umelých impulzných odoziev pre kontamináciu dát. Kvalita supervízie pre TDNN rozpoznávač by mohla byť potenciálne zvýšená natrénovaním najskôr kompletného systému vrátane TDNN na čistých dátach a jeho využitie pre zarovnanie dát (v tejto práci sme používali iba GMM systém s čistými dátami).

Systémy pre extrakciu x/r-vektorov sa učili klasifikovať nahrávky buď iba podľa rečníka alebo iba podľa prostredia (na výstupnej vrstve „one hot“), zaujímavé by však bolo vyskúšať trénovať neurónovú sieť extraktora na klasifikáciu oboch vecí naraz (multi-task training). Pri trénovaní rozpoznávača neboli zohľadnené akustické parametre impulznej odozvy (DRR, ELR). Stálo by za to vyskúšať, či by ich odhad z nahrávky reči ďalej použitý na vstupe rozpoznávača nemal ďalší prínos pre úspešnosť.

Literatúra

[1] Allen, J. B. a Berkley, D. A. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*. Apríl 1979, zv. 65, č. 4, s. 943 – 950. ISSN 1520-8524.

[2] Balakrishnama, S., Ganapathiraju, A. a Picone, J. Linear discriminant analysis for signal processing problems. In: *IEEE Southeast Con*. Február 1999, s. 78 – 81. DOI: 10.1109/SECON.1999.766096. ISBN 0-7803-5237-8.

[3] Braun, S., Neil, D. a Liu, S.-C. A Curriculum Learning Method for Improved Noise Robustness in Automatic Speech Recognition. In: *EURASIP*. September 2017. DOI: 10.23919/EUSIPCO.2017.8081267. ISBN 978-0-9928626-7-1.

[4] Brutti, A. a Matassoni, M. On the use of Early-To-Late Reverberation ratio for ASR in reverberant environments. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, s. 4638–4642. DOI: 10.1109/ICASSP.2014.6854481. ISBN 9781479928927.

[5] Cabrera, D., Xun, J. a Guski, M. Calculating Reverberation Time from Impulse Responses: A Comparison of Software Implementations. *Acoustics Australia*. August 2016, zv. 44, č. 2, s. 369–378. DOI: 10.1007/s40857-016-0055-6. ISSN 1839-2571.

[6] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P. a Ouellet, P. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*. 2011, zv. 19, č. 4, s. 788–798. DOI: 10.1109/TASL.2010.2064307.

[7] Dimitriadis, D., Thomas, S. a Ganapathy, S. An Investigation on the Use of i-Vectors for Robust ASR. In: *INTERSPEECH*. September 2016, s. 3828–3832. DOI: 10.21437/Interspeech.2016-1482. ISBN 978-1-5108-3313-5.

[8] Emmanouilidou, D. a Gamper, H. The effect of room acoustics on audio event classification. In: *Proceedings of the 23rd International Congress on Acoustics*. September 2019, s. 102–109. ISBN 978-3-939296-15-7.

[9] Farina, A. Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique. In: *Audio Engineering Society Convention 108*. Február 2000, s. 1–24.

[10] Gales, M. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language*. 1998, zv. 12, č. 2, s. 75 – 98. DOI: https://doi.org/10.1006/csla.1998.0043. ISSN 0885-2308.

[11] Gopinath, R. A. Maximum likelihood modeling with Gaussian distributions for classification. In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*. 1998, sv. 2, s. 661–664 vol.2. DOI: 10.1109/ICASSP.1998.675351.

[12] Khokhlov, Y., Zatvornitskiy, A., Medennikov, I., Sorokin, I., Prisyach, T. et al. R-Vectors: New Technique for Adaptation to Room Acoustics. In: *INTERSPEECH*. ISCA, September 2019, s. 1243–1247. DOI: 10.21437/Interspeech.2019-2645. ISBN 978-1-5108-9683-3.

[13] Ko, T., Peddinti, V., Povey, D., Seltzer, M. L. a Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, s. 5220–5224. DOI: 10.1109/ICASSP.2017.7953152.

[14] Ko, T., Peddinti, V., Povey, D. a Khudanpur, S. Audio augmentation for speech recognition. In: *INTERSPEECH*. ISCA, 2015, s. 3586–3589. ISBN 978-1-5108-1790-6.

[15] Kuttruff, H. *Room Acoustics*. 5. vyd. Abingdon: Spon Press, 2009. 223 s. ISBN 978-0-415-48021-5.

[16] Melot, J., Malyska, N., Ray, J. a Shen, W. Analysis of factors affecting system performance in the ASpIRE challenge. In: *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2015, s. 512–517. DOI: 10.1109/ASRU.2015.7404838.

[17] Miao, Y., Zhang, H. a Metze, F. Speaker Adaptive Training of Deep Neural Network Acoustic Models Using I-Vectors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2015, zv. 23, č. 11, s. 1938–1949. DOI: 10.1109/TASLP.2015.2457612.

[18] Nishiura, T., Hirano, Y., Denda, Y. a Nakayama, M. Investigations into early and late reflections on distant-talking speech recognition toward suitable reverberation criteria. In: *INTERSPEECH*. Január 2007, sv. 2, s. 1082–1085. ISBN 978-1-60560-316-2.

[19] Paliesek, J. *Změření vlivu akustiky prostředí na úspěšnost rozpoznávače řeči*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedúci práce Ing. Igor Szőke, P.

[20] Panayotov, V., Chen, G., Povey, D. a Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, s. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[21] Peddinti, V., Chen, G., Povey, D. a Khudanpur, S. Reverberation robust acoustic modeling using i-vectors with time delay neural networks. In: *INTERSPEECH*. ISCA, 2015, s. 2440–2444. ISBN 978-1-5108-1790-6.

[22] Peddinti, V., Manohar, V., Wang, Y., Povey, D. a Khudanpur, S. Far-Field ASR Without Parallel Data. In: *INTERSPEECH*. ISCA, September 2016, s. 1996–2000. DOI: 10.21437/Interspeech.2016-1475. ISBN 978-1-5108-3313-5.

[23] Peddinti, V., Povey, D. a Khudanpur, S. A time delay neural network architecture for efficient modeling of long temporal contexts. In: *INTERSPEECH*. ISCA, 2015, s. 3214–3218. ISBN 978-1-5108-1790-6.

[24] Povey, D. *Kaldi: Glossary of terms* [online]. [cit. 2020-12-16]. Dostupné z: https://kaldi-asr.org/doc/glossary.html.

[25] Povey, D. *Parallelization in Kaldi* [online]. [cit. 2020-12-13]. Dostupné z: https://kaldi-asr.org/doc/queue.html.

[26] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O. et al. The Kaldi Speech Recognition Toolkit. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. ISBN 978-1-4673-0365-1.

[27] Rath, S., Povey, D., Veselý, K. a Cernocky, J. Improved feature processing for deep neural networks. In: *INTERSPEECH*. ISCA, Január 2013, s. 109–113. ISBN 978-1-62993-443-3.

[28] Ravanelli, M. a Omologo, M. *Contaminated speech training methods for robust DNN-HMM distant speech recognition*. 2017.

[29] Rouvier, M. a Favre, B. Speaker adaptation of DNN-based ASR with i-vectors: Does it actually adapt models to speakers? In: *INTERSPEECH*. Január 2014, s. 3007–3011. ISBN 978-1-63439-435-2.

[30] Sehr, A., Habets, E., Maas, R. a Kellermann, W. Towards a better understanding of the effect of reverberation on speech recognition performance. In: *Proc. Intl. Workshop Acoust. Echo Noise Control (IWAENC)*. August 2010.

[31] Snyder, D., Garcia Romero, D., Povey, D. a Khudanpur, S. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In: *INTERSPEECH*. August 2017, s. 999–1003. DOI: 10.21437/Interspeech.2017-620. ISBN 978-1-5108-4876-4.

[32] Snyder, D., Garcia Romero, D., Sell, G., McCree, A., Povey, D. et al. Speaker Recognition for Multi-speaker Conversations Using X-vectors. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, s. 5796–5800. DOI: 10.1109/ICASSP.2019.8683760. ISBN 978-1-4799-8132-8.

[33] Stan, G. B., Embrechts, J. J. a Archambeau, D. Comparison of Different Impulse Response Measurement Techniques. *J. Audio Eng. Soc.* 2002, zv. 50, č. 4, s. 249–262.

[34] Szöke, I., Skácel, M., Mošner, L., Paliesek, J. a Černocký, J. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*. 2019, zv. 13, č. 4, s. 863–876. DOI: 10.1109/JSTSP.2019.2917582.

[35] Tang, Z., Meng, H. a Manocha, D. Low-Frequency Compensated Synthetic Impulse Responses For Improved Far-Field Speech Recognition. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, s. 6974–6978. DOI: 10.1109/ICASSP40776.2020.9054454. ISBN 978-1-5090-6632-2.

[36] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. a Lang, K. J. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1989, zv. 37, č. 3, s. 328–339. DOI: 10.1109/29.21701.

[37] Yoshioka, T., Sehr, A., Delcroix, M., Kinoshita, K., Maas, R. et al. Making Machines Understand Us in Reverberant Rooms: Robustness Against Reverberation for Automatic Speech Recognition. *IEEE Signal Processing Magazine*. 2012, zv. 29, č. 6, s. 114–126. DOI: 10.1109/MSP.2012.2205029.

[38] Zahorik, P. Direct-to-reverberant energy ratio sensitivity. *The Journal of the Acoustical Society of America*. December 2002, zv. 112, s. 2110–7. DOI: 10.1121/1.1506692.

Príloha A

Obsah DVD

Na priloženom disku sa nachádza nasledujúci adresárový strom (zjednodušené):

```
/
├── report ..... zdrojové súbory technickej správy v LaTeX
├── results ..... výsledky experimentov ako surový výstup skriptu RESULTS
│   ├── oracle-no-forced-aligment ..... výsledky experimentov zo sekcie 5.4
│   ├── oracle-with-forced-aligment ..... výsledky experimentov zo sekcie 5.5
│   ├── ivec-variants ..... výsledky experimentov zo sekcie 5.7
│   ├── xvec-variants ..... výsledky experimentov zo sekcie 5.8
│   ├── ivec+xvec ..... výsledky experimentov zo sekcie 5.9
│   └── nrepli+nrirs ..... výsledky experimentov zo sekcie 5.10
├── recipe ..... úpravy receptu Mini LibriSpeech vykonané v tejto práci
├── README.md .. návod pre sprevádzkovanie dodaných úprav receptu Mini LibriSpeech
├── poster.pdf ..... plagát s výsledkami práce
└── video.mp4 ..... video s výsledkami práce
```

Adresár `recipe` obsahuje aj samostatne použiteľné skripty popísané v sekcii 5.1 určené na umelú kontamináciu dát v SGE:

- `collect_vad` – výpočet segmentov nahrávok obsahujúcich reč
- `calculate_snr` – Výpočet SNRp nahrávok za použitia VAD
- `prepare_lists` – rozdelenie úlohy kontaminácie na menšie časti ďalej spustiteľné pomocou `process_list` na jednotlivých uzloch
- `process_list` – implementácia kontaminácie spúšťaná na uzloch
- `submit_reverb` – prepája funkcionalitu vyššie uvedených skriptov (vhodné je použiť práve tento skript namiesto spúšťania jednotlivých krokov)
- `ir_metrics.py` – výpočet ELR a DRR z danej impulznej odozvy

Bližšie informácie o spúšťaní sú v `README.md`, alebo v hlavičkách týchto skriptov.