



**MATEMATICKO-FYZIKÁLNÍ  
FAKULTA**  
Univerzita Karlova

**DISERTAČNÍ PRÁCE**

Jan Oldřich Krůza

**Iterativní zdokonalování přepisu  
zvukových nahrávek s využitím zpětné  
vazby posluchačů**

Ústav formální a aplikované lingvistiky

Vedoucí disertační práce: Doc. RNDr. Vladislav Kuboň,  
Ph.D.

Studijní program: informatika

Studijní obor: matematická lingvistika

Praha 2021

Prohlašuji, že jsem tuto disertační práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Název práce: Iterativní zdokonalování přepisu zvukových nahrávek s využitím zpětné vazby posluchačů

Autor: Jan Oldřich Krůza

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí disertační práce: Doc. RNDr. Vladislav Kuboň, Ph.D., Ústav formální a aplikované lingvistiky

Konzultant: Mgr. Nino Peterek, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Tato disertační práce se zabývá zpřístupněním zvukových záznamů jednoho mluvčího úzké i široké veřejnosti.

Motivací práce byla existence chátrajících nahrávek hovorů českého filozofa ing. Karla Makoně na kazetách a kotoučích. Cílem je zachování materiálu pro budoucí generace a zpřístupnění nahrávek pomocí digitálních technologií, především přístupnosti nahrávek na internetu a možnosti vyhledávání v nich.

Práce představuje tvorbu systému pro přepis velké sady zvukových záznamů se zapojením laické komunity. Navržené řešení spočívá ve vytvoření základního přepisu nízké kvality pomocí automatického rozpoznávání řeči a vyvinutí aplikace, která umožní od členů komunity i nahodilých zájemců získávat opravy automatického přepisu, použitelné jako trénovací data pro další zlepšování.

Popíše se samotný mluvený korpus. Představí se autor a jeho dílo, témata v nahrávkách, nahrávání samotné, digitalizace a získané přepisy. Dále se rozvede tvorba systému pro automatický přepis korpusu od sběru dat přes akustické a jazykové modelování, různé provedené experimenty až k vyhodnocení úspěšnosti. V neposlední řadě se popíše webová aplikace pro sběr manuálních přepisů. Zmíní se odlišnosti od ostatních systémů, detaily návrhu a řešení, mechanismus pro kompenzaci vysokých nároků na kvalitu přepisu a nízkých nároků na odbornost přispěvatelů a vyhodnocení funkčnosti po devíti letech provozu.

Klíčová slova: přepis zvukových nahrávek, uživatelská interakce, komunitní spolupráce

Title: Iterative Improving of Transcribed Speech Recordings Exploiting Listeners' Feedback

Author: Jan Oldřich Krůza

Institute: Institute of Formal and Applied Linguistics

Supervisor: Doc. RNDr. Vladislav Kuboň, Ph.D., Institute of Formal and Applied Linguistics

Consultant: Mgr. Nino Peterek, Ph.D., Institute of Formal and Applied Linguistics

Abstract: This Ph.D. thesis deals with making a corpus of audio recordings of a single speaker accessible to wide public and interested community.

The work has been motivated by the existence of a set of perishing recordings of the Czech philosopher Karel Makoň on magnetophone tapes. The aim is to conserve the material for future generations and to make it accessible using digital technologies, in particular publishing the recordings online and enabling the users to search through them.

The thesis introduces the creation of a system for transcribing a large set of speech recordings employing a lay community. The solution designed is based on obtaining a baseline low-quality transcription by means of automated speech recognition and developing an application that allows for collecting corrections of the automatic transcription in a fashion that makes it usable as training data for further improvement of said transcription.

The spoken corpus itself is described. The author and his works, topics covered in the talks, the process of recording and digitization as well as the gained transcription are introduced. Next, the development of a system for automated transcription of the corpus, from collecting data, to acoustic and language modeling, various experiments undertaken and evaluation are presented. Then, the web application for gathering manual transcript corrections is described. Differences to other settings, design and implementation details, a way to compensate high demand for transcription quality and low demand for worker expertise, as well as an evaluation of the system's performance after nine years of operation are covered.

Keywords: speech transcription, user inetraction, community cooperation

# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
1.1	Motivace k disertaci . . . . .	4
1.2	Obsah . . . . .	5
<b>2</b>	<b>Data</b>	<b>7</b>
2.1	Karel Makoň . . . . .	7
2.1.1	Život Karla Makoňe . . . . .	7
2.1.2	Spisovatelská a přednášková činnost . . . . .	8
2.1.3	Rysy Makoňovy nauky . . . . .	9
2.2	Témata v mluveném korpusu . . . . .	11
2.3	Nahrávání . . . . .	16
2.4	Digitalizace . . . . .	16
2.4.1	Volba identifikátorů . . . . .	17
2.4.2	Přepisy . . . . .	17
<b>3</b>	<b>Akustické vlastnosti Makoňova korpusu</b>	<b>20</b>
3.1	Výchozí akustická kvalita . . . . .	20
3.2	Metrika . . . . .	20
3.3	Shlukování . . . . .	30
3.4	Kompenzace . . . . .	31
3.4.1	Spektrální odečet šumu . . . . .	32
3.4.2	Neurální doménový transfer . . . . .	33
3.4.3	Vyhodnocení . . . . .	33
<b>4</b>	<b>Jednání parlamentu jako trénovací data</b>	<b>35</b>
4.1	Příprava dat . . . . .	35
4.1.1	Zarovnávání . . . . .	36
4.1.2	Tvorba potenciálních trénovacích vzorků . . . . .	37
4.1.3	Výběr trénovacích vzorků . . . . .	37
4.1.4	Shrnutí extrakce trénovacích dat . . . . .	38
4.2	Číslovky a zkratky . . . . .	39
<b>5</b>	<b>Automatický přepis</b>	<b>40</b>
5.1	Vybrané milníky v rozpoznávání řeči . . . . .	41
5.2	Kódování signálu . . . . .	41
5.3	HMM . . . . .	43
5.4	Předchozí práce v rozpoznávání řeči . . . . .	46
5.4.1	Ircing et al. 2001 . . . . .	46
5.4.2	Psutka et al. 2002 - 2005 . . . . .	47
5.4.3	Renals et al. 1994 . . . . .	47
5.4.4	Graves & Jaitly 2014 . . . . .	48
5.4.5	Deep Speech . . . . .	51
5.4.6	Shrnutí . . . . .	52
5.5	Přepis Makoňova korpusu pomocí GMM-HMM . . . . .	52
5.5.1	Modelované hlásky . . . . .	52

5.5.2	Tvorba akustického modelu . . . . .	54
5.5.3	Dekódování . . . . .	56
5.6	Jazykový model . . . . .	56
5.7	Rozdělení dat . . . . .	59
5.8	Experiment s keprální normalizací . . . . .	60
5.9	Aktivní učení . . . . .	60
5.10	Rozšíření trénovací množiny . . . . .	61
5.11	ASR na parlamentním korpusu . . . . .	61
5.12	Přepis Makoňova korpusu pomocí neuronových sítí . . . . .	62
5.13	OOV . . . . .	63
5.14	Úspěšnost . . . . .	63
<b>6</b>	<b>Webové rozhraní</b>	<b>66</b>
6.1	Porovnání s jinými scénáři . . . . .	66
6.1.1	Programy pro přepis . . . . .	66
6.1.2	Wiki . . . . .	67
6.1.3	Korpusy . . . . .	68
6.2	Popis webové aplikace . . . . .	68
6.2.1	Prototyp . . . . .	68
6.2.2	Základní rysy druhé verze . . . . .	69
6.2.3	Zobrazení přepisu . . . . .	73
6.2.4	Problém s rychlostí . . . . .	74
6.2.5	Řešení . . . . .	74
6.2.6	Vizuální odlišení manuálního a automatického přepisu . . . . .	75
6.2.7	Web Audio API . . . . .	75
6.3	Nucené zarovnání . . . . .	76
6.4	Rozdělení nahrávek na úseky . . . . .	77
6.4.1	Délka segmentů . . . . .	78
6.4.2	Metody hledání bodů předělu . . . . .	78
6.4.3	Výběr bodů předělu . . . . .	79
6.4.4	Pojmenování souborů . . . . .	81
6.4.5	Překryv úseků . . . . .	81
6.5	Použití aplikace . . . . .	82
6.5.1	Expertíza uživatelů . . . . .	82
6.5.2	Pořízení fonetického přepisu . . . . .	83
6.5.3	Fonetický zápis . . . . .	84
6.5.4	Vyhodnocení kvality přepisů . . . . .	86
6.6	Backend . . . . .	87
6.6.1	API . . . . .	87
6.6.2	Ukládání dat . . . . .	90
6.7	Budoucí práce . . . . .	90
<b>7</b>	<b>Vyhledávání</b>	<b>91</b>
7.1	Kvantitativní vyhodnocení . . . . .	91
7.1.1	Identifikace témat . . . . .	92
7.1.2	Korelace témat mezi nahrávkami a knihami . . . . .	93
7.2	Případová studie . . . . .	94

<b>8 Závěr</b>	<b>95</b>
8.1 Výsledky disertační práce . . . . .	95
8.2 Budoucí práce . . . . .	96
8.3 Poděkování . . . . .	97
<b>Seznam použité literatury</b>	<b>98</b>
<b>Seznam obrázků</b>	<b>106</b>
<b>Seznam tabulek</b>	<b>108</b>
<b>Seznam publikací</b>	<b>109</b>

# 1. Úvod

Muselo to být někdy v roce 2010, kdy mi moje kamarádka Alenka do ruky podala cédéčko s popiskem „Karel Makoň“ se slovy, že si to mám poslechnout. Doma jsem disk vložil do mechaniky a nechal se unášet slovy, která se mi vryla do paměti:

*„I v tomto systému lásky, ať konkrétně vypadá jakkoliv, platí, že tam není vším to, co dělám, nýbrž důležitou složkou je také to, co nedělám. Víš, podobnost je v klepání. Klepání je podobno dělat-nedělat, systému dělat-nedělat. Systému samočinného počítače, jedna, nula, nula, jedna, jedna, nula. Já žiju ve dvojnosti. Všechno v ní pulzuje. A taky tento systém, který ze dvojnosti vede, to je pulzace na vyšší úrovni.“*

Obsah, jakož i podmanivý hlas mluvčího, mne natolik strhly, že jsem se začal pít po tom, zda existují od tohoto Karla Makoně další nahrávky. Dozvěděl jsem se, že ano. Alenka mě odkázala na pana doktora Elgra ze Zlína, který jich má prý mnoho.

Vydal jsem se tedy do Zlína za panem doktorem Elgrem. Usměvavý šedivý pán s vyzařováním šlechtice mě uvedl do svojí pracovny, jejíž celá stěna byla pokryta skříněmi zaplněnými kazetami a kotouči. Vyprávěl mi, jak po desítky let s „Karlíčkem“ jezdil a nahrával každé jeho slovo.

Nejdřív se mne zmocnil úžas, pak euforie a nakonec mne polil studený pot při pomyslení, že magnetický signál neúprosně slábne a že tento poklad je odsouzen k zániku... pokud jej někdo nezdigitalizuje.

Domluvil jsem se s dr. Elgrem, že si budu jeho sbírku po částech půjčovat a celou ji zdigitalizuji, aby se její obsah uchoval pro další generace. Následující dva roky jsem strávil přehazováním kazet v přehrávači a pravidelnými návštěvami Zlína s krosnou.

Výsledkem byla sbírka asi tisíce zvukových souborů, které jsem několika přátelům, kteří o Makoňově díle věděli, rozdál na pevných discích a vystavil ke stažení na internetu.

Hlavní a urgentní cíl, aby byly nahrávky zachráněny před degradací, se tím splnil. To byl však začátek věci.

## 1.1 Motivace k disertaci

Hlavním a společným bodem negativní zpětné vazby ke zdigitalizovanému materiálu bylo, že je v podstatě nemožné se v něm vyznat. Ani lidé, kteří byli nahrávání osobně přítomni, nedokázali najít pasáž, kterou by si rádi poslechli. Je sice pravda, že neméně nepřehledná byla sbírka před digitalizací, ale tím, že byla najednou celá k dispozici, tento problém vyvstal a nabral na aktuálnosti.

Pojal jsem tedy záměr archiv nejen zachránit a zpřístupnit, ale umožnit jeho maximální užitek co nejširšímu okruhu zájemců, ať už v přítomnosti nebo v budoucnosti. Za nejpřínosnější počín se mi jevílo pořízení kompletního přepisu. Tím by se umožnilo jak vyhledávání, tak jakékoliv další zpracování a prozkoumávání materiálu.

Přepsat ručně tisíc hodin bylo zcela mimo moje možnosti, byť bych se pokusil financovat placené přepisovatele nebo přepisovat svépomocí. Vědomí, že se jedná



o mluvené slovo jednoho mluvčího v jedné tematické doméně mi skýtala naději na přepis automatický. Navíc vědomí o existenci lidí, kteří se o nahrávky zajímají jako já, mne přivedlo k nápadu skloubit ruční přepis s automatickým. Stál jsem před úkolem designu mluveného korpusu, což přede mnou dělali mnozí jiní, např. Crowdy[1].

Konzultace s dr. Ninem Peterkem mi pomohla vytýčit cestu:

1. Ručně pořídím přepis několika minut Makoňových nahrávek.
2. Vytvořím systém rozpoznávání řeči.
3. Natrénuji ho na pořízeném přepisu.
4. Pořídím automatický přepis celého korpusu.
5. Naprogramuji webovou aplikaci, která umožní uživateli nahrávky přehrávat, synchronně u toho zobrazovat přepis a opravovat v něm chyby.
6. Opravy od uživatelů budu hromadit a používat jako další trénovací data pro rozpoznávač.
7. Tak se iterativně dopracuji ke kvalitnímu kompletnímu přepisu za pomoci vlastního úsilí a přispění komunity.

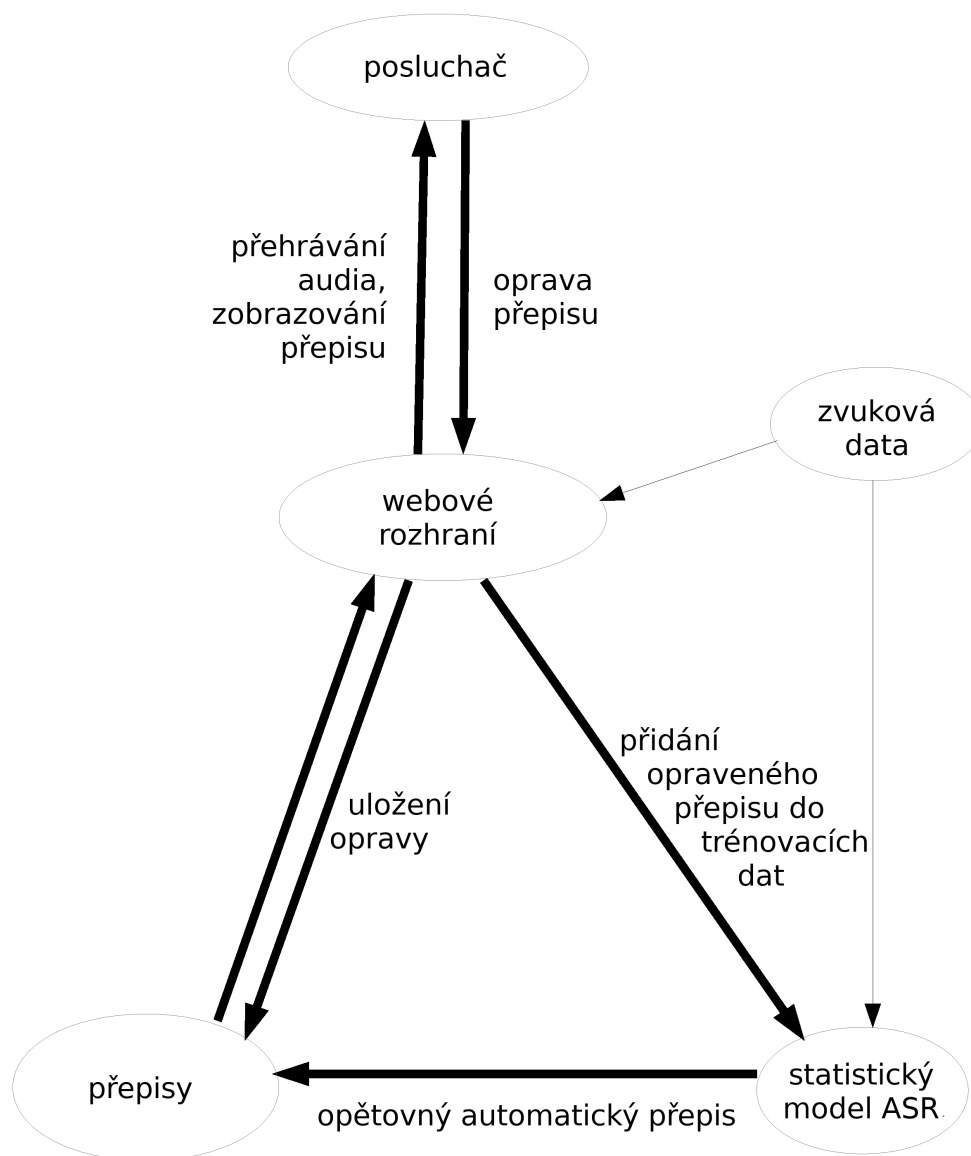
Architekturu ilustruje obrázek 1.1.

## 1.2 Obsah

Tématem této disertační práce je iterativní zdokonalování přepisu zvukových nahrávek s využitím zpětné vazby posluchačů, případně jejich další zpracování. S ohledem na dosažení tohoto cíle u výchozího korpusu zahrnuje tato práce následující body:

- rozbor samotných dat v kapitole 2 a 3,
- tvorba systému pro automatický přepis v kapitolách 4 a 5,
- webová aplikace pro sběr oprav přepisu v kapitole 6,
- a vyhledávání v korpusu v kapitole 7.

Práci uzavírá kapitola 8.



Obrázek 1.1: Schéma architektury systému.

## 2. Data

Zvukový odkaz Karla Makoně je východiskem pro tuto práci. O jeho díle neexistuje téměř žádná sekundární literatura, snad kromě článku v religionistickém časopise Dingir, viz Hájek (2007)[2]. V roce 2019 též byla obhájena bakalářská práce věnující se tématu utrpení u Karla Makoně jako mystika[3]. Osobně považuji Makoňovo dílo za jedno z nejzásadnějších vůbec v oblasti duchovního průkopnictví, a to jeho systematičností, obsáhlostí, návodností, novátorstvím a především hloubkou. Jeho nauka se od moderních duchovních směrů odlišuje kladným postojem k civilizačním trendům, nikoliv jejich zavrhováním, dále konzistentním souladem s rozumovým poznáním a pevnými základy v náboženských tradicích. Od vědeckého bádání se odlišuje zejména tím, že rozum a hmotu považuje za odrazové můstky k hlubšímu poznání, nikoliv za vrchol a jedinou platformu lidského poznání. Trvá však na tom, že duchovní zákonitosti jsou stejně tak pevně dané, univerzální a ověřitelné (ovšem pouze osobní, subjektivní zkušeností), jako zákony přírodní, popsané vědecky. Do třetice od klasické křesťanské literatury se liší obzvláště tím, že Ježíšovu nauku považuje za návod prvotřídní kvality pro vědomý vstup do věčného života zde na zemi a v těle, nikoliv po smrti. Věčným životem se míní stav, kdy člověk je vědomě věčnou bytostí nezávislou na pomíjejícím těle. Tvrdě kritizuje překonaný a naivní výklad, podle něž se ctnostným životem dá dojít po smrti do nebe, dále dojít spásy pouhou proklamací o víře v Krista a dodržováním příkázání a náboženských obřadů.

Odkrývá smysl života a návod na jeho uskutečnění, který nestojí na slepé víře ani na vlastní omezené lidské invenci.

### 2.1 Karel Makoň

#### 2.1.1 Život Karla Makoně

Ing. Karel Makoň se narodil 12. prosince 1912. Ve věku dvou let ho postihl zánět levého ramene. Lékaři doporučovali amputaci ruky, k čemuž jeho matka nedala souhlas a na vlastní zodpovědnost nechala dítě operovat. Vzhledem k tomu, že ještě nebyly objeveny krevní skupiny, nebyla možná transfúze a proto musela být operace prováděna opakovaně, aby dítě nevykrvácelo. Tehdejší anestetika nebylo možné mladému organismu podávat tak často, proto byly operace prováděny při vědomí. Malý Karel Makoň se, zažívaje nesnesitelnou bolest, naučil v raném věku opouštět při vědomí svoje tělo. Tato opakovaná zkušenost měla u něho následek, že po určitou dobu nepoznával svoji matku, zato začal spontánně rozpoznávat správné od nesprávného a důsledně činit, co poznával jako správné.

Období „činění správného“, kdy si kupříkladu zapověděl kouření, alkohol i veškerý pohlavní život, vyvrcholilo v Makoňových sedmnácti letech, kdy narazil na myšlenku, že „tento život je mostem do věčnosti“. Tím započalo období extází a vědomí, že je nesmrtelnou bytostí a smyslem jeho života je spojení s Bohem. Svojí matkou a prarodiči byl sice veden ke tradiční katolické víře, ale nikdy na ni nepřistoupil, protože „v nebi, kde by se jen díval na Boží tvář by byla strašná nuda“. Nikdy tedy nevěřil, v sedmnácti letech *poznal*.

V tomto období se ustavičně modlil za to, aby dokázal Boha více milovat. Tato

modlitba trvala devět let a jejím vyvrcholením byla deportace do koncentračního tábora v Sachsenhausen v roce 1939, coby českého vysokoškolského studenta.

V koncentračním táboře byl Makoň sužován více, než ostatní: měl jakýsi obzvláštní talent chytat rány a kopance. Prožíval nesmírné zmatení a frustraci nad tím, že tak dlouho tak věrně sloužil Bohu, a teď se s ním jedná jako s kusem hadru. Po čtyřech dnech utrpení nastal zlomový okamžik. Tamějším vězňům bylo zakázáno pod trestem smrti přihlížet zabití spoluvězňů příslušníky SS. Makoň si však nedal pozor a hleděl právě na takovou scénu. Vykonávající Němec si toho povšiml a vyzval Karla Makoně, ať zůstane stát na místě, že hned, jak dobjíe svoji momentální oběť, přijde zabít i jeho. Karel Makoň v tu chvíli raději odevzdal svůj život Bohu, a to bez přemýšlení a bezpodmínečně, se silou nabytou onou devítiletou modlitbou. Překvapivým výsledkem toho bylo, že SS-Mann tváří v tvář Makoňovi zbledl a v hrůze se obrátil na útěk.

Makoň tehdy obdržel všeobjímající poznání smyslu života a absolutní svobodu. Pohyboval se volně od baráku k baráku, nezažíval hlad ani jiný nedostatek, esesáci jako by ho neviděli. Trávil svoje dny v koncentračním táboře burcováním ostatních k probuzení k pravdě, kterou sám zažíval.

Zanedlouho byl z koncentračního tábora propuštěn. Celý zbytek svého života věnoval předávání svojí zkušenosti a hlavně návodu, jak k takové zkušenosti přijít bez nutnosti zažívat extáze i dramatické krize, neboť obojí považuje za nepřikladné.

Zemřel v roce 1993.

## 2.1.2 Spisovatelská a přednášková činnost

Již roku 1936 napsal Karel Makoň dopis *Utrpení a láska*, který poslal na podporu trpícímu příteli. Roku 1939 přeložil spis *Bhakti jóga*[4] od významného indického filozofa Svámího Vivékánandy. Jeho pozdější práce už rozvíjejí jeho vlastní dosažené poznání.

Do roku 1992, kdy v činnosti ustal, napsal a přeložil dílo o celkovém rozsahu 3 613 211 slov nebo též 25 069 991 znaků, čili necelých 14 tisíc normostran. Nejrozsáhlejším jeho dílem je trilogie *Mystika*, která sestává z dílů

1. *Západní starověká tradice* (1948),
2. *Srovnání jógy s křesťanskou mystikou* (1986–1989) obsahující překlad děl *Syntéza jógy*[5] od indického filozofa Šrí Aurobinda Ghóše a *Hrad nitra*[6] od středověké křesťanské mystičky sv. Terezie z Avily,
3. *Výklad evangelia Sv. Jana* (1950–1953).

Dalšími rozsáhlými knihami jsou *Cesta vědomí* (1973–1974), dále výklad církevního roku a církví doporučených biblických čtení *Postila* (1965), *Sladké jho* (1977–1980) obsahující překlad značné části díla *Précis de Théologie Ascétique et Mystique*[7] od francouzského teologa Adolphe-Alfreda Tanquerey, *Umění následovat Krista* (1971), autobiografické *Umění žít* (1969) a *Základní kurs nadživotnosti pro ty, kteří si myslí, že nevěří a základní kurs náboženství pro ty, kteří si myslí, že věří, neboť jsou si všichni rovni, pokud umírají, aniž by se během života znovu narodili* (1967–1968).

Nikoliv rozsahem, ale významem jsou hodny zmínky spisy *Obět mše svaté* (1951), kde je účast na katolické liturgii podána jako návod pro spojení s věčností, *Pohádka na dobrou noc* (1980), kde je odhalen duchovní smysl pohádky o Honzovi a *Blahoslavenství* (1973).

Makoňovo dílo se šířilo převážně samizdatem, a to i po Sametové revoluci. Psané dílo bylo několikrát kompletně přepsáno na psacích strojích a po příchodu osobních počítačů ještě jednou do digitální formy. Všechny knihy a spisy jsou volně k dispozici na stránkách [makon.cz](http://makon.cz). Jen hrstka knih byla vydána, a sice

1. Umění následovat Krista (1992)[8],
2. Pohádky nejen pro děti (1992 pod názvem *Odkrytá moudrost starých pravd*) [9],
3. Utrpení a láska (1995)[10],
4. Mystická koncentrace (1995 pod názvem *Mystická koncentrace a příprava k ní*)[11],
5. Otázky a odpovědi I - IV (1999 pod názvem *Světlo na cestu*)[12],
6. Blahoslavenství (2000)[13]<sup>1</sup>,
7. Úlohy (2002 pod názvem *Duchovní úlohy*)[14]<sup>1</sup>,
8. Základní kurs nadživotnosti (2005)[15].

Většina děl se snaží podat více či méně ucelený návod pro vědomý vstup do věčnosti, vždy z jiného východiska nebo pro jiný typ čtenáře. Například *Umění žít* je věnováno Makoňově nejmladší dceři a je z velké části vlastním životopisem. *Základní kurs nadživotnosti, pro ty, kteří si myslí, že nevěří, a základní kurs náboženství pro ty, kteří si myslí, že věří*, je kniha, která má společný úvod a závěr, ale hlavní stať je rozdělena na dvě oddělené části, jednu pro lidi nevěřící v Boha, kde se autor opírá o experimenty s tělem a o kritický přístup, zatímco v druhé části důkladně rozebírá smysl Otčenáše a radí, jak tuto modlitbu praktikovat pro její spojovací účel. *Umění následovat Krista* se zase soustředí na systematizaci cesty v rozdělení na očištnou, osvěcovací a spojovací část, jak to vykládá křesťanská mystická tradice.

### 2.1.3 Rysy Makoňovy nauky

Pokusím se krátce představit charakteristiky Makoňovy nauky, jak je hodnotím podle svojí osobní zkušenosti a svého názoru. Serióznější porovnání by bylo námětem na jinou disertaci na jiné fakultě.

Karel Makoň je moderním učitelem duchovní moudrosti. Jako mnozí vychází z křesťanství. Považuje bibli za vrcholný zdroj moudrosti a Ježíšův život a výroky za nejdokonalejší návod k duchovní realizaci, jaký je nám momentálně k dispozici. Zdůrazňuje, že je vždy potřeba se řídit Ježíšovým příkladem jako celkem, nikdy částí vytrženou z kontextu. Každý Ježíšův výrok a úkon má svůj protiklad. Jednou například Ježíš hlásá nenásilí a radí „nastavit druhou tvář“, ale podruhé bičem

---

<sup>1</sup> Duchovní úlohy a Blahoslavenství mají, zdá se, totožné ISBN.

vyhání kupce z chrámu. Jedině syntéza výroků a činů s jejich protiklady mohou podle Makoně poskytnout použitelný návod, kterým se dá v životě obecně řídit.

Striktně zavrhuje doslovný výklad tzv. nadpřirozených událostí. Například apokalypsa – konec světa a druhý příchod Ježíšův, je podle něho ryze individuální záležitostí, která se stane každému člověku v jiný okamžik podle jeho vývoje. Dokládá to Ježíšovým výrokem, že „nepomine toto pokolení, než se to všechno stane“ (Mk 13.30), dále srovnáním s fenoménem mystické smrti, známým od mnoha jednotlivců i různých tradic, a vlastní zkušeností. Stejně tak stvoření světa je podle něho popisem vývoje lidského jedince, obzvlášť jeho nitra. K tomu zdůrazňuje, že nejde o jednorázový čin, nýbrž soustavné tvoření, které neustále probíhá, a sedm dní stvoření je sedm kvalit, které jsou ve stvoření neustále přítomny.

Rozlišuje mezi Ježíšem, který symbolizuje naši věčnou podstatu, a Kristem, který symbolizuje spasitelský úkol Boží. Že tento není závislý na fyzické osobě Ježíše, dokládá jeho výrokem: „Dříve, než Abraham byl, já jsem.“ (J 8.58)

Karel Makoň má několik oblíbených pasáží z bible, ke kterým se často vrací. Asi nejvýznamnějšími z nich jsou podobenství o marnotratném synu a podobenství o hřivnách. V podobenství o marnotratném synu (Lk 15.11-32) popisuje symbol plně rozvinutého lidského života, kde promrhání znamená investici do pomíjejícího, a je nezbytnou podmínkou pro vzpomínku na otcův dům, tedy uvědomění si vlastní věčné podstaty, a sjednocení bytosti kolem touhy po návratu. Otcovo ocenění syna šatem, prstenem a zabitím telete ukazuje na fakt, že jde o vyšší a tedy žádoucí stav oproti „dobrému synovi“, který otcovo dědictví nepromarnil. Podobenství o hřivnách (Lk 19.11-27) předestírá jako návod pro životní situace obecně, kde se doporučuje spatřovat nejen ve statecích, ale i v situacích hřivny dané od Boha, se kterými nakládáme nikoliv pro sebe, ale pro něho. Zúčtování, kdy hospodář přichází, aby si vzal výtěžek z hřiven, máme vidět v situacích, kdy přicházíme o kontrolu nad výsledkem svého snažení či nad situací. Moment odevzdání hřiven hospodáři bez sebemenší pohnutky nechat si z výtěžku něco pro sebe je klíčovým a následné udělení měst namísto hřiven k hospodaření je univerzálním pravidlem rozmnožení darů a zodpovědnosti.

Křesťanské tradici vyčítá operování s nevyzpytatelnou Boží milostí. Bůh podle Karla Makoně není člověk a tím méně náladový člověk, aby se mu tu něco zlíbilo a ondy nezlíbilo nebo znelíbilo. I města za hřivny v podobenství obdrželi služebníci nikoliv na základě toho, jakou měl hospodář náladu, nýbrž podle míry svého hospodaření. Stejně tak je zákonité, kdy člověka potká mystická zkušenost a vůbec cokoliv, co tradice připisuje nevyzpytatelné Boží milosti. Podmínky pro tuto dispozici důkladně popisuje. Já jen shrnu, že stěžejním bodem je, jak to vyplývá z podobenství o hřivnách, vynaložení veškerých lidských sil (znásobení hřiven bez přítomnosti hospodáře) pro nadpozemský cíl (hospodaření pro hospodáře, ne pro sebe) a následné dokonalé odevzdání, když jsou lidské síly vyčerpány.

Dalším výrazným rysem Makoňovy nauky je, že vše podřizuje dosažení království Božího, čímž se odlišuje od mnoha moderních duchovních učitelů, kteří mnohdy vycházejí z lidských potřeb šťastného života, vztahů, hojnosti a podobně. Jednak ho to připodobňuje ke klasickým katolickým autorům a jednak (podle mne právě proto) jeho nauka zaujme jen nepatrný zlomek lidí, kteří mají zájem o duchovno. Světské lidské problémy nebagatelizuje, doporučuje naopak univerzální metodu pro jejich řešení, například v díle *Zlatý klíč*, ale pro člověka, který nehledá království Boží především, je tato metoda v podstatě nepřístupná.

Na rozdíl od většiny moderních učitelů moudrosti má Karel Makoň velmi pozitivní postoj ke všem civilizačním změnám, včetně technizace, rozmachu všudypřítomného vlivu systému, daním atp. Považuje je za příležitost nežít pro sebe, nýbrž pro společnost, a tím trénovat život pro věčnost. Naopak vůči sexualitě, sám užívá termín „pohlavní život“, se staví mnohem zdrženlivěji, než většina mně známých moderních autorů. Vidí v sexualitě především vybíjení boží síly za účelem vstupu do zvířecího ráje a doporučuje aspoň část života prožít bezpohlavně.

## 2.2 Témata v mluveném korpusu

Celý mluvený korpus Karla Makoně, jakož i jeho celé psané dílo, má jednotné téma, jež by se dalo shrnout jako návod k vědomému spojení s věčností. V průběhu času i podle toho, komu byla konkrétní promluva určena, se však mění i témata jemnějšího rozlišení.

Můžeme nalézt témata opakující se napříč celým korpusem. Namátkou mohou zmínit:

1. soupeření Eliáše s Bálovými kněžími,
2. stvoření jako probíhající proces,
3. Job,
4. prvních 30 let Ježíšova života,
5. křest v Jordánu,
6. zázrak na svatbě v Káně galilejské,
7. podobenství o marnotratném synu,
8. podobenství o hřivnách,
9. symbolika apoštolů coby lidských schopností,
10. Lazar,
11. úkol Jidášův,
12. ukřižování,
13. obrácení Šavla ve svatého Pavla,
14. manželé, kteří padli mrtvi, ve Skutcích,
15. Svatá Terezie z Avily,
16. Otec Pio,

17. Svatý František z Assisi,
18. Svátá Terezie z Lisieux,
19. Svatý Augustin,
  
20. Lao C',
21. Milarepa,
22. Siddhárta Gautama Buddha,
23. Rámakrišna,
24. Karel Weinfurter,
  
25. operace v dětství,
26. extatické stavy,
27. konání správného,
28. devět let modlitby a skrytá sebeláska,
29. koncentrační tábor,
30. vlité poznání,
  
31. celobytné sjednocení,
32. symbolika matematických vzorců,
33. posmrtný život,
34. hadí síla,
35. Satan,
36. stylizace života,
37. zákonitost Boží milosti,
38. sat, čit, ánanda,
39. indická tradice,
40. mithraismus.

Jmenovaná témata můžeme rozdělit do následujících kategorií

- starozákonní postavy a události,
- život Ježíšův,



- ostatní postavy a události Nového zákona,
- křesťanští světci,
- ostatní významné osobnosti,
- události z Makoňova vlastního života
- prvky na cestě k Bohu obecně.

Systematická identifikace témat a anotace korpusu vzhledem k nim je předmětem budoucí práce. Inspirací pro tematickou anotaci může být např. Skorkovská (2011)[16].

O zmapování témat a jejich pokrytí v korpusu proběhlo a probíhá několik pokusů. Prvním z nich jsou strojově psané indexy k magnetofonovým páskám. Ty jsem nafotil do 258 fotografií, pro ilustraci viz obrázek 2.1. Jejich obsahem je posloupnost záznamů, z nichž každý je uvozen pozicí počítadla na magnetofonu, za čímž následuje shrnutí tématu přibližně do 50 znaků. Některé záznamy jsou zvýrazněny podtržením či kapitálkami. Typická délka jednoho takto označeného úseku je 1–10 minut. Tyto indexy jsou přiložené k nahrávkám. Problémem je, že u kotoučů není patrné, který digitalizovaný soubor odpovídá které stopě označené v indexu jako *a*, *b*, *c* nebo *d*.

Krom toho existují podrobnější indexy ve formě listů formátu A4, psané z menší části na stroji, z větší části psacím písmem, kde interval mezi jednotlivými úseky je často v řádu desítek sekund. Těch je nafocených 350 stran. Viz příklad na obrázku 2.2.

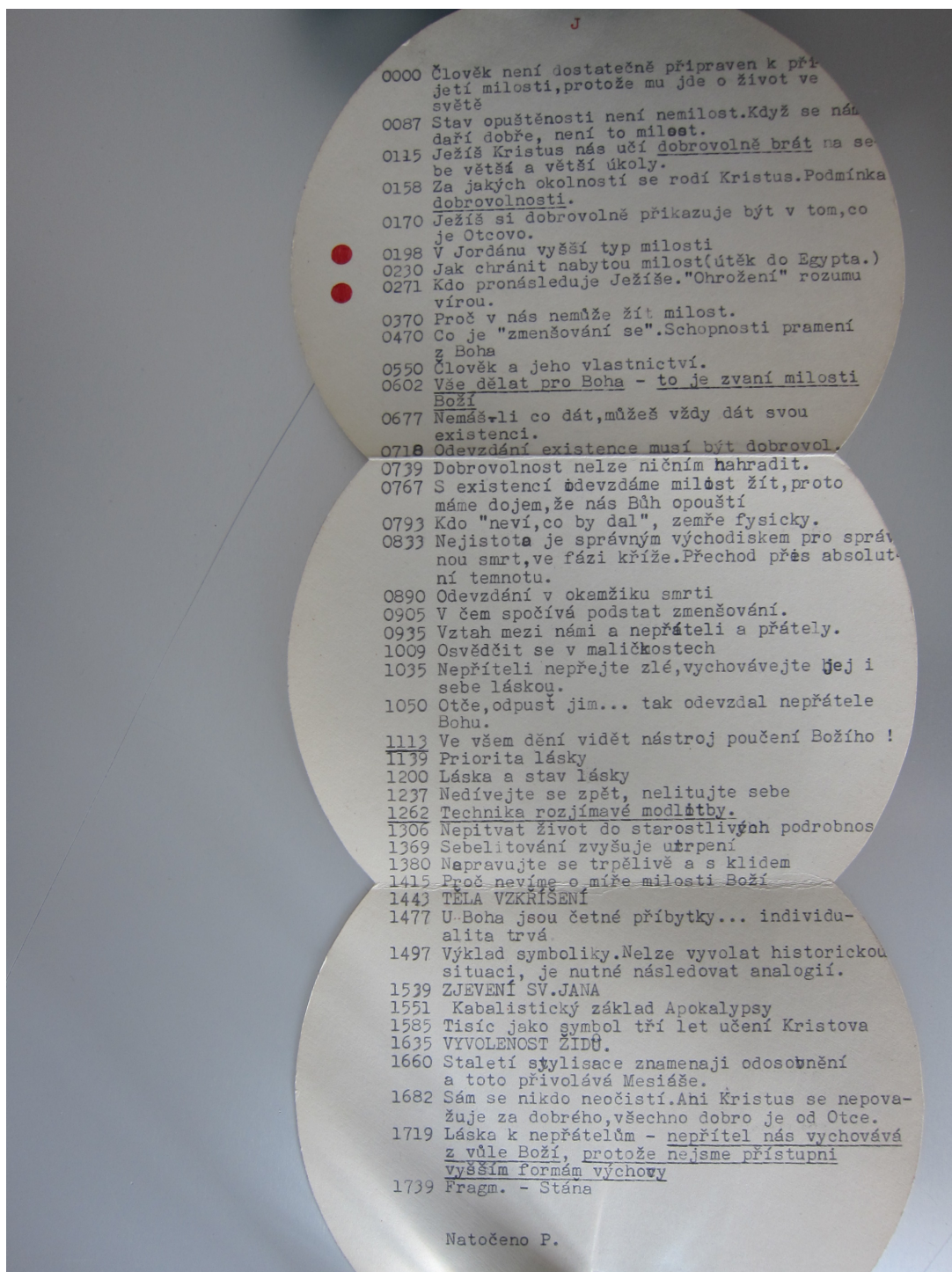
V rámci tohoto projektu učinil Ing. Milan Tulach vždy po přepisu přednášky výběr pasáží, které považoval za stěžejní, a opatřil je časovými značkami. Pokryl tak toho času 44 nahrávky.

K poslouchaným nahrávkám si dělám poznámky stejným způsobem jako jsou k ručně psaným indexům, ale digitálně a pozice označuji identifikátorem nahrávky a časovou pozicí. Zatím jsem takto pořídil 3361 záznam z 211 nahrávek.

Pozice počítadla magnetofonu v indexech jsem se pokusil převést na časové údaje. Počítadlo se inkrementuje při každé otáčce cívky, na kterou se páska navíjí. Uplynulý čas je tedy kvadratická funkce pozice počítadla. Mělo by tedy stačit nalézt několik málo odpovídajících párů pozic počítadla a času a z toho interpolovat koeficienty v hledaném polynomu.

Tak jsem dospěl k výrazu  $t = 0,0008605c^2 + 1,674c + 3,998$ . Bohužel vyšlo najevo, že po hodině trvání nahrávky se kotouč otáčí tak pomalu, že pozice počítadla se změní až za několik desítek sekund. To způsobuje, že interpolační koeficienty jsou velmi nepřesné a v kombinaci s nepřesností při určování nové pozice to způsobuje, že začátek hledaného tématu se nachází i více než minutu před či za predikovanou pozicí.

Tento problém lze řešit buď přesnějším určením funkce času podle pozice počítadla, a to použitím hardwarového počítadla či přesnější interpolací pomocí většího množství bodů, anebo využitím predikce začátku tématu v několikaminutovém okolí pomocí počítačnělingvistických metod.



Obrázek 2.1: Index přiložený k jednomu z kotoučů.



## 2.3 Nahrávání

O průběhu nahrávání mám pouze kusé a anekdotické informace. Nejstarší datum, na které jsem u nahrávky narazil, je z roku 1970. Je možné, že některé nahrávky jsou staršího data, ale nic nenasvědčuje tomu, že by jich bylo mnoho a byly výrazně starší. Vzhledem k tomu, že Karel Makoň začal evangelizovat už v koncentračním táboře na konci roku 1939 a přestal až v roce 1992, je pravděpodobné, že záznam je k méně než polovině slov, která vyřkl.

Přednášky v úzkém kruhu přátel se konaly na různých místech Československa, později ČR. Existovala skupinka v Plzni, v Praze a v Gottwaldově, dnešním Zlíně. Některé z nahrávek jsou z několikadenních skupinových setkání v Kalech u Brna, na chatě Čeřínek v Českomoravské vrchovině nebo ve Žiaru na Slovensku. Ostatní pocházejí většinou z přátelských setkání u někoho v soukromí.

Naprostou většinu nahrávek, které mám k dispozici, pořídil dr. Elger. Nahrávalo se s tehdejší nejmodernější běžně dostupnou technikou a pásky byly pečlivě skladovány. Nahrávky z ostatních míst také existují, ale jsou spíše raritou a často jsou hůře zachované a ne tak systematicky označené.

Část nahrávek (30% celkové délky) je nahraných na kotouče, zbytek na kazety. Ve většině případů má každý kotouč a každá kazeta identifikátor. Některé kusy jsou bez identifikátoru, některé dvojice mají totožný identifikátor a různý obsah.

Většina nahrávek je z kazet s identifikátorem ve formátu YY-MN. Například 85-05 je pátá kazeta z roku 1985. Z takto označených kazet pochází 686 výsledných souborů z celkových 802, které mají původ v kazetách.

Celkem z 39 kotoučů, které jsem sám digitalizoval, jich 36 bylo nahraných rychlostí 9,53 centimetru za sekundu. Zbylé tři rychlostí 2,38 centimetru za sekundu. Na jeden průchod takového kotouče se vejde šest hodin záznamu, ale za cenu citelného snížení kvality, zvláště po dekadách skladování. 24 kotouče jsou označeny písmenem. Posloupnost je více méně abecední, ačkoliv tři různé kotouče sdílejí identifikátor s písmenem „I“ a nedostala se ke mně žádná páska s písmenem „G“; asi se ztratila. 85 kotoučů (včetně těch, které jsem nedigitalizoval já) má v identifikátoru ročník v rozmezí od 1973 do 1988. Vyskytuje se zde však mnoho duplicit, takže skutečný počet rozličných nahrávek je v této kategorii pravděpodobně mnohem menší. Deset kotoučů má číselný identifikátor, dvacet devět textový a dva byly vůbec bez identifikátoru.

Existují také dva videozáznamy. Jeden, tříhodinový, je ke zhlédnutí na YouTube: <https://www.youtube.com/watch?v=UaNm9jnnJiA>

## 2.4 Digitalizace

U většiny kazet byla digitalizace prováděna stylem jedna strana do jednoho souboru. U kotoučů to byl jeden kanál jednoho průchodu z kotouče na kotouč do jednoho souboru. Výjimku zde tvoří kazety digitalizované v módu auto-reverse, s čímž jsem v průběhu dvou let digitalizace experimentoval.

Následuje vyčerpávající výčet médií, které odpovídají jednotlivým digitalizovaným souborům:

- strany kazet: 615 souborů,
- celé kazety: 140 souborů,

- průchody z kotouče na kotouč: 112 souborů,
- převzaté, nejisté: 222 soubory,
- dvě po sobě jdoucí kazety: 1 soubor.

Převzaté soubory byly digitalizovány dříve, než jsem se k nim dostal. Formát pro digitalizaci, který jsem používal, byl nejdříve 44100Hz pro prvních 810 souborů, posléze 48kHz, 16 bitů v reálném čase. Výjimku z digitalizace v reálném čase tvoří kotouče nahrané rychlostí 2,38 cm/s. Ty byly digitalizovány standardní rychlostí 9,53 cm/s, načež jim byla nastavena čtvrtinová vzorkovací frekvence.

Pro digitalizaci jsem používal nejdříve zařízení *Ion Tape 2 PC*, které poskytuje USB rozhraní jako externí zvuková karta. Později jsem začal používat přehrávač *Denon DRW-585* a externí zvukovou kartu *Lexicon Alpha* coby převodník z analogového signálu do digitálního formátu. Kotouče jsem digitalizoval pomocí přehrávače *Tesla B-115* připojeného zezadku do *Ion Tape 2 PC*, později do *Lexicon Alpha*.

### 2.4.1 Volba identifikátorů

Každý takto zdigitalizovaný soubor jsem pojmenoval podle následujícího algoritmu: Pokud šlo o kazetu s identifikátorem, použil jsem tento identifikátor a k němu připojil písmeno A nebo B pro rozlišení strany. Pokud byl na obalu nějaký další popisek, připojil jsem ho za pomlčku k oběma stranám. Pokud byl popisek na kazetě, připojil jsem ho pouze k souboru odpovídající strany.

V případě kotoučů jsem identifikátor prefigoval řetězcem *kotouc-* a připojil rozlišovací dvoumístné číslo počínaje 01 a označení stopy *-a* až *-d*.

Písmeno *a* dostal vždy levý kanál prvního průchodu, *c* pravý kanál. Písmeno *b* dostal levý kanál zpátečního průchodu a *d* pravý kanál. Ukazuje se, že toto značení nebylo nejšťastněji zvolené, protože některé kotouče jsem dostal převrnuté na opačnou stranu, takže dochází k nepředvídatelné záměně mezi písmeny *a*, *c* versus *b*, *d*.

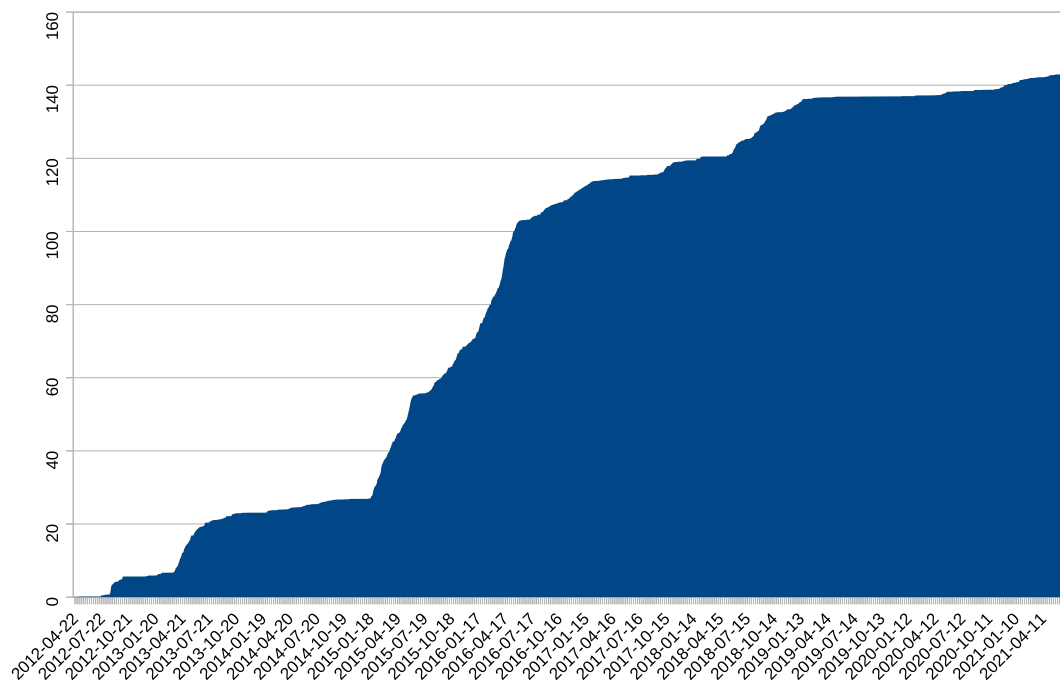
Pokud nahrávka identifikátor neměla, použil jsem řetězec *neident* a pořadové číslo.

### 2.4.2 Přepisy

K 16. květnu 2021 existuje ke korpusu kompletní přepis o 8 066 791 slovech, z nichž 728 286, tedy 9,03% je přepsáno ručně. Ruční přepisy pokrývají přesně 107 hodin 41 minutu a 49 sekund z celkových 1050 hodin 5 minut a 3 sekund nahrávek.

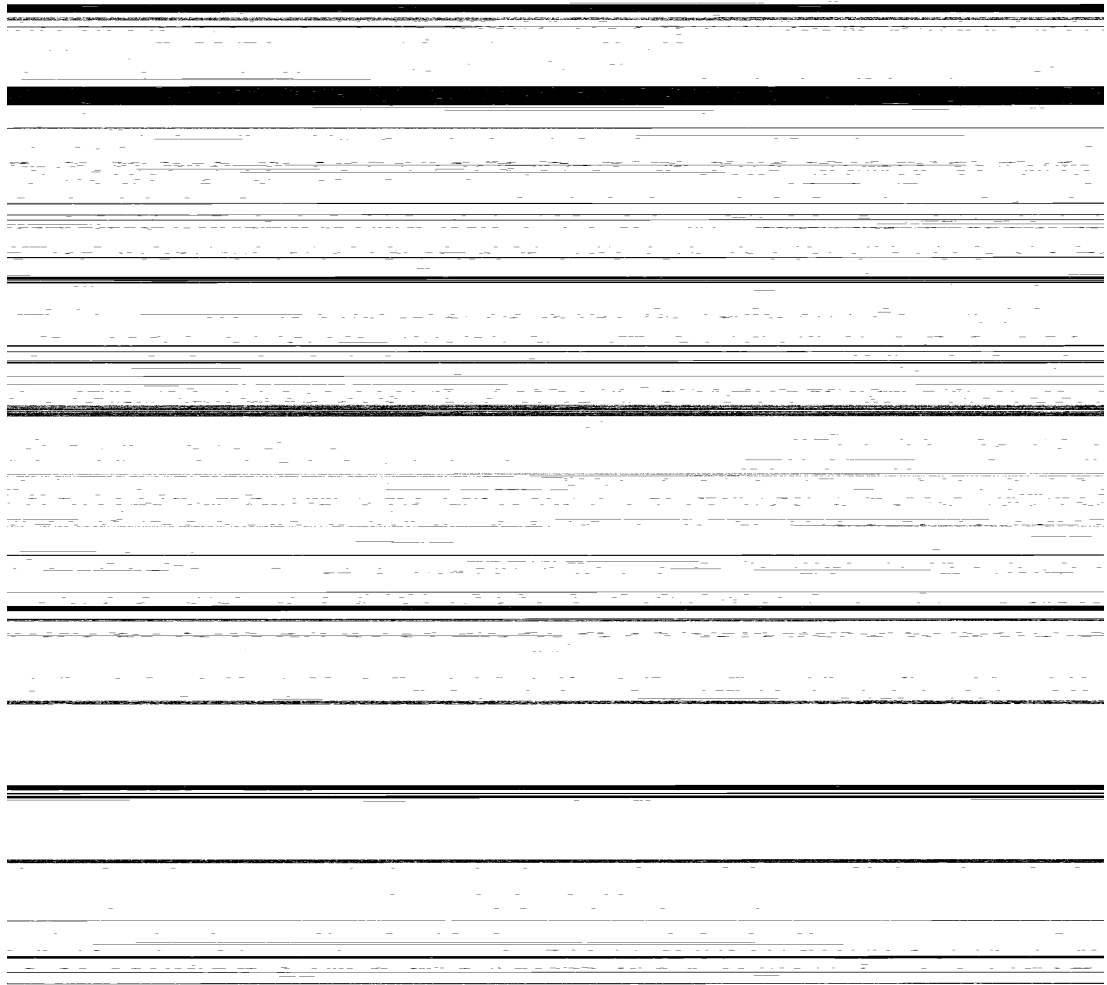
První manuálně přepsaný segment byl pořízen v dubnu 2012 a od té doby do dneška jich bylo posláno 131 483. Z toho 117 176, to jest 89%, pochází od čtyř přispěvatelů (mezi něž já nepatřím). Obrázek 2.3 ukazuje, jak přibývaly hodiny přepisů v čase. Počítá se zde celkový objem příspěvků, které se namnoze překrývají, takže skutečný čas přepsaného záznamu je „jen“ zmiňovaných 107 hodin.

Podíl manuálně a automaticky přepsaných pasáží ilustruje obrázek 2.4. Každý pixel reprezentuje jedno slovo v přepisu, přičemž černé pixely představují manuálně přepsaná slova.



Obrázek 2.3: Přibývání celkového času oprav v hodinách.

Mechanismus pořizování přepisů dopodrobna rozvádí kapitola 6.



Obrázek 2.4: Distribuce automaticky (bílá) a manuálně (černá) pořízených přepisů.

# 3. Akustické vlastnosti Makoňova korpusu

Mluvený korpus Karla Makoňe vyniká vzhledem ke své velikosti konzistencí téměř výlučně jediného mluvčího a velmi úzkou tematickou doménou. Jistou protiváhu této konsistentnosti představují jeho akustické vlastnosti.

## 3.1 Výchozí akustická kvalita

Akustická kvalita nahrávek je největší slabinou korpusu. Kvalita není konzistentně špatná, je velmi kolísavá. Na kvalitu záznamu má vliv jeho stáří, použité médium, rychlost záznamu, způsob skladování, použitý magnetofon, mikrofon, pozice mikrofonu, akustické vlastnosti prostředí jako ozvěna, hluk na pozadí, momentální dispozice mluvčího a také to, zda se jedná o původní nahrávku nebo její kopii<sup>1</sup>.

Obrázky 3.1 až 3.9 ukazují spektrogramy nahrávek různých kvalit. Vždy se jedná přibližně o třísekundový úsek a součástí popisku je odkaz pro přehrání odpovídajícího zvuku.

Systematicky můžeme újmu na kvalitě rozdělit takto:

1. aditivní šum, jevící se jako hučení až syčení, obzvláště patrný v tichých pasážích,
2. stacionární nebo téměř stacionární rušení, například monotónní pískání na několika frekvencích, které produkují nekvalitní obvody magnetofonu nebo mazací tón,
3. nestacionární rušení, např. řeč na pozadí, bouchnutí dveří apod.,
4. velká ozvěna místnosti nebo špatně ekvalizovaný mikrofon zesilující některé frekvence na úkor jiných,
5. nelineární zkreslení magnetofonu,
6. kolísání rychlosti.

K těmto újmám na kvalitě dochází při záznamu na magnetofonový pásek. Při přehrávání nastávají další.

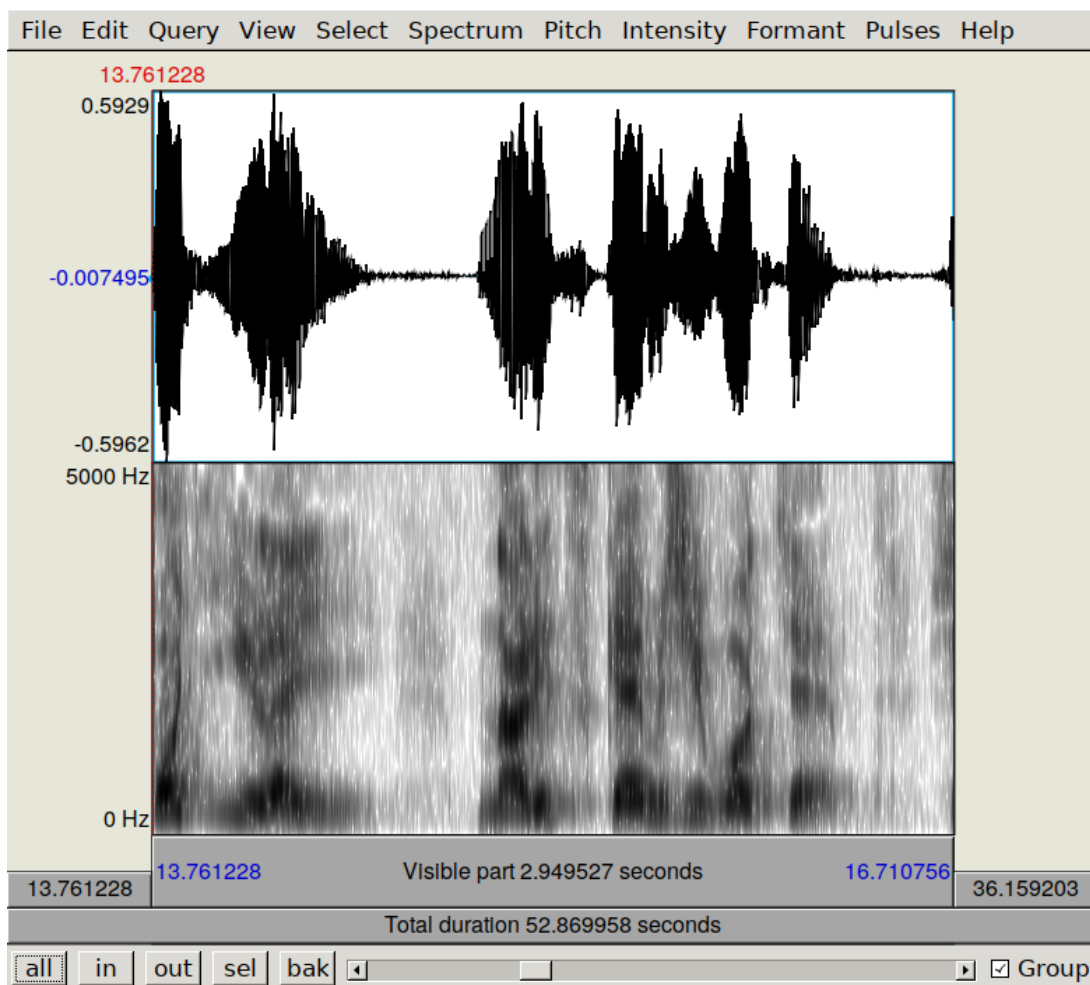
## 3.2 Metrika

Předpokladem pro práci s akustickými vlastnostmi dat je porovnávání na jejich základě. Je nutné mít metriku, která by odrážela akustickou podobnost jednotlivých souborů. K tomu využívám algoritmu, který navrhuje Mandel a Ellis[17] v implementaci programu Musly od Dominka Schnitzera[18]. Pro tuto metriku používám v textu pojem *akustická vzdálenost*.

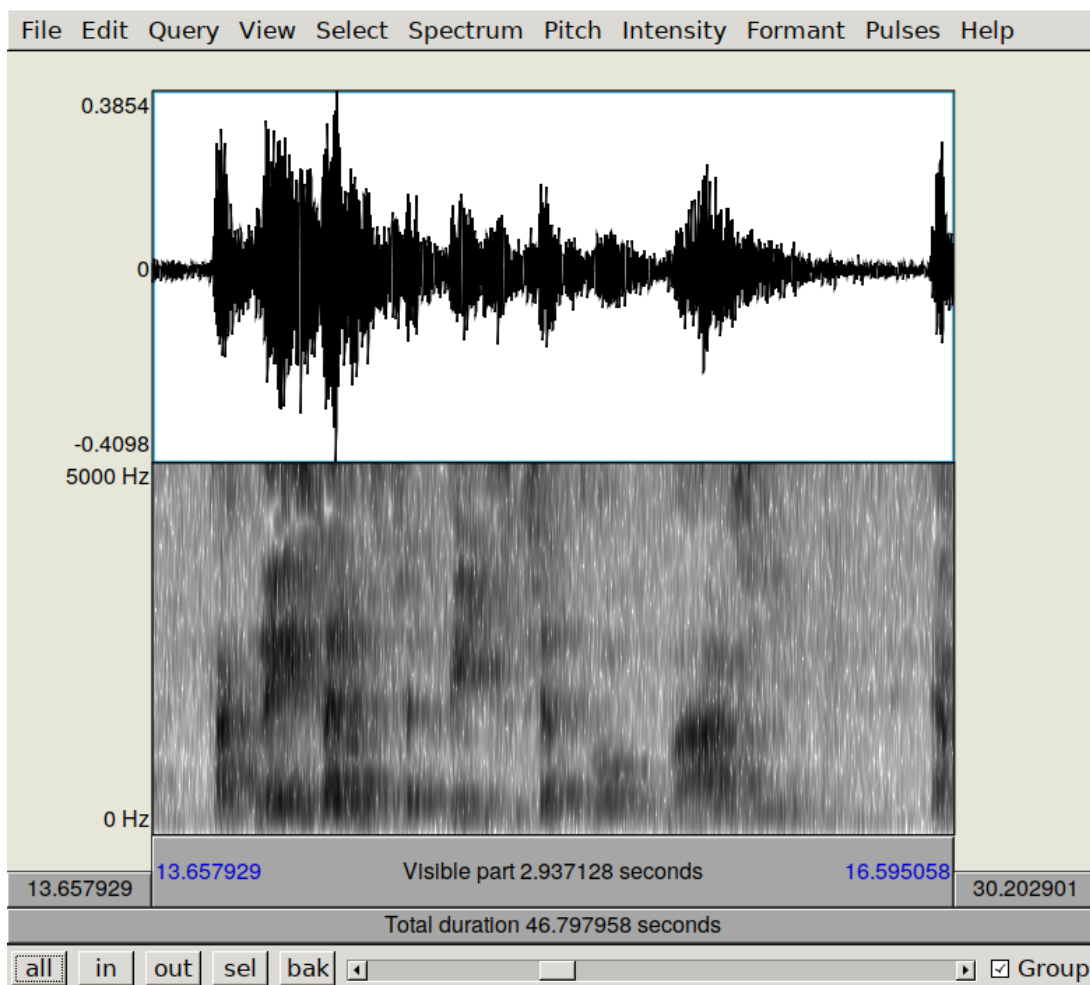
---

<sup>1</sup>Kopírování magnetofonových pásek je ztrátový proces.



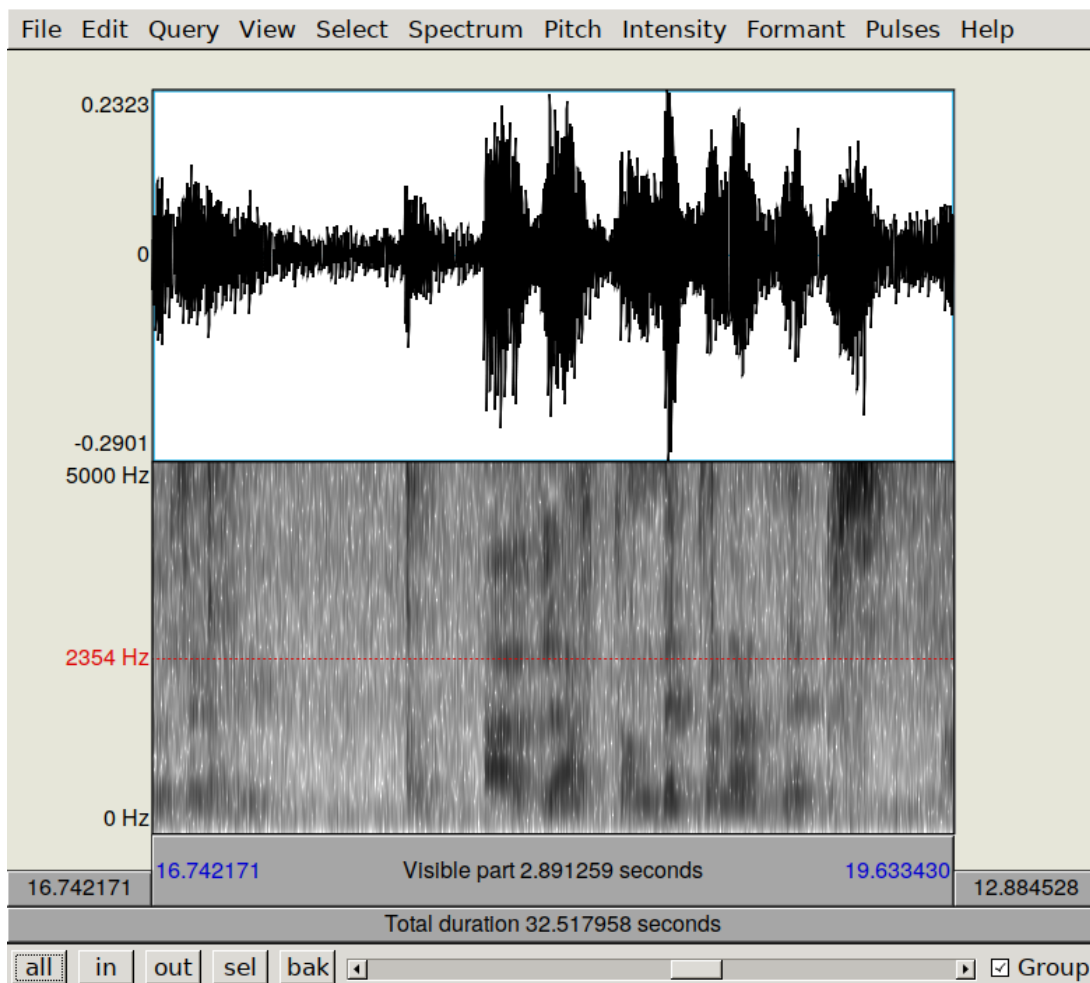


Obrázek 3.1: Kvalitní záznam bez zjevných defektů.  
<http://radio.makon.cz/zaznam/90-02A#ts=673.14>



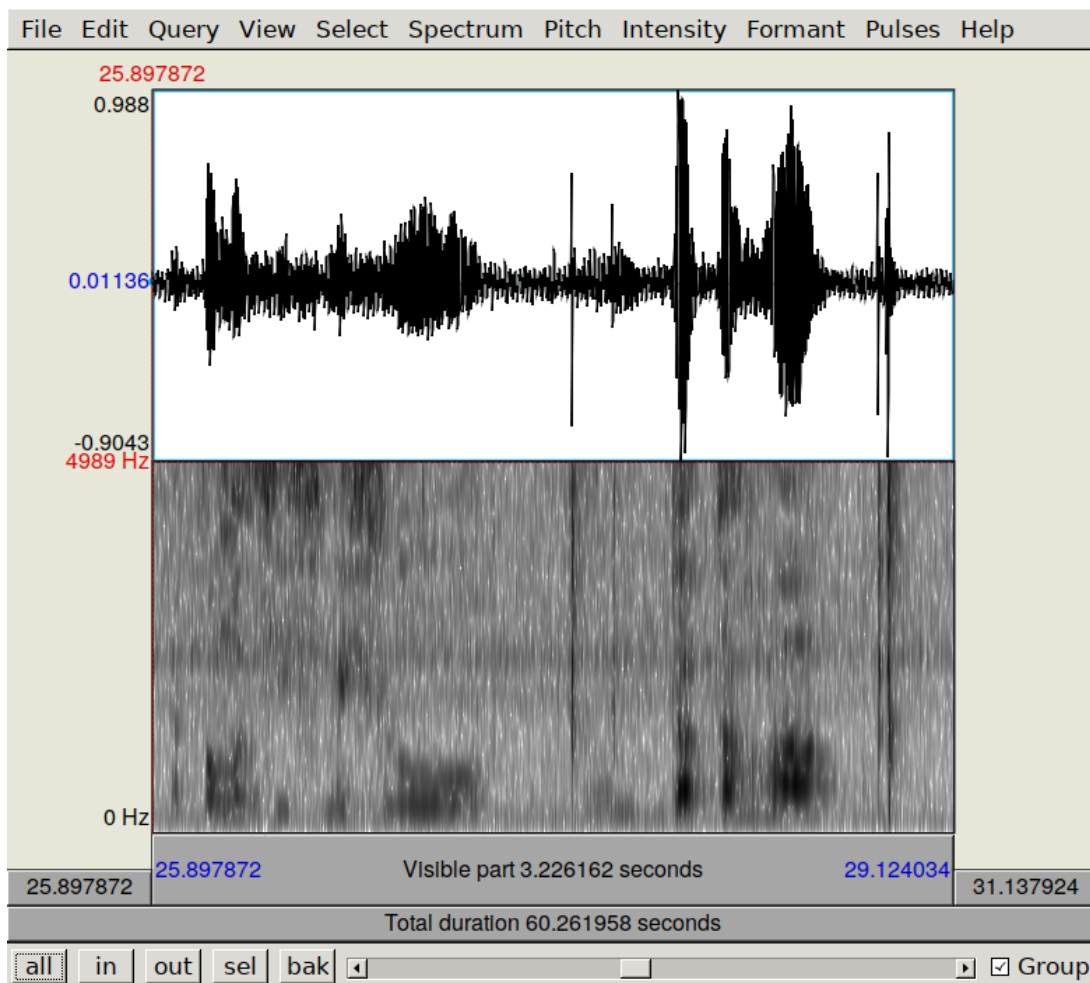
Obrázek 3.2: Výrazné echo.

<http://radio.makon.cz/zaznam/90-24A-24.4.90#ts=664.33>



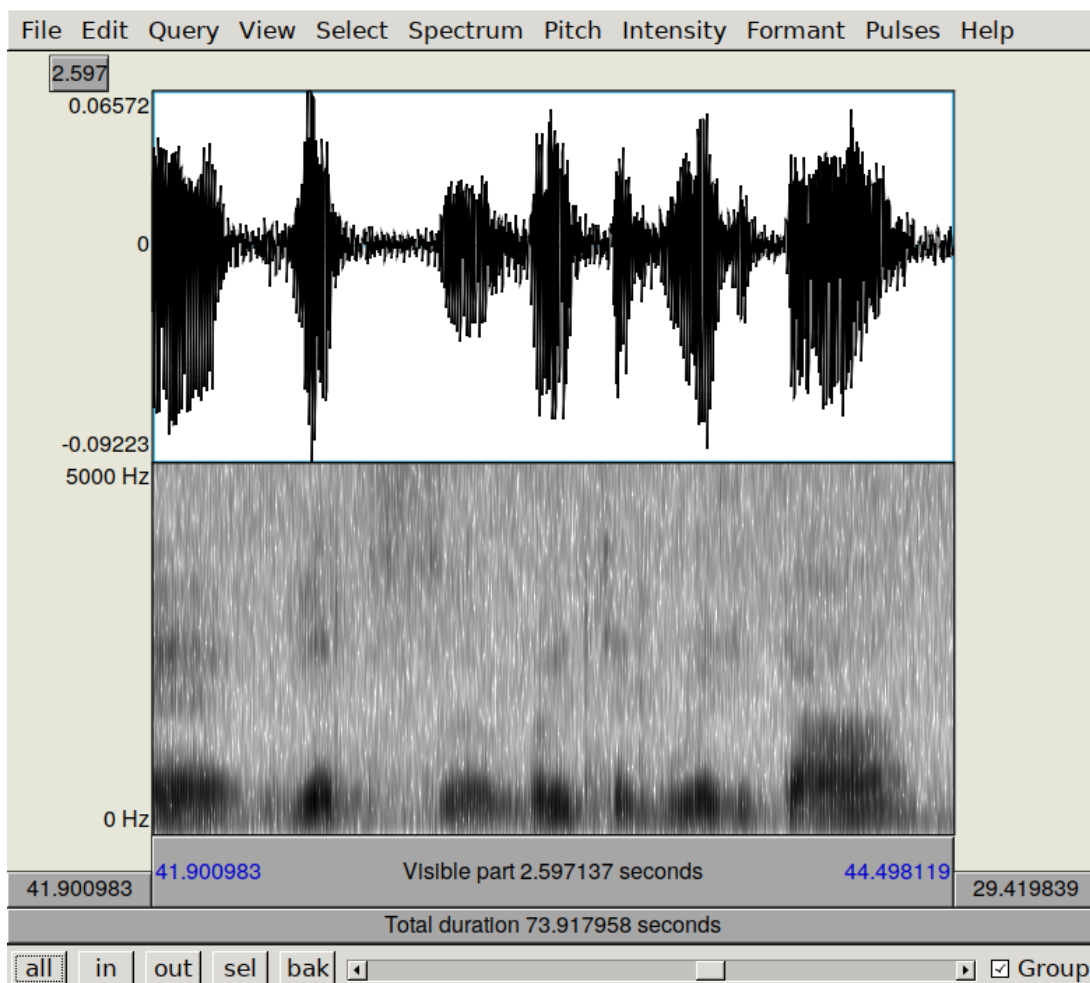
Obrázek 3.3: Širokopásmový šum.

<http://radio.makon.cz/zaznam/92-04A#ts=691.37>

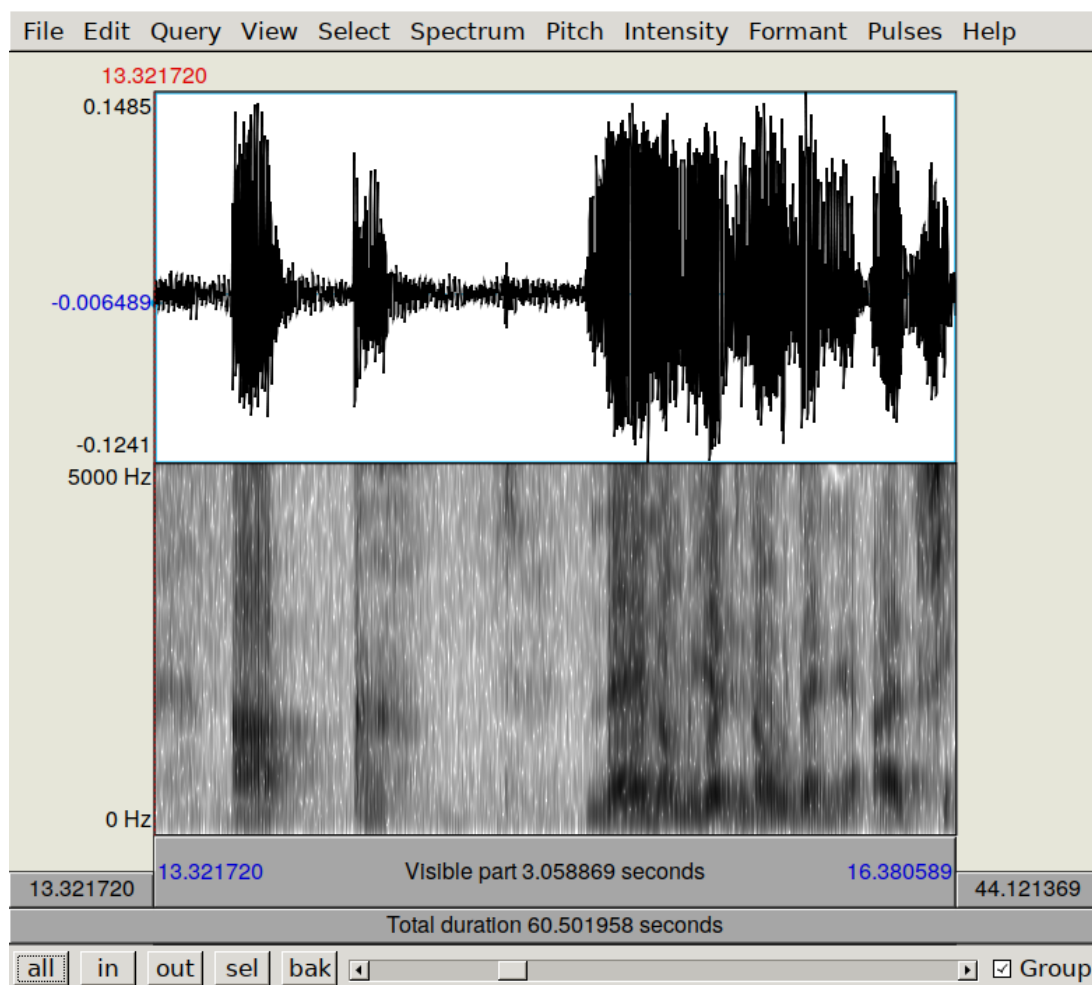


Obrázek 3.4: Úzkopásmový šum.

<http://radio.makon.cz/zaznam/92-03B#ts=664.43>

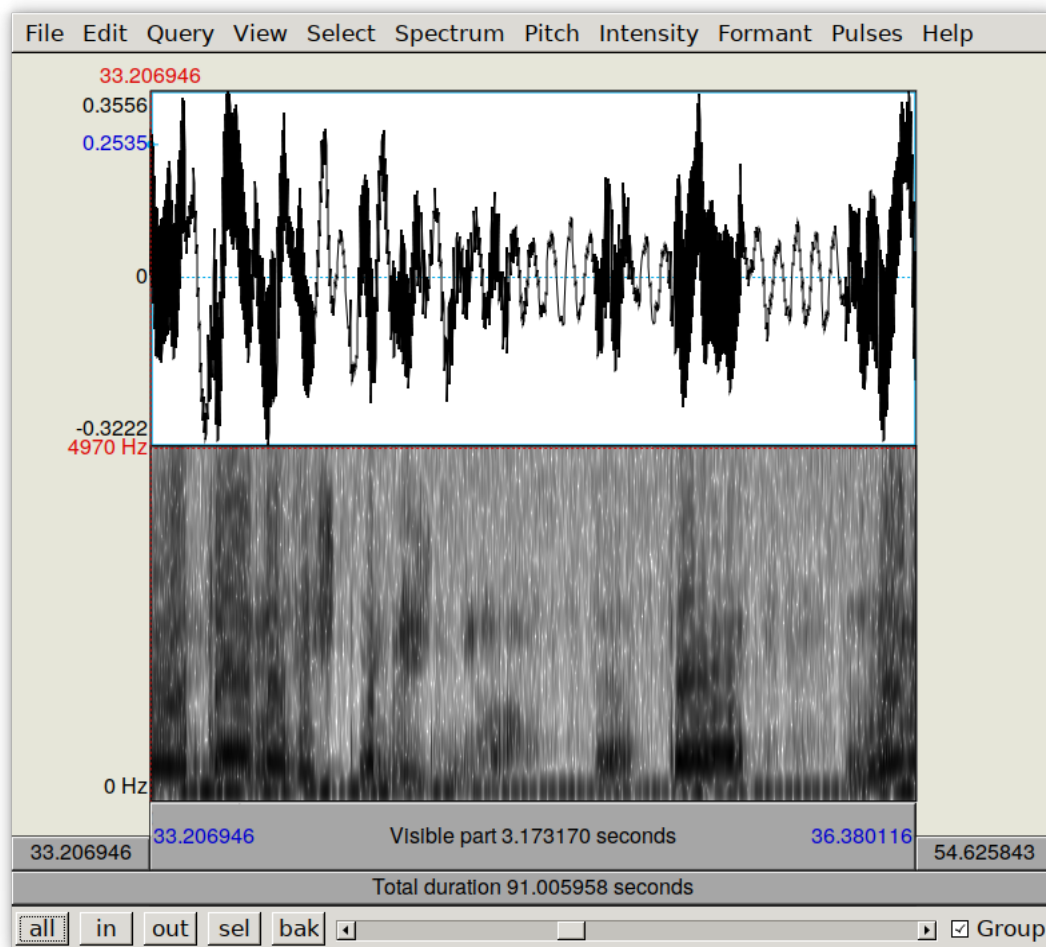


Obrázek 3.5: Absence vysokých frekvencí.  
<http://radio.makon.cz/zaznam/88-04A#ts=678.94>

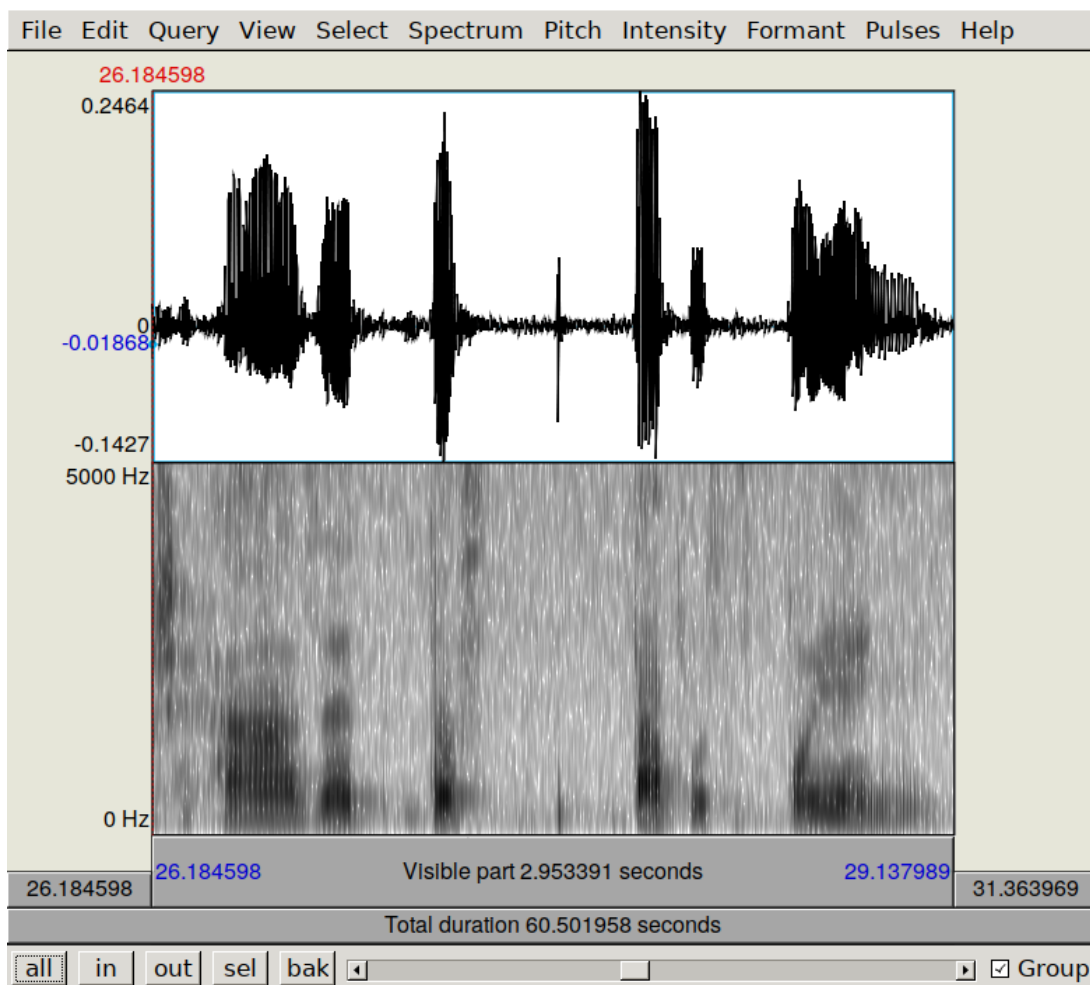


Obrázek 3.6: Zrychlený záznam způsobený zpomalením převíjení pásky při nahrávání.

<http://radio.makon.cz/zaznam/90-18A-XX-zrychlene#ts=2473.56>



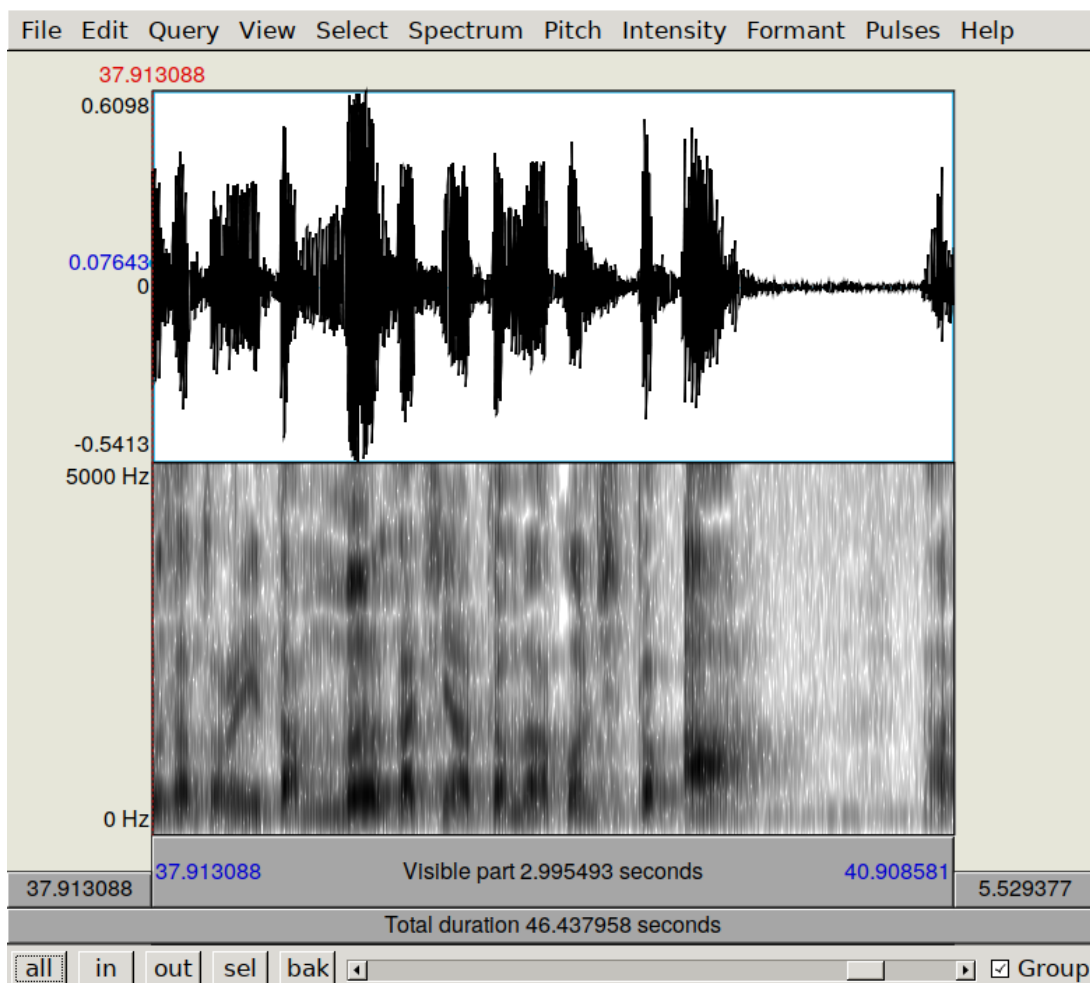
Obrázek 3.7: Silně degradovaná nahrávka pořizená rychlostí 2,38 cm/s.  
<http://radio.makon.cz/zaznam/kotouc-plzen-neident01-a#ts=660>



Obrázek 3.8: Pomalá mluva.

<http://radio.makon.cz/zaznam/76-04A-Kaly-7-IEOUA#ts=13.79>





Obrázek 3.9: Rychlá mluva.

<http://radio.makon.cz/zaznam/89-11B#ts=203.17>

Metrika se počítá následovně: Z každého porovnávaného zvukového záznamu se pořídí dvacetidimenzionální melfrekvenční kvestrální koeficienty. K nim se najde pomocí E-M algoritmu normální distribuce, která je generuje. Na těchto distribucích se pak spočte symetrizovaná Kullback-Leiblerova divergence, tedy  $KL(a|b) + KL(b|a)$ .

V rámci nahrávek dochází často ke změnám akustických vlastností. To je dáno hlavně tím, že se nahrávání uprostřed pásky přerušilo a obnovilo se v jiných podmínkách. Není proto žádoucí porovnávat celé nahrávky. Ideální by bylo detekovat akustické zlomy v nahrávkách a korpus přerozdělit ne podle hranic nahrávek, ale podle těchto zlomů.

Pro jednoduchost jsem nahrávky rozdělil do menších úseků a matici akustické vzdálenosti jsem udělal na nich. Pokusil jsem se využít hotových úseků rozdělených v bodech ticha, viz sekci 6.4. Výsledná matice o rozměrech  $80\,000 \times 80\,000$  však trpěla defekty při čtení a zápisu, proto jsem velikost úseků pro tento účel zvětšil na 10 minut a tím dosáhl počtu 8146 úseků.

Pro orientaci uvedu některé údaje z matice akustické vzdálenosti. Medián vzdálenosti je 55,8. Maximální vzdálenost je  $3,40 \cdot 10^{38}$ , ovšem ta nastává v okrajových případech, bez zjevného důvodu patrného lidskému uchu. Bez této astronomické maximální vzdálenosti dosahují vzdálenosti hodnot do 27 814. Počet nulových vzdáleností je 275 z celkových 33 174 585 a skutečně odpovídají duplicitním úsekům.

Abychom tato čísla mohli interpretovat, porovnejme je se vzdálenostmi jiných zvuků, které si snad čtenář dokáže představit. Provedl jsem pro zajímavost porovnání 1) řeči tří mluvčích ze záznamu jednání Poslanecké sněmovny, 2) řeči a ruchu na pozadí z filmu a 3) řeči a populární hudby.

Tři různé mluvčí jednoho záznamu jednání Poslanecké sněmovny Parlamentu České republiky mají vzdálenost v rozmezí 6,5 až 9,5. Dva různé úseky téhož mluvčího mají vzdálenost 1,4. Tyto záznamy z PSP ČR jsou i pro lidské ucho velmi podobné a na rozdíl ve vzdálenosti mezi různými mluvčími oproti vzdálenosti v rámci jednoho mluvčího je vidět, že algoritmus funguje dobře.

Jako druhý příklad vezměme typickou, úsměvně známou ukázkou mluveného slova s hlukem na pozadí, a sice úryvek „to je dost, žes nás taky jednou vyvez, žes udělal něco pro rodinu“ z filmu Slavnosti sněženek. Jednotliví mluvčí (Blážena Holišová a Rudolf Hrušínský) mají vzájemnou vzdálenost 13,9. Holišová od samotného malotraktoru 48,2 a Hrušínský od téhož 23,7.

Srovnejme ještě mluvenou řeč s populární hudbou. Uvedu dva extrémní příklady, na které jsem narazil: Skladba Billie Jean od Michalea Jacksona má od řeči poslance Sklenáka vzdálenost 23,1, zatímco refrén skladby Shadow Sun od metalové skupiny Moonspell od řeči poslance Okamury 519.

Je tedy patrné, že akustická variabilita korpusu Karla Makoně je i podle tohoto měřítka obrovská.

### 3.3 Shlukování

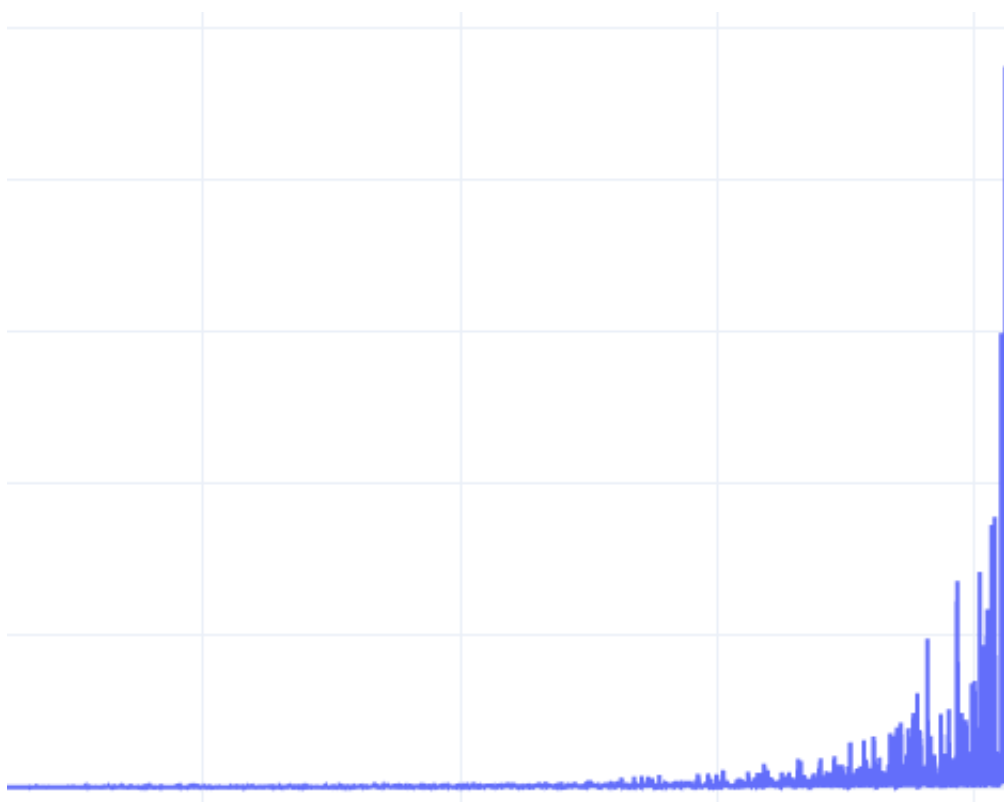
Kompenzovat akustické nedostatky znamená měnit akustické kvality dat tak, aby lépe odpovídaly nějakému kritériu. To může být subjektivní: aby se zvuk určitému posluchači lépe poslouchal. Může být také objektivní, strojově vyhodnotitelné, což pak umožní nasazení strojových metod. Jako vhodné objektivní

kritérium lze zvolit akustickou vzdálenost k záznamům kýžených kvalit.

V korpusu Karla Makoně se nevyskytuje asi žádná nahrávka ve vysoké kvalitě srovnatelné s materiálem pořízeným ve studiových podmínkách. Velká část nahrávek je však veskrze srozumitelná a bez zásadních defektů. Z celého korpusu jsem ručně vybral množinu 431 souboru v uspokojivé kvalitě.

Tato referenční množina je mnohem konzistentnější co do akustické metriky než celý korpus, ale v porovnání s ilustračními příklady mimo korpus stále řádově divergentnější. Vzdálenosti se pohybují od 1,6 do 14 653 s průměrem 54,7 a mediánem 9,40.

Na množině úseků, které jsem porovnával akustickou metrikou, jsem provedl hierarchické shlukování[19]. Vzdálenost clusteru jsem nastavil jako maximální vzdálenost mezi dvěma prvky, aby jednotlivé clustery byly co nejkompaktnější. Průběh shlukování ilustruje obrázek 3.10.



Obrázek 3.10: Velikosti clusterů během hierarchického shlukování.

## 3.4 Kompenzace

Akustické nedostatky ztěžují rozpoznávání řeči a jsou tak už dlouho předmětem bádání. Ku příkladu Gillespie a Atlas (2002)[20] ukazují zdrcující vliv dozvuku (angl. *reverberation*) na rozpoznávání řeči pomocí markovovských modelů a zkoumají možnosti kompenzace. Jošioka et al. (2012)[21] předkládají souhrn technik pro kompenzaci dozvuku, Ko et al. (2017)[22] digitálně simulují dozvuk v trénovacích datech. Seltzer et al. (2013)[23] se věnují tematice šumu v rozpoznávání řeči založeném na hlubokých neuronových sítích.

### 3.4.1 Spektrální odečet šumu

Jako baseline svého druhu jsem se pokusil automatizovaně aplikovat zavedenou metodu zvanou redukce šumu, *noise reduction*. Pro jednoduchost zde předpokládám, že každá nahrávka, tedy soubor o délce cca. 45 - 90 minut, trpí konstantním stacionárním šumem. To pro všechny nahrávky neplatí, ale pro ověření způsobilosti metody to není podstatné. Pro redukci šumu jsem použil program *sox*. Metoda spočívá pro každou nahrávku v těchto krocích:

1. Identifikovat a izolovat vzorek čistého stacionárního šumu,
2. extrahovat profil šumu a
3. aplikovat redukci šumu na základě získaného profilu.

O body 2 a 3 se postará *sox*. Co se týče získání vhodného vzorku šumu, vyvinul jsem následující metodu:

1. Určit a extrahovat všechna predikovaná ticha za použití zarovnaného automatického přepisu.
2. Seřadit ticha sestupně podle délky a vybrat jich 100 kolem 25. percentilu. Tak se zajistí, že se nepoužijí ani příliš krátká ani příliš dlouhá ticha. V dlouhých bývají nestacionární ruchy, proto se jim vyhýbám.
3. Pomocí programu musly vygenerovat matici vzdáleností na tiších.
4. Vybrat deset tich s nejmenším mediánem na vzdálenostní matici. Tato ticha jsou nejvíce podobná ostatním, a tím pádem s nejmenší pravděpodobností obsahují nestacionární události.
5. Konkatenovat vybraná ticha.

Subjektivní vyhodnocení potvrzuje očekávaný výsledek, že relativně kvalitní záznamy, které trpí pouze trochou aditivního šumu, se po redukci lépe poslouchají. Záznamům trpícím jinými defekty a celkově nižší kvalitou je někdy po operaci hůře rozumět.

Pro kvantitativní vyhodnocení jsem natrénovat systém rozpoznávání řeči, jenž popisují v kapitole 5, na datech po odstranění šumu. Chybovost na slovech vzrostla skoro na sto procent, což znamená, že systém zcela přestal fungovat. Přesnou příčinu zatím neznám. V tabulce 3.1 uvádím chybovost modelu natrénovaného na původních datech a datech po redukci, testovaného opět na původních datech a datech po redukci šumu.

WER	původní model	model po redukci
původní testovací sada	19,2%	94,0%
sada po redukci šumu	69,8%	94,1%

Tabulka 3.1: Word error rate při trénovacích i testovacích datech před redukcí šumu a po ní.

### 3.4.2 Neurální doménový transfer

Revoluční článek[24], v němž Žú et al. (2017) představují CycleGAN, čili cyklicky konzistentní generativní oponentní síť, dal lidstvu do rukou mocný nástroj a zábavnou hračku, která našla využití pro odstraňování mlhy z fotografií[25], udělování cizích grimas tvářím[26], v biomedicíně[27] a také pro zpracování mluvené řeči: Kaneko a Temeoka (2017)[28] předkládají doménový transfer hlasu a Hoseini-Asl et al. (2018)[29] činí totéž pro účely rozpoznávání řeči. Nejblíže mému problému je Pascual et al. (2017) s projektem SEGAN[30], kde se GAN používá pro odstranění šumu.

CycleGAN je vytvořena pro použití na dvou sadách dat, z nichž každá reprezentuje určitou doménu, mezi nimiž lze najít mapovací funkci. V případě korpusu Karla Makoně je jedna jasná doména akusticky relativně dobrých nahrávek a potom celý zbytek, který ovšem konzistentní doménu netvoří. Poškozené nahrávky trpí různými kombinacemi neduhů v různé míře. Nelze proto CycleGAN přímo aplikovat na „zdravé“ a „poškozené“ nahrávky. Jak by taky bylo lze nalézt funkci mapující nahrávku bez neduhů na poškozenou, když není jasné, jaké poškození by měla vykazovat? Tento směr sice není kýžený, ale pro natrénování dané architektury nezbytný.

Adaptace GAN tak, aby si s nekonzistentními daty na jedné straně převodu poradila, by jistě byla zajímavým a přínosným počinem, nicméně začít se dá využitím výše zmíněného shlukování, které poskytuje potřebné konzistentní domény poškozených nahrávek.

Experiment jsem provedl na dvou shlucích: 1) na přebuzených nahrávkách a 2) na nahrávkách pořízených nízkou rychlostí 2,38 cm/s. Oba shluky jsem vybral tak, aby měly maximální interní vzdálenost 25. Pro trénování jsem použil metodu navrhanou dvojicí Kaneko a Tameoka (2017)[28], jak ji implementoval Lei Mao<sup>2</sup>. Trénink běžel po 200 epoch.

### 3.4.3 Vyhodnocení

Tabulka 3.2 uvádí WER při automatickém přepisu na tom kterém shluku původně a po transferu. Toto porovnání bylo provedeno pomocí modelu natrénovaného pouze na promluvách Karla Makoně. Robustnější model popsáný v sekci 5.12 má chybovost na nízkorychlostních nahrávkách 42,1% a na nahrávkách přebuzených 34,8%.

	původní	po transferu
přebuzené	45,0%	44,1%
nízkorychlostní	68,5%	93,9%

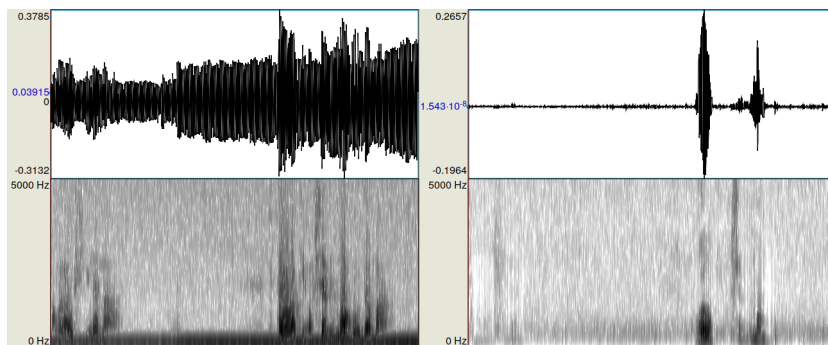
Tabulka 3.2: Word error rate u dvou skupin poškozených nahrávek před doménovým transferem a po něm.

Mírné umenšení chybovosti u přebuzených nahrávek je statisticky nevýznamné a nepřináší kýžené řešení problému s neuspokojivými výsledky rozpoznávání poškozených nahrávek. Snad důležitější je však zvýšený komfort při poslechu. Bo-

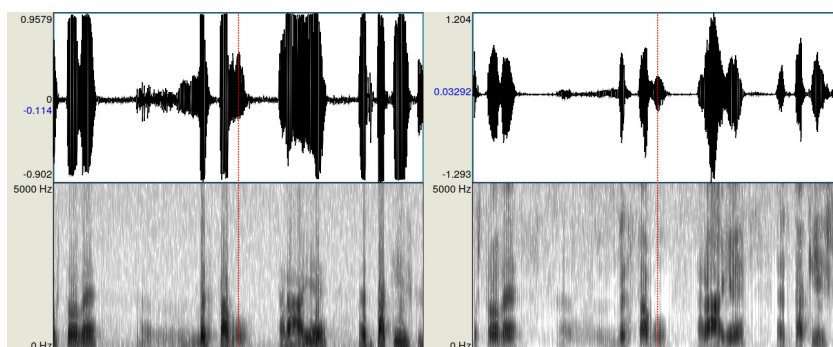
<sup>2</sup>[github.com/leimao/Voice\\_Converter\\_CycleGAN](https://github.com/leimao/Voice_Converter_CycleGAN)

hužel pro tento přínos zatím nemám kvantitativní vyhodnocení, ale subjektivně ho potvrdit mohu.

U katastrofálního zvýšení chybovosti po transferu velmi poškozených nahrávek pořizovaných nízkou rychlostí je nutno se pozastavit. Obrázek 3.11 ukazuje, jak se u těchto nahrávek doménový transfer odrazil v průběhu signálu a ve spektru. Obrázek 3.12 ukazuje pro porovnání totéž u přebuzených nahrávek. Povšimněme si, jak u nahrávek pořizených nízkou rychlostí po přesunu zcela zmizela některá slova. Je-li signál příliš těžko odlišitelný od ruchů, transfer patrně raději změní takový úsek v ticho.



Obrázek 3.11: Průběh signálu (nahore) a spektrogram (dole) nahrávky pořizené rychlostí 2,38 cm/s před doménovým transferem (vlevo) a po něm (vpravo).



Obrázek 3.12: Průběh signálu (nahore) a spektrogram (dole) přebuzené nahrávky před doménovým transferem (vlevo) a po něm (vpravo).

## 4. Jednání parlamentu jako trénovací data

Trénovacích dat pro rozpoznávání řeči není nikdy dost. V době psaní tohoto textu je manuálně přepsaných asi 100 hodin z mluveného korpusu Karla Makoně. To je pro natrénování modelu pro jednoho mluvčího použitelné množství. Nabízí se však otázka, zda by více trénovacích dat od jiných mluvčích mohlo pomoci.

Veřejně je k dispozici několik zdrojů dat pro účely trénování rozpoznávače češtiny:

- Vystadial[31] se 77 hodinami záznamů internetových rozhovorů[32],
- The Prague Database of Spoken Czech[33] se 122 hodinami spontánních dialogů anotovaných na několika úrovních[34],
- Korpus expresivní mluvy COMPANION s 5 hodinami namluvenými jednou profesionální mluvčí[35],
- Otázky Václava Moravce: 35 hodin přepsaných záznamů české talk show[36],
- STAZKA: 35 hodin záznamů z vozidel na silnicích obsahujících anotované promluvy[37],
- 88 hodin automaticky přepsaných záznamů z jednání poslanecké sněmovny [38].

Celkem se tak dostaneme přibližně na 350 hodin dalších trénovacích dat.

Ze záznamů jednání poslanecké sněmovny však existují také ruční stenografické přepisy. Velká část jednání a jejich přepisů je veřejně ke stažení na webových stránkách poslanecké sněmovny. Pokusil jsem se proto z těchto dat připravit korpus pro trénování rozpoznávání řeči.

Tvorbu tohoto korpusu popisuje článek Krůza (2020)[39]. Paralelně se mnou do podoby trénovacích dat pro rozpoznávání řeči upravili parlamentní záznamy i Kratochvíl et al. (2020)[40]. Jejich výsledkem je korpus o velikosti 444 hodin, oproti mým 1058 hodinám. Chybovost na parlamentních záznamech samotných je u zmíněného článku (7.10%) srovnatelná s mojí (7.89%). Tvorbu korpusu, který na tento navazuje, nikoliv jako primárně nástroj pro trénování rozpoznávačů řeči, nýbrž jako anotovaný korpus k obecnému lingvistickému bádání, popisuje článek, který má vyjít v září roku 2021 na konferenci TSD[41]. V současnosti spolupracuji na dalším vydání parlamentního korpusu, kde se technologie popsaná v této kapitole využívá a rozvíjí.

### 4.1 Příprava dat

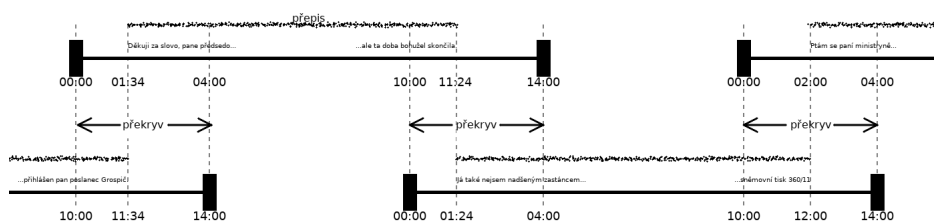
Pokud je mi známo, jsou ruční přepisy jednání Poslanecké sněmovny k dispozici pouze ve formátu čitelném pro člověka. Přepisy nejsou ve zdrojovém kódu webové stránky nijak oddělené a jsou promíchané s metainformacemi. Je tedy nutné extrakci pojmout jako aproximační úkol. Používám velice jednoduchý algoritmus, který má své nedostatky, ale pokrývá drtivou většinu záznamů. Extrahuji

podstrom všech elementů s hodnotou atributu `[align=justify]` vyjma elementů `<b>`, neboť ty obsahují jména mluvčích.

Znamé nedostatky spočívají jednak v tom, že se jména mluvčích sice správně z přepisu oddělují, ale navzdory jejich hodnotě coby metainformace zahazují, a jednak v tom, že se z přepisu vynechávají odkazy na jiné schůze. Ty jsou totiž formátované jinak než ostatní části, jak je vidět např. v přepisu schůze z 12. února 2020 od 10:10<sup>1</sup>. Oprava obého je otázkou napsání chytřejšího scraperu a pro účely vybudování korpusu pro trénink rozpoznávání řeči je obé bezvýznamné: Označení mluvčích z principu, vynechání odkazů pro jejich řídkost.

### 4.1.1 Zarovnávání

Jedna z překážek použití stenografického přepisu pro trénování rozpoznávačů řeči je velmi volné párování neboli zarovnání přepisů ke zvuku. Každý zvukový záznam má 14 minut a se sousedními se na každé straně překrývá vždy čtyři minuty. Přepisy jsou rozdělené na úseky odpovídající těmto záznamům. Zarovnání je tedy do desetiminutových bloků s dvouminutovým přesahem na každé straně. Na obrázku 4.1 je vyobrazeno schéma tohoto apriorního zarovnání.



Obrázek 4.1: Apriorní zarovnání a překryv zvukových záznamů k přepisům. Vyobrazen je záznam z 12. února 2020 kolem 10. hodiny. Přepis záznamu vlevo nahoře pokrývá pozice od 01:34 do 11:24. Vpravo dole pak od 01:24 do 12:00.

Systémů pro zarovnávání dlouhých zvukových záznamů existuje několik, publikovali je např. Moreno et al. (1998)[42] nebo Hazen (2006)[43]. Oba jsou založeny na využití předem získaného a zarovnaného automatického přepisu. Taktéž tento přístup využívám, ale zjednodušený a přizpůsobený úloze.

Za použití výše zmíněného datasetu[38] jsem natrénoval markovovský akustický model obdobný tomu, jenž je popsán v kapitole 5. Jazykový model jsem natrénoval ze stažených stenografických přepisů. Těch je podstatně víc než nahrávek, protože z mně neznámého důvodu je valná část odkazů na zvukové záznamy nefunkčních, končíc chybovým kódem 404 nebo v menším počtu případů 403.

Pro celý korpus jsem pomocí programu julius vygeneroval automatický přepis se zarovnáním. Automaticky vygenerovaný zarovnaný přepis každého záznamu jsem pak porovnal s odovídajícím manuálním přepisem pomocí Levenshteinovy metody počítání editačních operací nad písmeny. Zjistil jsem samotné editační operace pro přechod z automatického přepisu k manuálnímu a pro každé slovo v automatickém přepisu spočetl, kolik úprav naň připadá. Na základě toho definuji pro každé automaticky vygenerované slovo spolehlivost párování se slovem

<sup>1</sup><https://www.psp.cz/eknih/2017ps/stenprot/040schuz/s040372.htm>



manuálně zapsaným jako

$$1 - \frac{e(w)}{l(w)}, \quad (4.1)$$

kde  $e(w)$  je počet editačních operací nad slovem  $w$  a  $l(w)$  je délka slova  $w$  v písmenech.

Obrázek 4.2 zachycuje proces zarovnání manuálních přepisů se zvukovým záznamem.



Obrázek 4.2: Schéma zarovnání zvukových záznamů ke stenografickým přepisům na úrovni slov.

### 4.1.2 Tvorba potenciálních trénovacích vzorků

Kvalita trénovacích dat pro rozpoznávání řeči závisí i na tom, jak jsou rozdělena. Za prvé je žádoucí, aby byly trénovací vzorky podobně dlouhé jako testovací vzorky[44]. V případě automatického přepisu korpusu Karla Makoně lze nastavit délku vstupních úseků libovolně. V obecném případě nelze předvídat. Za druhé je pro trénink výhodné, aby jednotlivé vzorky měly podobnou délku kvůli efektivnímu využití operační paměti grafické procesní jednotky. Stačí jeden dlouhý vzorek a dávka (*batch*) způsobí vyčerpání paměti. Naopak mnoho kratoučkových vzorků způsobí, že se paměť při dávce zaplní jen málo a plýtvá se časem. Za třetí, chci-li, aby byla trénovací sada použitelná i pro ostatní, je dobré, aby se délkou vzorků příliš neodlišovala od ostatních datových sad.

Nad to je ale důležité, aby přepis dokonale odpovídal obsahu. A protože vyřezáváme trénovací vzorky z delších souborů, při čemž může dojít k nepřesnostem, je záhodno volit místa řezu tak, aby padla pokud možno do delších pauz v řeči. Je to týž problém jako v sekci 6.4, kde je podrobně popsán. V tomto případě jsem dospěl k hranicím 12 - 30 sekund a data tak rozdělil na úseky o délce v tomto rozpětí.

### 4.1.3 Výběr trénovacích vzorků

Po rozdělení desetiminutových nahrávek na úseky vhodné délkou pro trénink je nutné spárovat tyto úseky s odpovídajícími úseky manuálních přepisů, a ty, které jsou spárovány spolehlivě, zařadit do samotné trénovací množiny. V podsekci 4.1.1 jsem popsal, že máme párování jednotlivých slov v automatickém a manuálním přepisu s určitou měrou spolehlivosti a že automatický přepis je spárován se zvukovým záznamem. Zbývá tedy vybrat úseky, které můžeme považovat za spolehlivě přepsané, a zbytek vyřadit.

Nahrávky se překrývají tak, že z každého čtrnáctiminutového souboru je jen deset minut pokryto odpovídajícím přepisem, takže vyřadit musíme minimálně 29% úseků. Dospěl jsem k následujícím kritériím:

1. Spolehlivost prvního a posledního slova v úseku alespoň 70%.
2. Průměrná spolehlivost všech slov v úseku alespoň 70%.
3. Alespoň 5 slov v úseku.

Důvodem k zahrnutí kritéria minimální spolehlivosti krajních slov je záměr zajistit správné hranice úseku. Druhé kritérium počítá s průměrnou spolehlivostí a ne třeba s minimem, protože je přípustné, aby některá slova měla i nulovou spolehlivost, tedy aby v nich všechna písmena byla špatně. Iniciální přepis obsahuje mnoho chyb, to je také důvod, proč se trénuje s manuálním přepisem a ne s automatickým. Je-li ale příliš mnoho slov spárováno nespolehlivě, roste šance, že spárování je skutečně liché. Počet slov v úseku je mezi kritérii proto, že u malého počtu slov je vysoká šance náhody, při které je chybně určeno vysoké skóre spolehlivosti nesprávně spárovaným slovům.

Pro potvrzení tvrzení, že slova s nulovou spolehlivostí jsou přípustná, uvedu příklad: slova „*finanční úřady dotace*“ se v jednom případě přepsala jako „*finanční u jít dotace*“. Slova „*u*“ a „*jít*“ mají obě se slovem „*úřady*“ prázdnou množinu společných písmen, a tedy nulovou spolehlivost. Přesto je obklopující fráze jako celek zarovnána správně.

Ještě vyvstává otázka, proč použít průměr a ne medián spolehlivosti v druhém kritériu. Je to z toho důvodu, že liší-li se výrazně počet slov v manuálním a automatickém přepisu, projeví se to jako mnoho operací přidání či smazání písmene na jednom slovu v automatickém přepisu. V takovém případě mohou všechna slova mít stoprocentní spolehlivost, jen jedno hluboce pod nulou. Takový úsek by pak byl při použití mediánu přijat, zatímco průměr výrazně negativní hodnotu započte a úsek správně odmítne.

#### 4.1.4 Shrnutí extrakce trénovacích dat

Konstanty a algoritmy použité při extrakci trénovacích dat ze záznamů jednání Parlamentu České republiky jsou jen hrubě zvoleny a je velký prostor pro jejich odladění. Vedou ale už teď k velmi kvalitní datové sadě o velikosti 1058 hodin. Z celkového počtu 539 057 úseků jich bylo 142 530 (26%) zahrnuto do trénovací sady. Z celkového počtu 396 527 zavržených úseků jich 350 258 (88%) bylo zavrženo kvůli nespolehlivým hraničním slovům. Toto kritérium je však aplikováno jako první, takže je v tomto čísle zahrnuto i mnoho úseků, které by jinak byly odmítnuty některým dalším kritériem.

Sníží-li se potřebná spolehlivost ze 70% na 50%, zvýší se počet přijatých úseků o 17%. Přidá se tak 5% úseků z celkového počtu. Pokud však započteme fakt, že 29% úseků je nutně odstraněno kvůli překryvům, je celkový přírůstek ve skutečnosti 9% celkového počtu. Je to možnost, jak zvýšit objem trénovacích dat za cenu zvýšení počtu úseků se špatně určenými hranicemi.

## 4.2 Číslovky a zkratky

Číselných výrazů je v parlamentních prepisech mnoho. Představují 489 880 ze 25 010 269 tokenů v kompletním stenografickém prepisu, to jsou téměř dvě procenta. Ve výše popsaných trénovacích datech je aspoň jedno číslo ve 24% vzorků.

Původní moje řešení spočívalo v zahrnutí číslic do abecedy a tedy v pokusu o přepis číselných výrazů přímo na číslice. V systému rozpoznávání řeči natrénovaném na těchto datech, popsaném v následující sekci, bohužel výsledkem bylo, že číselné výrazy se vždy přepsaly na prázdný řetězec.

K problému se lze postavit čtyřmi způsoby:

1. ignorovat ho,
2. vyřadit číslice z trénovacích dat,
3. manuálně číslice rozepsat do slov,
4. číslice rozepsat automaticky.

První možnost ignorování problému asi netřeba rozebírat. Vyřazení vět s číslicemi je snadný a použitelný přístup, ale byla by škoda přijít o čtvrtinu trénovacích dat a o drtivou většinu příkladů číslovek. Manuální přepis by byl jistě ideální, ale vzhledem k objemu dat pro mne nerealizovatelný. Zbývá se tedy pokusit o čtvrtou možnost.

Pro automatický rozpis číslic do slov lze využít hotového aparátu: iniciálního přepisu a algoritmu pro zarovnávání se stenografickým prepisem.

Rozpis provádím ve dvou krocích:

1. vygenerování možných čtení číslicového výrazu,
2. výběr nejpravděpodobnější varianty.

Pro generování možných čtení čísel jsem vyšel z algoritmu použitého v modulu pro Perl `Lingua::CS::Num2Word`, který jsem doplnil o řád miliard pro kardinální číslovky, rozšířil tak, aby se místo jedné varianty generovaly pokud možno všechny gramaticky přípustné, a zahrnul podporu genitivu a akuzativu, decimálních a ordinálních číslovek, dat a hodin.

Ve stenografických prepisech před dalším zpracováním rozvedu všechny tokeny obsahující číslice do variant rozpisu a při zarovnávání s iniciálním automatickým prepisem vyberu tu variantu, která má nejmenší editační vzdálenost.

Spolu s číslicemi expanduji také zkratky a symboly. Např. velmi častý symbol paragraf (§) rozepisují do variant *paragraf*, *paragrafu*, *paragrafů*, *paragrafem*, *paragrafech*, které se podle iniciálního přepisu vyskytují jako jediné časté. Dalšími častými zkratkami s různými variantami rozpisu jsou *čl.* – *článek* / *člátku* / ..., *odst.* – *odstavec* / *odstavce* / ..., *tzv.* – *takzvaný* / *takzvaného* / ...

Po provedení expanze se podobnost stenografického a iniciálního automatického přepisu zvýšila, což se odrazilo i na zvýšení počtu přijatých úseků z 26% na 35%. Množství trénovacích dat v hodinách vzrostlo o 86, tedy na 1144.

O využití datové sady pro rozpoznávání řeči pojednává sekce 5.11.

## 5. Automatický přepis

Koncept celého projektu se zakládá na přítomnosti počátečního přepisu a jeho následném zdokonalování. Je tedy nutné opatřit způsob, jak korpus strojově převést do textové podoby. Tím se dostáváme do oblasti z historických důvodů zvané *automatické rozpoznávání řeči*, anglicky *automated speech recognition*, zkracované jako *ASR*, ačkoliv z povahy věci by přesnějším výrazem byl *automatický přepis*.

Automatický přepis lze chápat jako úlohu transformace signálu nebo jako rozpoznávání vzorců v datech. Algoritmus takových úloh je obvykle velice složitý, neboť variabilita vstupních dat je značná a také zadání je inherentně vágní: mnohdy je i pro člověka obtížné rozeznat, jakou hlásku nebo jaké slovo určitý zvuk představuje.

Řešení problému, jak systému předat potřebnou znalost mapování vstupních dat na kýžený výstup, se řeší kombinací dvou přístupů: přímým zakomponováním lidské expertizy a strojovým učením z dat. Pojem strojové učení vznikl už v roce 1959[45] a obor se od té doby rozvíjí a získává na popularitě. Už legenda strojového zpracování přirozeného jazyka kladenský rodák Bedřich Jelínek, lépe známý pod svým poangličtěným jménem Frederick Jelinek, proslul svým výrokiem, že kdykoliv vyhodí lingvistu, zvýší se mu úspěšnost systému.

Ve strojovém učení pracujeme s pojmy *trénovací data* a *testovací data*. V obou případech se jedná o správně spárovaná vstupní a výstupní data. V případě ASR tedy jde o páry záznamů mluveného slova a odpovídajících přepisů. Trénovací data jsou ta, která má systém k dispozici, aby z nich odvodil znalost potřebnou pro danou úlohu, tedy v našem případě pro převod zvuku do textu. Testovací data jsou potom ta, na nichž se vyhodnocuje úspěšnost systému.

Toto vychází z předpokladu, že v trénovacích datech je zobecnitelná informace, která platí i o testovacích datech. Také se vychází z předpokladu, že testovací data jsou od trénovacích striktně oddělena, neboť jedině tak lze na základě výsledku testování vyvozovat relevanci i pro další, neznámá data.

V mém případě je úloha specifická tím, že se nesnažím vyvinout systém na univerzální predikci neznámých dat: Jedná se mi o přepis daného korpusu, který mám celý k dispozici. Karel Makoň je po smrti, takže dalších nahrávek se od něho nenadějeme. Testovací sada oddělená od trénovací je však podstatná pro vyhodnocení, ze kterého se dá soudit o výkonu na částech korpusu, pro který chybí ruční přepis.

Výhoda strojového učení oproti ručním pravidlům je zřejmá: v datech je informace uložena objektivně, pravdivě. Oproti tomu v ručně psaných pravidlech je zanesen lidský faktor, který s sebou nese předpojatost, omylnost a nepřesnost. Strojový čas je navíc mnohdy dostupnější než lidský.

Nevýhodou strojového učení je, že je k němu zapotřebí trénovacích dat, a to čím více, tím lépe, aneb „there is no data like more data“<sup>1</sup>. Je známo, že neuronová síť i hloubky 1 je s dostatečným počtem neuronů schopna modelovat jakoukoliv spojitou funkci s omezeným definičním oborem (Csáji 2001)[46]. Kdybychom tedy měli neomezeně mnoho kvalitních trénovacích dat a výpočetní kapacity, nebylo by důvodu aplikovat lidskou expertizu.

Jelikož jsou však zdroje omezené, tyjí systémy typicky z obou zdrojů: z lid-

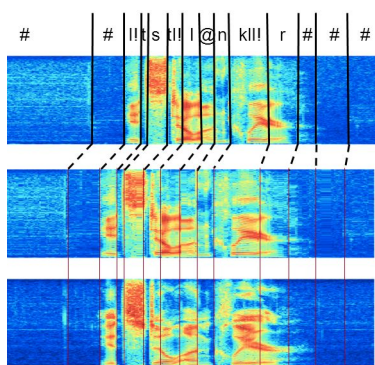
---

<sup>1</sup>Robert Mercer, 1985

ské odbornosti a strojového učení, přičemž historicky s dokonalejšími počítači a větším množstvím dat v systémech ubývá jednoho a přibývá druhého.

## 5.1 Vybrané milníky v rozpoznávání řeči

Již v roce 1952 vyvinuly Bell Laboratories analogové obvody, které dokázaly pro konkrétního mluvčího rozpoznat vyslovené číslice v telefonním přenosu.[47] Systémy v sedmdesátých letech porovnávaly vstupní zvukový záznam s množinou předloh a výstup určili podle nejpodobnější varianty[48]. Porovnávání se šablonami má zásadní nevýhodu v tom, že totéž slovo, ba tatáž hláska může být pokaždé vyslovena jinak rychle, jinými slovy v řeči je velká časová variabilita. Ještě v osmdesátých letech se tato technika namnoze používala a časová variabilita se kompenzovala roztažením či smrsknutím vzorků, aby si délkou odpovídaly. Tato metoda se anglicky nazývá *dynamic time warp* a vznikla v roce 1971[49], viz obrázek 5.1. V těchto přístupech pochází veškerá expertiza ze strany člověka.



Obrázek 5.1: Ilustrace rozpoznávání řeči pomocí šablon technikou dynamic time warp.

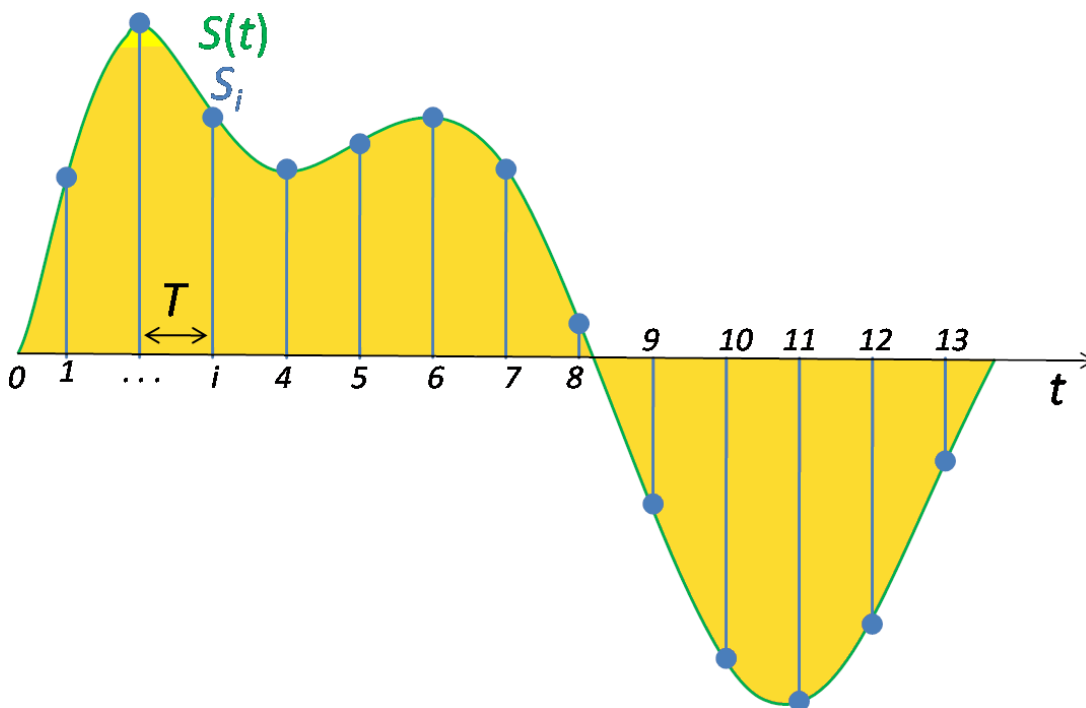
Velkým průlomem bylo použití skrytých markovovských modelů (*Hidden Markov Models, HMM*). Vychází se z představy, že mluvčí předává informaci (posloupnost písmen, či slov) zakódovanou v kanálu akustického vlnění, a naším úkolem je původní informaci rozkódovat. Markovovské modely ve spojení s gaušovskými směsmi (*Gaussian Mixture Models, GMM*) byly dominantní technologií rozpoznávání řeči až do průlomu hlubokých neuronových sítí.

Hluboké učení přineslo zatím poslední skokový posun kupředu. Hluboké neuronové sítě nejdříve vstupují jako součást schématu s HMM, sloužíce pro odhad aposteriorní pravděpodobnosti přepisu na základě akustických dat a nahrazují složku GMM, čímž vznikají hybridní systémy DNN-HMM místo dosavadních GMM-HMM. Posléze se objevují systémy realizované pomocí jedné neuronové sítě, která řeší celou úlohu rozpoznávání.

## 5.2 Kódování signálu

Jedním z nejdůležitějších stavebních prvků v systémech rozpoznávání řeči je předzpracování zvukové vlny do vhodnějšího formátu.

Zvukový signál je velmi prostý: je to spojitá funkce času do reálných čísel. Má-li se zaznamenat digitálně, volí se technika vzorkování (*sampling*), kdy se z průběhu funkce v čase, jehož spojitost či diskrétnost je otázkou mimo rámec této práce, vybere vzorek v rozestupu definované periody, viz obrázek 5.2. Čím větší vzorkovací frekvence, tím 1) větší věrnost při reprodukci, 2) vyšší tónovou frekvenci lze reprezentovat a 3) větší náročnost na uložení a zpracování.



Obrázek 5.2: Vzorkování signálu.  $t$  je čas,  $S(t)$  je průběh signálu,  $T$  je vzorkovací perioda,  $S(i)$  je hodnota  $i$ -tého vzorku.

Vzorkovací frekvence se podle účelu záznamu pohybuje obvykle od 8 kHz pro datově úsporný přenos hlasu do 96 kHz pro studiové zpracování hudby. Pro účely rozpoznávání řeči se osvědčila vzorkovací frekvence 16 kHz jako minimum, ve kterém je obsažena prakticky veškerá relevantní informace.

Jako vstupní data pro skryté markovovské modely nejsou prostá reálná čísla<sup>2</sup> v řádu tisíců za sekundu praktická, pročež se signál nejdříve transformuje do jiné formy. Tato část systému se zve *front-end* a implementuje se nejčastěji pomocí melfrekvenčních keprálních koeficientů (*MFCC*) nebo řidčeji pomocí perceptuální lineární predikce (*PLP*). Účelem není jen transformace do praktičtějšího prostoru, ale též odstranění nerelevantní informace a normalizace té relevantní.

Převod z akustického vlnění do melfrekvenčních keprálních koeficientů[50] transformuje proud jednoho reálného čísla šestnáctkrát za milisekundu do proudu reálněčíselného vektoru stokrát za sekundu. Běžně každý vektor kóduje časové okno o délce čtyřicetiny sekundy a okna jsou od sebe vzdálena setinu sekundy, takže se překrývají, viz obrázek 5.3. Na každém časovém okně se provede diskrétní

<sup>2</sup>Pro počítačovou implementaci jsou reálná čísla díky svému nekonečnému desetinnému rozvoji nereálná, proto se v praxi pracuje s tzv. čísly s plovoucí desetinnou čárkou (*floating-point numbers*).

Fourierova transformace

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \quad (5.1)$$

a rozdělí se do frekvenčních oken podle škály mel[51] aplikováním trojúhelníkového filtru

$$Y_{t,m} = \sum_{k=1}^N W_m(k) |X_{t,k}|^2 \quad (5.2)$$

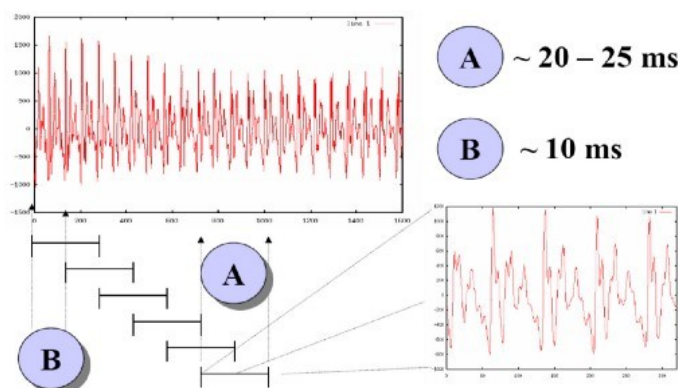
kde  $W_m$  je  $m$ -tý trojúhelníkový filtr a  $k$  je index ve výsledném vektoru DFT, viz obrázek 5.4. Frekvenčních oken bývá 12 a opět se překrývají. Frekvenční okna se s rostoucí frekvencí zvětšují tak, aby zůstávala konstantní v jednotkách mel a aby tedy odpovídala lidskému vnímání spektra. Hodnoty se logaritmují, opět aby byly úměrnější lidskému vnímání hlasitosti. Jako třináctá hodnota se přidá buď základní frekvence nebo častěji akustická energie.

$$E = \sum_{t=t_1}^{t_2} x_t^2 \quad (5.3)$$

Poté se provede inverzní diskretní Fourierova transformace. Od těchto hodnot se na základě kontextu přidá první a druhá derivace,

$$d(t) = \frac{c(t+1) - c(t-1)}{2} \quad (5.4)$$

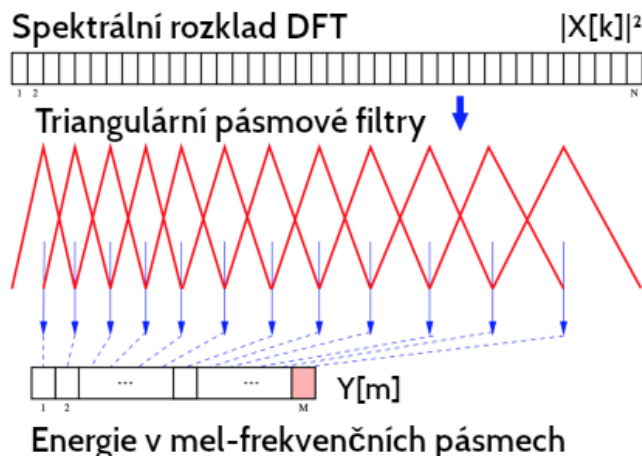
čímž získáváme devětatřicetirozměrný reálněčíselný vektor každou setinu sekundy.



Obrázek 5.3: Schéma překrývajících se oken při převodu z vlnového průběhu do melfrekvenčních keprálních koeficientů.  $A$  je šířka okna,  $B$  je rozestup mezi okny.

## 5.3 HMM

V této sekci stručně představím architekturu systémů rozpoznávání řeči založenou na skrytých markovovských modelech, jak se běžně používala ještě na začátku tohoto tisíciletí a této disertace.



Obrázek 5.4: Schéma aplikace filtrů do pásem podle stupnice mel.

Úlohu automatického rozpoznávání řeči můžeme formalizovat takto: Na základě vstupní posloupnosti hodnot akustického vlnění  $A$  hledáme takovou posloupnost slov (*words*)  $W$ , aby pravděpodobnost  $P(W|A)$  byla co největší, tedy:

$$\operatorname{argmax}_{\tilde{W}} P(W|A) \quad (5.5)$$

Pravděpodobnost  $P(W|A)$  ovšem dlouho nebylo jak odhadnout, proto se využilo Bayesova pravidla:

$$P(W|A) = \frac{P(A|W)P(W)}{P(A)}. \quad (5.6)$$

Pravděpodobnost vstupních dat je ve vztahu k odhadovanému modelu konstantní a kladná, proto se jmenovatel zanedbává. Zbývá tedy odhadnout  $P(A|W)$  a  $P(W)$ . Pro to první se vžil název *akustický model (AM)*, pro to druhé *jazykový model (language model, LM)*.

Akustický model je právě tím místem, které zastávají skryté markovovské modely. Jedná se o generativní modelování, tedy pro účely řešení úlohy předpokládáme, že vstupní akustická sekvence je generována markovovským procesem, viz obrázek 5.5.

Markovovský model realizuje akustický model jako sumu odhadnutých pravděpodobností přechodů stavy automatu při pozorování dané vstupní posloupnosti:

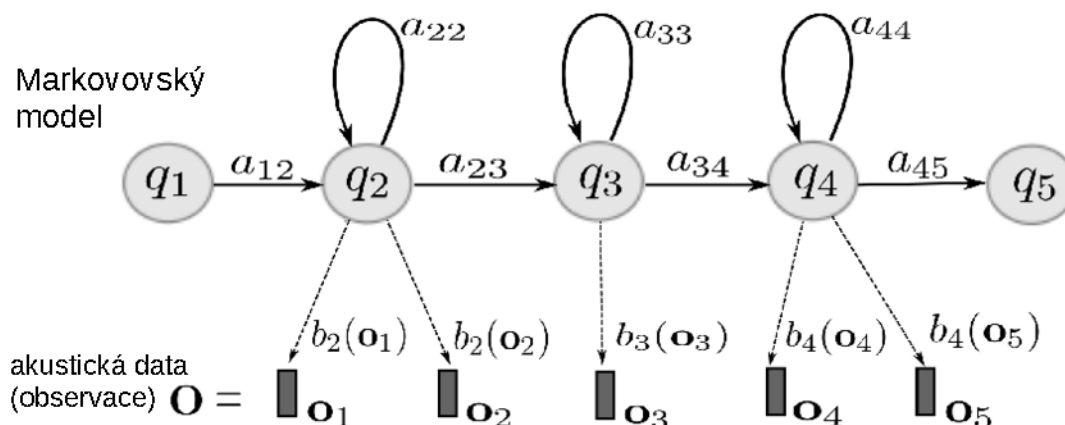
$$P(A|M) = \sum_X a_{x(0)x(1)} \prod_{t=1}^T b_{x(t)}(o_t) a_{x(t)x(t+1)} \quad (5.7)$$

kde  $x(0)$  je vstupní a  $x(T+1)$  výstupní stav modelu,  $a_{xy}$  je přechodová pravděpodobnost z  $x$  do  $y$ ,  $b_x$  je emisní pravděpodobnost symbolu  $x$  a  $X$  je množina možných stavových posloupností.

Parametry, které HMM definují, jsou vymezení stavů v prostoru vstupních dat a přechodové pravděpodobnosti mezi stavy. Musí se tedy určit množina modelů a prostor vstupních dat.

V prostoru vstupních dat, jak bylo řečeno, se musí vymežit stavy jednotlivých markovovských modelů. Intuitivně si toto lze představit jako akustický popis toho,





Obrázek 5.5: Schéma použití markovovského řetězce jako generativního modelu v rozpoznávání řeči.

jak vypadá spektrum, když se říká ta která hláska. Vymezení se provádí pomocí pravděpodobnostní distribuce určené středem a variancí v každé dimenzi.

Modelují se obvykle hlásky, ačkoliv v úvahu přicházejí i slabiky nebo v některých jazycích s kromobyčejně pravidelnou výslovností i přímo písmena. V obvyklém případě jednoho markovovského modelu pro každou hlásku se modely definují jako pětistavové: vstupní stav, stav modelující počátek hlásky, stav modelující prostředek hlásky, stav modelující konec hlásky a koncový stav. Krom toho existuje model pro ticho, které zahrnujeme mezi hlásky, a model pro krátkou pauzu, který je zvláštní tím, že může mít nulovou délku.

Přechodová matice se omezuje takto: Z počátečního stavu se přejde vždy do druhého. Z vnitřních stavů se přejde vždy buď do následujícího nebo se setrvá. Tak se nakonec vždy dá projít z počátečního stavu do koncového. Vzhledem k tomu, že ticho modeluje i neřečové události a často odpovídá dlouhým úsekům, povoluje se u něho, jakož i u krátké pauzy, přechod mezi druhým a čtvrtým stavem v obou směrech. Model pro krátkou pauzu sdílí stavy s modelem ticha, jen navíc umožňuje přechod z počátečního stavu rovnou do koncového.

Až na ticho a krátkou pauzu má tedy každá přechodová matice třikrát dvě pravděpodobnosti, které je potřeba natrénovat. Jestliže pak rozeznáváme u češtiny čtyřicet jednu hlásku plus ticho plus krátkou pauzu, bude tento jednoduchý akustický model mít

$$41 \times (6 + 2 \times 39) + 2 \times 39 + 8 + 9 = 3539 \quad (5.8)$$

volných parametrů.

Protože modelujeme hlásky a na výstupu očekáváme písmena, je potřeba adaptace na obou stranách akustického modelu. Na straně vstupu jde o konverzi trénovacích dat z dvojic *promluva - ortografický přepis* do dvojic *promluva - fonetický přepis*. Na straně výstupu jde o vytvoření výslovnostního slovníku, kde každému slovu je přiřazena množina potenciálních výslovností.

Jazykový model je nástroj pro odhadování pravděpodobnosti konkrétního textu. Praktický monopol zde mají tzv. n-gramové modely, kde se pravděpodobnost odhaduje přes počty výskytů konkrétních slov, dvojic slov, až n-tic slov v daném korpusu. Až v poslední době se objevují sofistikovanější přístupy s využitím rekurentních neuronových sítí.

Volné parametry akustického modelu se odhadují Baum-Welchovým iterativním algoritmem[52] taktéž na základě trénovacích dat. Oba základní stavební kameny: akustický i jazykový model tedy svoji efektivitu získávají z dat, takže značná část expertizy systému pochází z nich, nikoliv již z lidské ruky.

Důležitým zdokonalením systému je rozšíření množiny markovovských modelů. Jednotlivé realizace hlásek se od sebe tak odlišují, že modelovat např. všechny výskyty hlásky „h“ jedním modelem je nedostačující. Zavádí se proto kontext: Místo jednoho modelu pro danou hlásku máme model pro hlásku v kombinaci s předchozí a následující hláskou. Je-li hlásek 42 (včetně ticha), pak nyní bude modelů  $42^3 = 74088$ . 42 je příliš málo, 74088 je příliš mnoho. Navíc se od sebe hlásky liší jen v některých kontextech. Ještě navíc se jich většina vůbec nevyskytne v trénovacích datech. Proto se po rozštěpení na tyto tzv. „trifóny“ opět většina spojí do jednoho, tak že sdílí stavy, proto se jim říká *tied-state triphones*. Přejížděcí matice sdílí tak jako tak, těch se štěpení netýká, aspoň v obvyklé realizaci.

Posledním nezbytným standardním rozšířením je rozštěpení pravděpodobnostních distribucí ve stavech. Jeden gaušsian vždy nedostačuje, aby pokryl prostor v melfrekvenčních koeficientech, kde se daná hláska realizuje. Místo jednoho gaušsianu proto uděláme několik a dáme jim váhy, jejichž součet bude jedna. Štěpení může probíhat libovolně dlouho, dokud nedojde k přetrénování.

Základním postupem dekodování je Viterbiho algoritmus, ale v praxi se jen málokdy používá bez úpravy.

Náčrtnutá architektura je jakási startovní čára: nejjednodušší systém rozpoznávání řeči, který se dá zlepšit mnoha úpravami, nebo v posledku zcela nahradit moderním přístupem založeným zcela na hlubokých neuronových sítích. V následující sekci nastíním několik předchozích prací, které mi sloužily jako inspirace pro automatický přepis mluveného korpusu Karla Makoně.

## 5.4 Předchozí práce v rozpoznávání řeči

### 5.4.1 Ircing et al. 2001

V roce 2001 publikovala skupina slovných vědců v čele s Pavlem Ircingem práci o rozpoznávání češtiny pomocí HMM[53]. Akustický model trénují pomocí nástroje HTK na 22 hodinách materiálu. Parametrizují pomocí MFCC se dvěma úrovněmi derivace a s využitím keprstránní normalizace na úrovni vět. Modelují trifóny a ke shlukování používají fonologicky motivované skupiny podobně jako to je běžné pro angličtinu. K dekodování používají dekodér od AT&T na bázi konečného převodníku[54].

Jazykový model používají autoři bigramový při velikosti slovníku 60 tisíc forem. Článek představuje inovativní pokus o překonání problému neznámých slov, tedy těch, které jsou v testovacích datech, ale ne ve slovníku. Vytvářejí jazykový model založený nikoliv na celých slovech, nýbrž na kmenech a koncovkách. Tento přístup je intuitivně pro češtinu vhodný, ale u bigramového modelu se neosvědčil.

Udávaná úspěšnost systému s celoslovním jazykovým modelem je 70,47%.

### 5.4.2 Psutka et al. 2002 - 2005

V roce 2003 publikovali Psutka et al. článek o automatickém přepisu svědectví pamětníků holocaustu[55]. Článek navazoval na práci z předchozího roku[56], kde se konstatuje velká obtížnost přepisu materiálu, který obsahuje množství emocionálního projevu, nespisovného jazyka a akustických nedostatků. V článku z roku 2003 se klade důraz na experimenty s jazykovým modelem. Prezentovaným přínosem je využití dat z velkého korpusu češtiny, který celkově dobře nereprezentuje data, na kterých se provádí automatický přepis.

Experiment spočívá ve výběru vhodných vět z velkého korpusu do trénovacích dat pro jazykový model. K dispozici je malý korpus (A) již přepsaných výpovědí, tedy dokonale reprezentativní data. Dále velký korpus (B) z novin a literatury. Z obou korpusů se natrénovaly jazykové modely MA a MB. Věta X z korpusu B byla přidána do trénovacích dat, jestliže  $P(X|B) < P(X|A)$ . Jinými slovy, použily se věty z velkého korpusu, které se více podobaly datům z malého než datům z velkého. Tímto postupem se dosáhlo relativního umenšení chybovosti přes 4%.

Další navazující práce v rámci projektu Malach[57] z roku 2005 se již nezabývá jen češtinou, ale i slovenštinou a ruštinou. Článek přináší rozbor metod pro zacházení s různými výslovnostmi téhož slova. Autoři prezentují, že oproti udržování variant jako zvláštních položek ve slovníku je lepší mít každé takové slovo ve slovníku jenom jednou, ale s různými výslovnostními variantami, takže se zbytečně nerozdobuje jazykový model, ale neobětuje se akustická přesnost.

Všechny systémy prezentované v těchto článcích jsou založeny na systému HTK a používají parametrizaci na základě perceptuální lineární predikce (PLP).

### 5.4.3 Renals et al. 1994

Na rozdíl od článků popisujících tvorbu systémů rozpoznávání řeči pro praktické nasazení diskutovaných výše, tento článek od pětice autorů z roku 1994[58] představuje nové postupy, které se musejí nejdříve etablovat a musejí nalézt dostatečnou podporu ze strany výpočetních kapacit a množství dostupných dat.

Článek představuje několik významných posunů v přístupu k modelování mluvené řeči. Především jde o užití vícevrstevného perceptronu (*multi-layer perceptron, MLP*) pro přímé odhadování aposteriorní pravděpodobnosti přepisovaného symbolu na základě vstupních dat. Prezentuje také využití rekurentní neuronové sítě, kterýžto přístup doznává nyní širokého užití. Dále pojednává o prediktivních MLP a diskriminativních HMM.

MLP je posloupnost vstupní vrstvy, několika skrytých vrstev a vrstvy výstupní. Sousední vrstvy jsou propojeny váhovou maticí:

$$y_i^L = f\left(\sum_j w_{ij}^{L,L-1} y_j^{L-1}\right), \quad (5.9)$$

kde  $y_i^L$  je výstup  $i$ -té jednotky ve vrstvě  $L$ ,  $w_{ij}^{L,L-1}$  je prvek váhové matice mezi vrstvami  $L-1$  a  $L$  a  $f$  je aktivační funkce, v případě tohoto článku sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (5.10)$$

Důležitým poselstvím je zde vystoupení ze zažitých teoretických předpokladů v rozpoznávání řeči pomocí HMM: Autoři ukazují nevhodnost modelování hlásek generativním gaušovským procesem při použití rozkladu Bayesovým pravidlem. Na sadě DARPA o velikosti přibližně 4000 vět představují systém rozpoznávání řeči s využitím MLP jako technologie odhadu aposteriorní akustické pravděpodobnosti s výsledkem 3,6% WER.

#### 5.4.4 Graves & Jaitly 2014

Po nahrazení části procesu v markovovské mašinerii neuronovými sítěmi následoval další, dnes s odstupem času logicky vyhlížející krok, a sice přechod k systému postavenému kompletně na neuronové síti. V tomto článku[59] jde o další přesun expertizy z člověka na strojové učení, umožněné lepší dostupností většího množství dat a výpočetní síly. Použité algoritmy se samozřejmě taktéž zdokonalovaly, ale v základu byly mnohé známy již léta. Velkou výhodou navrhovaného systému je, že je tvořen jedním kompaktním modelem, který se trénuje najednou, za minimalizace chybové funkce odpovídající reálnému cíli: přesné transkripci.

Jediný bod, kde autoři zasáhli do systému, který se jinak trénuje „z jednoho konce na druhý“ (*end-to-end*), je předzpracování signálu. Jde jim však předněji o demonstraci přístupu *end-to-end* než o vytvoření praktické aplikace. Proto aby se zásah udržel zcela minimálním, operují na spektrogramech, ne na melfrekvenčních koeficientech. Spektrogramy generují, protože odvození relevantních akustických vlastností je náročná operace a razantně by zvýšila nároky na trénovací data, přičemž představený experiment je trénován na méně než stohodinové sadě.

Prezentovaný systém eliminuje zejména použití markovovského procesu jako modelu temporálního průběhu řeči a nutnost explicitní fonetické vrstvy. Používá CTC (*connectionist temporal classification*)[60] jako chybovou funkci. Pro modelování používá rekurentní neuronovou síť s *long short-term memory (LSTM)*[61]. Vzhledem k významnosti této architekturní změny použité postupy stručně nastíním.

Rekurentní neuronová síť (RNN) pro vstupní posloupnost  $\mathbf{x} = (x_1, \dots, x_T)$  spočte skrytou posloupnost  $\mathbf{h} = (h_1, \dots, h_T)$ , jakož i výstupní posloupnost  $\mathbf{y} = (y_1, \dots, y_T)$  iterováním následujících rovnic pro  $t$  od 1 do  $T$ :

$$h_t = f(W_{ih}x_t + W_{hh}h_{t-1} + b_h), \quad (5.11)$$

$$y_t = W_{ho}h_t + b_o, \quad (5.12)$$

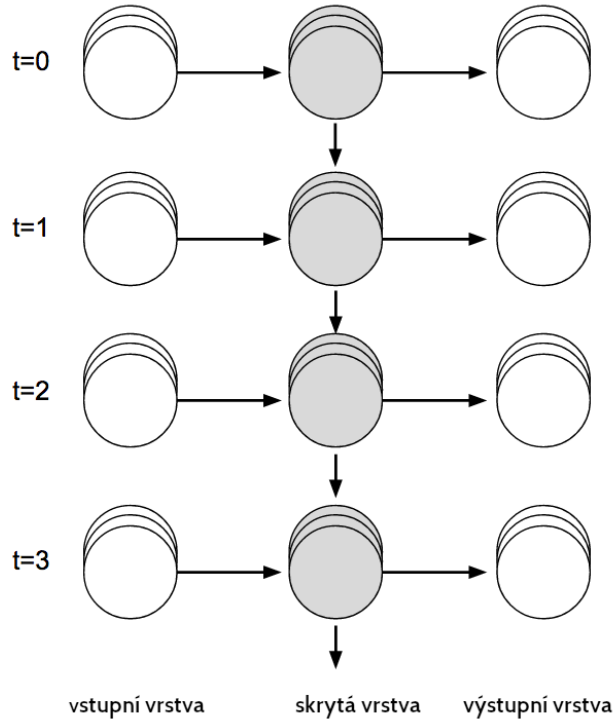
kde  $W$  jsou matice vah (např.  $W_{ih}$  je matice vah mezi vstupním a skrytým vektorem),  $b$  je *bias* (např.  $b_h$ , je bias skrytého vektoru) a  $f$  je aktivační funkce skryté vrstvy, viz obrázek 5.6. Jako aktivační funkci autoři používají

$$f_t = o_t \tanh(c_t) \quad (5.13)$$

kde

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \quad (5.14)$$

$$c_t = g_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (5.15)$$



Obrázek 5.6: Schéma rekurentní neuronové sítě. Vertikálně je znázorněn průběh času a horizontálně jednotlivé vrstvy sítě.

$$g_t = \sigma(W_{xg}x_t + W_{hg}h_{t-1} + W_{cg}c_{t-1} + b_g), \quad (5.16)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \quad (5.17)$$

$\sigma$  je logistický sigmoid,  $i$  je vstupní brána LSTM,  $g$  je brána zapomnění *forget gate*,  $o$  je výstupní brána,  $c$  jsou aktivační vektory buňky *cell activation vectors*.  $i$ ,  $g$ ,  $o$  i  $c$  mají shodnou délku se skrytým vektorem  $h$ . Na obrázku 5.7 je vyobrazena buňka LSTM.

Síť je rekurentní v obou směrech, neboť tomu nic nebrání, když vstupem jsou celé věty. Prezentovaný systém proto využívá obousměrné LSTM, viz obrázek 5.8. Ta definuje dopřednou skrytou vrstvu  $\vec{h}$  jako

$$\vec{h}_t = f(W_{x\vec{h}}x_t + W_{h\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}), \quad (5.18)$$

zpětnou skrytou vrstvu  $\overleftarrow{h}$  jako

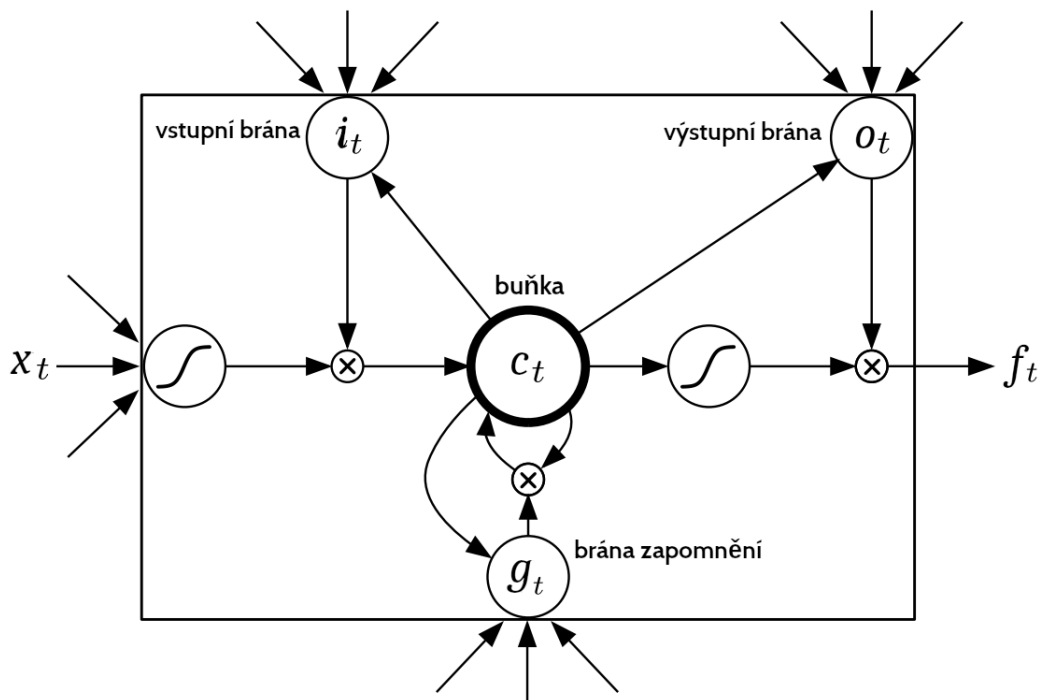
$$\overleftarrow{h}_t = f(W_{x\overleftarrow{h}}x_t + W_{h\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (5.19)$$

a výstupy jako

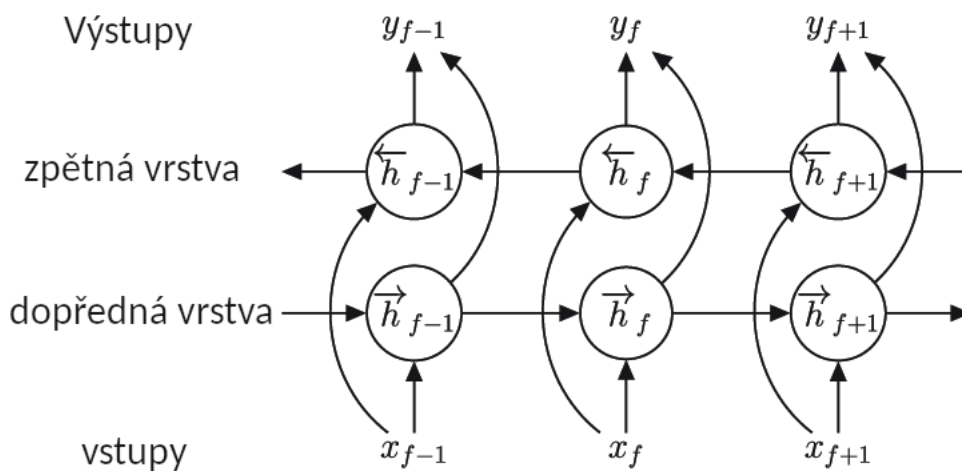
$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_o. \quad (5.20)$$

Použitá neuronová síť používá více skrytých vrstev, aby se umožnilo hluboké trénování, takže pro  $N$  vrstev ( $n = 1..N$ ) je vzorec třeba rozšířit:

$$h_t^n = f(W_{h^{n-1}h^n}h_t^{n-1} + W_{h^n h^n}h_{t-1}^n + b_h^n), \quad (5.21)$$



Obrázek 5.7: Schéma buňky LSTM.



Obrázek 5.8: Schéma obousměrné rekurentní neuronové sítě

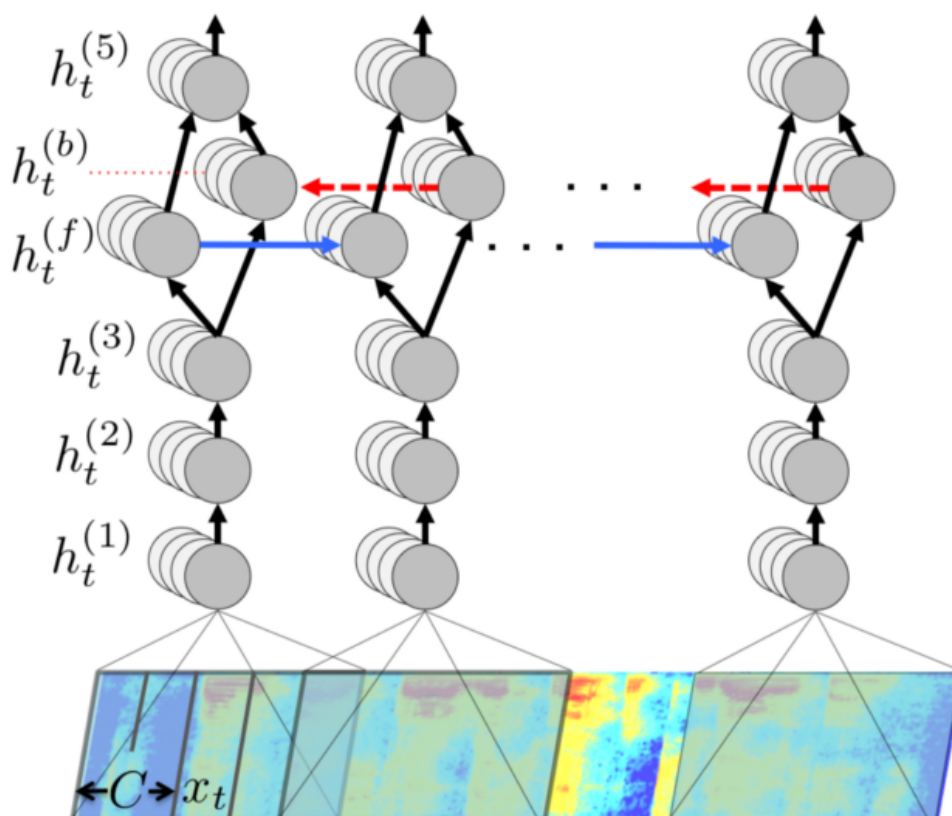
kde  $h^0 = x$ .

$$y_t = W_{h^N y} h_t^N + b_o. \quad (5.22)$$

Chybová funkce CTC je upravena tak, aby upřednostňovala kandidáty s nízkou WER a implementována pomocí metody Monte Carlo. Dekóduje se standardně pomocí paprskového prohledávání (*beam search*). Systém natrénovaný na jedenaosmdesátihodinové sadě z Wall Street Journalu sice s 8,2% WER nepřekonal baseline implementovaný pomocí DNN-HMM se 7,8%, ale to při tak malé trénovací sadě není žádné překvapení, a i tak ukázal životaschopnost metody.

### 5.4.5 Deep Speech

Na výše popsany článek navázali ve výzkumném centru Baidu a navrhli systém pojmenovaný DeepSpeech[62], který využívá tentýž přístup s důrazem na praktickou použitelnost a optimalizaci pro trénink na velkých množstvích dat. Na rozdíl od Gravesa (2014) používají jen jednu obousměrnou rekurentní vrstvu a další čtyři skryté dopředné. Jako aktivační funkci ve skrytých vrstvách používají místo sigmoidu shora omezené ReLu a na výstupu standardní softmax. U dopředných vrstev se aplikuje dropout. Rekurentní vrstva není LSTM z důvodu optimalizace rychlosti. Obrázek 5.9 znázorňuje architekturu sítě.



Obrázek 5.9: Architektura systému DeepSpeech.

### 5.4.6 Shrnutí

Během vývoje rozpoznávání řeči došlo k mnoha významným objevům a průlomům. Od porovnávání vzorků spektrogramů se v sedmdesátých letech přešlo ke generativním markovovským modelům. Ty se používaly až do příchodu hlubokých neuronových sítí, které nyní v ASR dominují stejně jako v jiných oblastech strojového učení.

## 5.5 Přepis Makoňova korpusu pomocí GMM-HMM

Zjednodušený řetězec vedoucí od zvukových dat k jejich přepisu v našem případě vypadá takto:

1. sběr trénovacích dat,
2. stavba akustického modelu,
3. stavba jazykového modelu,
4. automatické rozpoznávání.

V průběhu práce jsem sestavil dva zcela odlišné akustické modely: Jeden založený na skrytých markovovských modelech a jeden založený na neuronových sítích. Hlavním důvodem bylo to, že když jsem začínal, nebyly ještě hluboké neuronové sítě tak rozšířené. Jsou ale i dvě další výhody, které použití markovovských modelů opodstatňují: 1) Nepotřebují tolik trénovacích dat, takže se více hodí do začátku, kde je přepsaná množina malá,<sup>3</sup> a 2) umožňují získat přesné zarovnání slov a hlásek na časové pozice ve zvukovém záznamu. Dosud neznám žádný nástroj, který by toto poskytoval bez použití HMM.

Nejdříve popíšu výstavbu markovovského modelu a poté v sekci 5.12 se budu věnovat modelu založenému na neuronových sítích.

Stěžejními nástroji pro tvorbu systému jsou 1) HTK pro akustický model, 2) KenLM pro jazykový model a 3) Julius pro dekodování.

### 5.5.1 Modelované hlásky

Modeluji základní hlásky českého jazyka[63] reprezentované pomocí systému symbolů PACal[64]. Kromě základních hlásek používám dvojhlásky, ticho a krátkou pauzu. Ráz (glotální plozivu) nevyznačuji, jakož ani neřečové události. Důvodem toho je, že vyznačování těchto jevů nelze očekávat od anotátorů, se kterými tato práce počítá. V tabulce 5.1 jsou použité hlásky uvedeny.

V závislosti na množství trénovacích dat jsem nahrazoval některé hlásky častějšími podobnými. V tabulce 5.2 jsou záměny vyčísleny.

---

<sup>3</sup>První trénovací sadu jsem pořídil svépomocí přepisem asi 15 minut nahrávky 85-05A.



IPA	PACal	grafém	IPA	PACal	grafém
a	a	a	ɱ	mg	tram <u>v</u> aj
a:	aa	á	n	n	<u>n</u> e
aʊ	aw	au	ŋ	ng	tan <u>k</u>
b	b	b	ɲ	nj	ň
(ts)	c	c	o	o	o
(tʃ)	ch	č	o:	oo	ó
d	d	d	oʊ	ow	ou
ɟ	dj	ď	p	p	p
(dz)	dz	dz	r	r	r
(dʒ)	dzh	dž	r̥	rsh	tř <u>i</u>
ɛ	e	e	r̥	rzh	ř <u>í</u> z
ɛ:	ee	é	s	s	s
eʊ	ew	eu	ʃ	sh	š
f	f	f	t	t	t
g	g	g	c	tj	ť
h	h	h	ʊ	u	u
i	i	i	u:	uu	ú, ů
i:	ii	í	v	v	v
j	j	j	x	x	ch
k	k	k	z	z	z
l	l	l	ʒ	zh	ž
m	m	<u>m</u> ák		sil	
				sp	

Tabulka 5.1: Použité hlásky: IPA, PACal a nejčastější odpovídající grafém.

	před záměnou		po záměně	
	IPA	PACal	IPA	PACal
	ɱ	mg	m	m
	aʊ	aw	a ʊ	a u
	o:	oo	o	o
*	(dz)	dz	(ts)	c
	(dʒ)	dzh	(tʃ)	ch
*	eʊ	ew	ɛ ʊ	e u

Tabulka 5.2: Použité záměny hlásek; hvězdičkou jsou vyznačeny záměny použité ještě v době psaní textu.

## 5.5.2 Tvorba akustického modelu

### 1. Vytvoření počátečních modelů.

Všechny hlásky se inicializují jako shodné. Každá hláska je reprezentována pěti stavy (vstupním, výstupním a třemi vnitřními). Přejídnové pravděpodobnosti se nastaví tak, aby byly možné jen kýžené přechody, to jest ze vstupního do druhého a z každého z vnitřních stavů do sebe samého nebo do následujícího. Pravděpodobnosti se inicializují na 60% pro setrvání a 40% pro postup ve druhém a třetím stavu a 70% pro setrvání a 30% pro postup ze čtvrtého do výstupního. Střed a variance jsou určeny identicky podle globálních hodnot.

V této počáteční sadě je obsaženo ticho (`sil`), ale nikoliv krátká pauza (`sp`). To se týká nejen parametru udávající množinu hlásek, ale také trénovacího fonetického přepisu. Pro kódování používám formát MFCC s první a druhou derivací, základní frekvencí a keprální normalizací (`MFCC_0_D_A_Z` v notaci HTK).

Následují dvě iterace tréninku Baum-Welchovým algoritmem.

### 2. Přidání modelu pro krátkou pauzu.

Z modelu pro ticho se odvodí model pro krátkou pauzu tak, že se povolí přechody z druhého do čtvrtého stavu a zpět s pravděpodobností 0,2 a ze vstupního do koncového stavu s pravděpodobností 0,3, aby byl model robustnější a mohl modelovat pauzu mezi slovy, která je nezřídka nulová.

Stavy mezi modelem pro ticho a pro krátkou pauzu se sdílejí. Trénuje se opět dvěma iteracemi BW-algoritmu, od teď již s modelem pro krátkou pauzu jak v množině hlásek, tak ve fonetickém přepisu.

### 3. Nucené zarovnání a odvržení zmetkových vzorků.

Pomocí Viterbiho algoritmu[65] se provede tzv. *forced alignment*, tzn. nucené zarovnání na úrovni hlásek. Jinými slovy určí se přesný čas, kde začíná a končí která hláska. Při tom se určí hranice, pod kterou když klesne *likelihood* daného přepisu na základě odpovídající nahrávky, tato se z trénovacích dat odstraní jako pravěpodobně vadná. Pro zarovnání se použije Viterbiho algoritmus v provedení programu HVite s prahem pro odmítnutí věty 150. Následují další dvě iterace BW-algoritmu.

### 4. Přepočítání variance.

Variance modelů byla určena podle původní trénovací sady. Nyní jsme z ní vyřadili některé vzorky, proto proběhne její přepočtení, opět následované dvěma trénovacími iteracemi.

### 5. Přejídn k trifónům

Z nuceného zarovnání máme přepis obohacený o konkrétní fonetické realizace. Z té se nyní snadno vytvoří přepis trifónový tak, že ke každé hlásce přidáme jeho levý a pravý kontext, pokud nejsou na začátku nebo na konci věty.

Je-li hlásek 45, pak trifónů je až  $45^3 = 91125$ . Ne všechny se v trénovacích datech objeví. V praxi jich mám kolem 14 tisíc. Pokud by každý trifón měl vlastní separátní model, došlo by k opačnému problému než v případě monofónů, totiž že by celkový model měl příliš mnoho parametrů. Přechodové matice mohou všechny trifóny odvozené od jednoho monofónu sdílet. Avšak které trifóny mají sdílet střed a varianci stavů a které mají mít vlastní, je třeba rozhodnout opatrněji.

Pro určení, které modely je vhodné sloučit, používám rozhodovací stromy. Na základě předem definovaných kritérií se u každého emitujícího stavu každé skupiny trifónů provede rozdělení na dva shluky, což umožní zvýšení *log likelihood* dat. Vybere se kritérium, které log likelihood zvýší nejvíce a postup se opakuje, dokud zvýšení neklesne pod danou hranici. Takto získané shluky se pak sloučí do jednoho logického trifónu.

Pro tvorbu rozhodovacích stromů je potřeba ručně vytvořit otázky, na jejichž základě bude algoritmus dělit hlásky do shluků. K tvorbě otázek můžeme použít lingvisticky motivovanou kategorizaci v naději, že aspoň některé lingvistikou definované kategorie budou z pohledu trénovacích dat tvořit konzistentní shluky. Pro tvorbu otázek jsem vycházel z předlohy pro angličtinu, jak je uvedeno v HTK Book, a z kategorizace českých hlásek na Wikipedii ([https://cs.wikipedia.org/wiki/Fonologie\\_čestiny](https://cs.wikipedia.org/wiki/Fonologie_čestiny)). Otázky použité v rozhodovacím stromě viz v digitální příloze.

## 6. Přejchod ke gaušovským směsem

Posledním krokem ve zvětšování komplexity modelu je přechod z modelování stavů prostými gaušovskými pravděpodobnostními distribucemi k jejich směsem (angl. *mixtures*). Spočívá v tom, že se přesněji modelují variantní realizace jednotlivých hlásek. Daná hláska v jednom stavu HMM pak není modelována jednou gaušovskou distribucí, nýbrž složením několika. Každá má svůj střed, svoji varianci a svoji váhu, jejichž celkový součet musí být roven jedné.

Štěpí se vnitřní stavové modely jednotlivých hlásek. Optimální počet složek směsi je tedy potřeba zjistit pro trojnásobek počtu použitých trifónů. To jsou řádově tisíce až desítky tisíc. V okamžiku psaní tohoto textu používám 8444 reálných trifónů; 13746, počítám-li i ty logické. To znamená přes dvacet pět tisíc distribucí, u nichž je potřeba určit optimální počet složek.

Aby byl úkol aspoň aproximací dosažitelný, je třeba hledat efektivněji než prohledáváním celého prostoru hrubou silou. První pomocí zde je, že modely jsou na sobě více méně nezávislé: Nalezneme-li optimální počet složek pro jeden z nich, nemělo by to ovlivnit optimální počet složek u jiného.

Rozštěpení v jedné směsi proběhne tak, že se složka s největší vahou rozštěpí na dvě totožné s tím, že jedna dostane malinko větší váhu než druhá, aby se při trénování mohly rozejít. To se provede u všech vnitřích stavů všech fyzických hlásek, t.j. u všech markovovských modelů. Provedou se čtyři trénovací iterace a úspěšnost se vyhodnotí na sadě heldout.

Pokud u některé složky klesne její váha pod daný práh, vymaže se, čímž se zamezí zbytečnému nárůstu parametrů, a není proto potřeba zkoušet štěpit

jednotlivé modely samostatně. Arci, štěpením modelů jednoho po druhém jsem nikdy nedosáhl lepšího výsledku, než štěpením všech modelů najednou.

Závislost úspěšnosti na počtu složek není monotónní, proto ve štěpení pokračuji, i když někdy úspěšnost klesne. Konkrétně zastavím štěpení, pokud úspěšnost klesne o více než 30% oproti nejvyšší dosažené nebo pokud klesne více než třikrát za sebou, ne však když je složek méně než 16.

### 5.5.3 Dekódování

Pro dekodování, čili samotný přepis na základě akustického a jazykového modelu, používám nástroj Julius[66]. Julius pracuje na základě dvouprůchodového algoritmu. Při prvním průchodu se využívá paprskového prohledávání (*beam search*) na slovníku s vahami pro každé slovo zvlášť, takže průchod je velmi rychlý a málo paměťově náročný. Datové struktury jsou ponechány do druhého průchodu, který jde v opačném směru a zapojuje n-gramový jazykový model. V tomto průchodu se hledá  $k$  nejlepších kandidátů v *trellis* (dosl. pergola; datová struktura) z prvního průchodu.

## 5.6 Jazykový model

Jazykový model obecně je odhad pravděpodobnostního rozdělení posloupností slov v přirozeném jazyce[67]. V kontextu rozpoznávání řeči je tandemovým partnerem akustického modelu[68]. Teprve kombinace akustického a jazykového modelu určí výsledné slovo, které se na dané pozici rozpozná jako nejpravděpodobnější.

Výběr jazykového modelu je omezen nástrojem pro rozpoznávání. Lze zvolit pouze takový model, který nástroj dokáže využít. Všechny nástroje, které jsem použil, podporují N-gramové jazykové modely: HVite bigramový, Julius až trigramový a DeepSpeech libovolného řádu.

Pro trénování jazykového modelu mám k dispozici čtyři druhy dat:

1. Manuální přepisy nahrávek,
2. automatické přepisy,
3. Makoňovy spisy,
4. obecné české texty<sup>4</sup>.

Každá z těchto kategorií skýtá různé množství textu a různou věrnost modelovanému materiálu. Nejvěrnější jsou samozřejmě manuální přepisy Makoňových nahrávek, kterých je nejméně. V okamžiku psaní tohoto textu je to 728 286 slov. Automatické přepisy, jejichž přínos pro jazykové modelování je nejasný, představují 7 338 504 slov. Makoňovy spisy obsahují 3 328 720 slov. Obecné české texty jsou nejdostupnější z těchto komodit. Nejobsáhlejší dostupný korpus, který jsem našel, je Mononews z WMT[69] obsahující 1 019 497 060 slov. Velikosti zdrojů jsou tedy v poměru 1:10:5:1400.

<sup>4</sup>Obecnými českými texty nemyslím texty v *obecné češtině*, nýbrž obecné ve smyslu všech, které šlo opatřit.

Inspirován výše zmiňovaným článkem od Psutky et al. 2003[55], kde autoři tvoří trénovací korpus z reprezentativního subkorpusu a selekce vět z velkého nereprezentativního subkorpusu, pokusil jsem se jejich přístup replikovat. Na základě předběžných pokusů jsem zjistil, že nejlepší výsledek dává jazykový model natrénovaný z Makoňových spisů a manuálních přepisů, naopak přidání automatických přepisů nemá prakticky žádný efekt. Výchozím bodem pokusu tedy byl reprezentativní korpus z Makoňových textů a korpus obecných českých textů. Cílem bylo vybrat z obecného korpusu ideální podmnožinu vět vzhledem k úspěšnosti rozpoznávání. Pro tyto účely jsem natrénoval dva unigramové jazykové modely, jeden z reprezentativního (Makoňova) korpusu:  $M$  a druhý z obecného (WMT):  $W$ .

První hledání proběhlo podle návodu ve zmiňovaném článku: Do trénovací sady se vybraly ty věty  $s$  z obecného korpusu, které byly pravděpodobnější podle modelu  $M$  po aplikaci hledaného koeficientu  $t$  než podle modelu  $W$ :

$$P(s|W) < t \cdot P(s|M) \quad (5.23)$$

S klesajícím  $t$  a tím s rostoucím počtem přidávaných vět monotónně klesala úspěšnost. Pohled na vybrané věty odhalil problém: byly to namnoze takové, které byly velmi nepravděpodobné podle obecného modelu, takže špatně reprezentovaly češtinu.

Druhé hledání proběhlo s úpravou, že se použijí toliko věty, které nejsou podle reprezentativního modelu příliš nepravděpodobné. Průměrná *log likelihood* věty vážená počtem slov je v reprezentativním modelu -3,51 a standardní odchylka je 0,52. Práh pro inkluzi věty jsem tedy nastavil na -4. Takto jsem došel ke zvýšení úspěšnosti o 2,7%, z 0,112 na 0,109 WER. Kritérium pro zahrnutí věty z obecného korpusu bylo tedy toto:

$$10^{-4} < P(s|W) < t \cdot P(s|m) \quad (5.24)$$

U třetího hledání jsem zcela ignoroval pravděpodobnost věty podle obecného modelu a rozřazoval pouze na základě pravděpodobnosti podle reprezentativního modelu. Touto metodou jsem dosáhl zvýšení úspěšnosti o 8,0% z 0,112 na 0,103 WER. Kritérium:

$$P(s|M) < t \quad (5.25)$$

Vývoj úspěšnosti podle kritéria přidání vět z obecného korpusu do jazykového modelu shrnuje tabulka 5.3 a obrázek 5.10. Výsledek považuji za zajímavý, neboť shledává jinou, dokonce jednodušší metodu v tomto případě účinnější než navrhovanou autory.

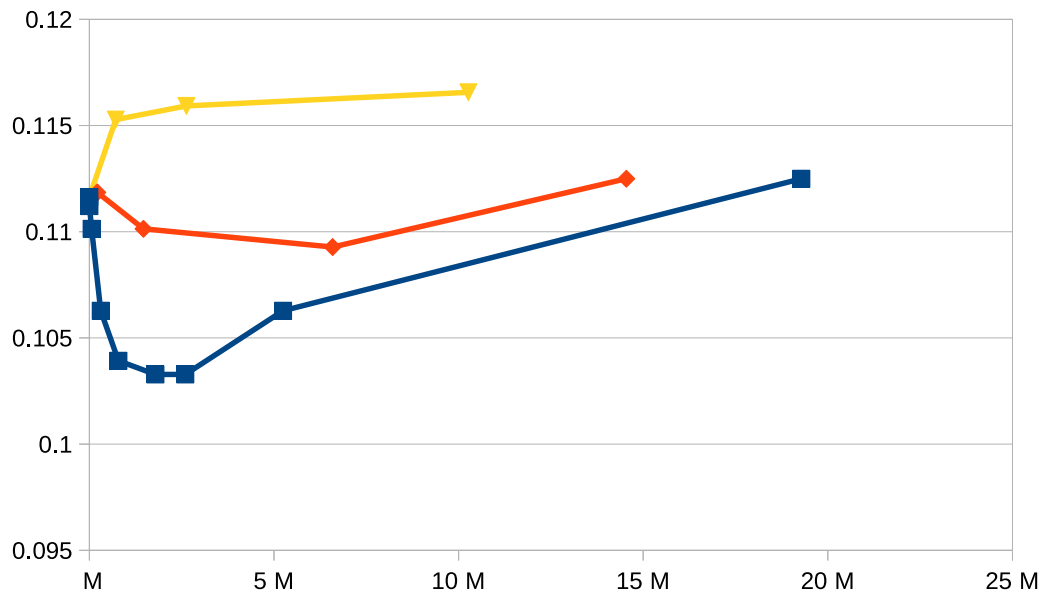
Jazykový model stavím pomocí nástroje KenLM[70]. Aplikuji vyhlazování technikou Knesser-Ney[71], jak je v něm zabudována.

Je třeba podotknout, že u manuálních přepisů dochází k jednomu nežádoucímu jevu. Přepisy se pořizují tak, aby byly maximálně věrné tomu, co je vyřčeno. Proto se doslova... vlastně do hlásky přepisují i přebrepty a zadrhnutí. Je otázkou, zda takové jevy chceme mít v jazykovém modelu. Aníž bych se na ni pokoušel poskytnout definitivní odpověď, můj přístup, jak se s tímto vypořádat, je, instruovat uživatele těmito pokyny: 1) Každé slovo, po kterém následuje přerušení toku mluvy doplnit třemi tečkami, 2) pokud se mluvčí zadrhne uprostřed slova nebo vysloví něco, co není slovem, připojit k tomuto útvaru pomlčku<sup>5</sup>. Například ve

<sup>5</sup>Standardní postup, se kterým jsem se seznámil až později, je místo pomlčky použít symbol „+“.

kritérium	počet přidanych vět			WER
	#	%W	$\div M$	
$\emptyset$	0	0,000%	0,00	11,2%
$w < 0,65m$	715246	0,991%	2,09	11,5%
$w < 0,8m$	2629381	3,644%	7,70	11,6%
$w < m$	10263276	14,223%	30,05	11,7%
$w < 0,65m \ \& \ m > 10^{-4}$	217359	0,301%	0,64	11,2%
$w < 0,8m \ \& \ m > 10^{-4}$	1466100	2,032%	4,29	11,0%
$w < m \ \& \ m > 10^{-4}$	6587540	9,129%	19,29	10,9%
$w < 1,3m \ \& \ m > 10^{-4}$	14544308	20,156%	42,58	11,2%
$m > 10^{-2}$	2380	0,003%	0,01	11,1%
$m > 10^{-2,5}$	61654	0,085%	0,18	11,0%
$m > 10^{-2,8}$	308102	0,427%	0,90	10,6%
$m > 10^{-3}$	786947	1,091%	2,30	10,4%
$m > 10^{-3,2}$	1790919	2,482%	5,24	<b>10,3%</b>
$m > 10^{-3,3}$	2598266	3,601%	7,61	10,3%
$m > 10^{-3,51}$	5246049	7,270%	15,36	10,6%
$m > 10^{-4,03}$	19282850	26,723%	56,45	11,2%

Tabulka 5.3: Úspěšnost rozpoznávání s použitím různých částí obecného korpusu. Kritérium rozhoduje o zařazení věty do jazykového modelu. Proměnná  $m$  je pravděpodobnost věty podle unigramového modelu z Makoňových přepisů a spisů, vážená počtem slov. Proměnná  $w$  je totéž podle modelu z obecných českých textů. Počet přidanych vět je uveden v celkovém počtu, v procentech celkové velikosti obecného korpusu a v násobcích velikosti reprezentativního korpusu.



Obrázek 5.10: Hledání optimální podmnožiny obecného korpusu pro jazykové modelování. Na ose X je počet přidanych vět, na ose Y WER. Žlutá čára s trojúhelníčky reprezentuje první pokus (řádky 2-4 v tabulce 5.3); Červená čára s čtverečky na hrotech druhý pokus (řádky 5-8); modrá čára s čtverečky na stranách třetí pokus (řádky 9-16).

věťe „*To je ta relativnost dobrého a zlého, tak kdybychom... to je tam jinak postaveno.*“ patří tři tečky za slovo *kdybychom*. Ve větě „*To je první š- špatný pohled, chybný pohled na rodiče a na předcházející generaci.*“ patří pomlčka za vyslovené š, za kterým teprve následuje vyslovené slovo *špatný*.

Při stavbě jazykového modelu toto značení umožní, aby se věta při setkání s takovým slovem ukončila nebo aby se takové slovo přeskočilo.

## 5.7 Rozdělení dat

Pro natrénování modelu strojovým učením je potřeba trénovacích dat a pro vyhodnocení jeho úspěšnosti dat testovacích, která ve fázi trénování nesmí být algoritmem spatřena. Při trénování samotném se pak mnohdy používá vyhrazených, tzv. *heldout* dat<sup>6</sup> pro průběžné měření úspěšnosti. V případě trénování akustického modelu s použitím HTK je tomu nejinak. Heldout data jsou používána pro zjištění optimálního počtu složek v gaušovských směsích modelů jednotlivých hlásek a testovací data jsou používána pro závěrečné vyhodnocení.

Anotovaná data mi přibývala velice pozvolna. Začínal jsem s několika minutami, ovšem přírůstky byly časté. Nemohl jsem si tedy dovolit udělat od začátku pevnou testovací sadu, kterou bych používal po celou dobu provádění experimentů. Místo toho jsem s každou novou dávkou anotovaných dat celou sadu rozdělil podle vět v poměru 18:1:1 do trénovací, heldout a testovací sady. Tak jsem měl neustále vyvážený poměr jednotlivých datových sad. Zřejmou velkou nevýhodou bylo, že nešlo spolehlivě porovnávat výsledky jednotlivých experimentů vzhledem k variabilní testovací sadě.

Až když jsem měl několik desítek hodin anotovaných dat, vyhradil jsem si fixní testovací sadu. Běžně se testovací sada vybere jako náhodná podmnožina vzorků z trénovací sady tak, aby měla kýženou velikost. V mém případě vzorků zvíci hodinových nahrávek jsem sadu určil manuálně jako úsek druhé až jedenácté minuty (tedy deset minut, vždy jednu minutu po začátku) v pěti nahrávkách,

1. jedné kazety z roku 1976,
2. jedné z roku 1982,
3. jedné z roku 1986,
4. jedné z roku 1990 a
5. jednoho nedatovaného kotouče.

Sadu heldout nyní vybírám jako každou čtyřicátou větu. Z každé dvacáté jsem snížil na polovic nejen abych neplýtvat trénovacími daty, nýbrž také protože vyhodnocování směsí zabírá při trénování zdaleka nejvíce času, a ten je přímo úměrný velikosti sady heldout.

---

<sup>6</sup>Často zaměňovaných s vývojovou testovací sadou označovanou běžně jako *dev* / *dtest*.

## 5.8 Experiment s kepstrální normalizací

Kepstrální normalizace je standardní technikou pro kompenzaci různorodých akustických podmínek v rámci trénovacích a testovacích dat, viz Viikki a Laurila (1998)[72]. Princip této techniky spočívá v tom, že se ode všech melfrekvenčních kepstrálních koeficientů odečte jejich průměr z akusticky konzistentního úseku. Já toto poněkud hrubiánsky činím na celých nahrávkách, které nejsou vždy akusticky konzistentní. Často však ano a nalezení akusticky podobných množin je jedním z mých plánů pro budoucí práci.<sup>7</sup>

Nabízí se však otázka, zda má smysl odečítat průměr z celé nahrávky, nebo by to mělo být jen z řečových událostí, tedy s vynechaným tichem (šumem, hluky atd.) Tato idea, přišedší ke mně od dr. Davida Klusáčka, mne zaujala natolik, že jsem se ji pokusil ověřit. Vytvořil jsem proto metadata s časovými pozicemi všech izolovaných řečových událostí na základě zarovnaného automatického přepisu a sadu skriptů pro manipulaci se soubory MFCC.

Ke každému souboru MFCC jsem vytvořil kopii, ze které jsem odstranil všechny výskyty hlásek `sil` a `sp`. Průměry kepstrálních koeficientů jsem pak spočetl na těchto kopiích a odečetl je od hodnot v originálech. Trénování i testování jsem pak prováděl na takto upravených souborech místo standardní normalizace, jak ji poskytuje systém HTK.

S použitím takto normalizovaných nahrávek se chybovost na slovech snížila ze 46,4% na 45,9%. Aparát pro dekódování a manipulaci souborů MFCC považuji za vítaný vedlejší produkt.

## 5.9 Aktivní učení

Aktivní učení spočívá ve vhodném výběru trénovacích dat, viz např. Cohn (1996)[73]. V mém případě trénovací sada postupně roste a nabízí se tedy využít techniky aktivního učení tak, aby se získávala co nejvhodnější trénovací data.

Podnikl jsem experiment, ve kterém jsem ve webovém rozhraní ke sběru trénovacích dat (viz kapitolu 6) slova s nízkou *confidence measure* (*c.m.*) (pod 0,3) podtrhl červenou přerušovanou čarou, jejíž sytost spojitě rostla s klesající *c.m.* Instruoval jsem pak uživatele aplikace, aby přednostně přepisovali věty, které jsou opticky co nejčervenější. Bohužel přirozené puzení uživatelů přepisovat nahrávku kompletně a lineárně od začátku způsobilo, že přepisu, kde se tato instrukce do-držuje, je zcela mizivé množství.

Druhý experiment spočíval v tom, že webová aplikace sama vybírala věty pro přepis na základě toho, kolik obsahovala slov s nízkou *c.m.* Uživatelé pak byli instruováni, aby nahrávku jen poslouchali, a opravu vložili, až když se přehrávání samo přeruší. Tento pokus skončil neúspěchem, neboť změna v chování aplikace byla pro uživatele natolik nepříjemná, že jsem je raději navrátil do původního.

---

<sup>7</sup>Tato úloha je již vyřešena a popsána v sekci 3.2 a následující.



## 5.10 Rozšíření trénovací množiny

Skóre confidence measure se dá využít ještě jiným způsobem: lze vybrat úseky v korpusu kromě testovací sady<sup>8</sup>, kde automatický přepis uvádí vysokou míru c.m. a přidat tyto úseky do trénovací množiny. Tento experiment jsem provedl tak, že minimální délku úseku jsem stanovil na 1 sekundu, minimální počet obsažených hlásek na 10 a minimální c.m. na 0,6. Práh 0,6 jsem určil namátkovou kontrolou, která poukazovala na zanedbatelnou chybovost takových úseků. Celkem tento výběr poskytl 99 hodin audia, tedy 10% celého korpusu. Chybovost rozpoznávání se zvýšila ze 46,4% na 49,4% a doba trénování se zvýšila také.

## 5.11 ASR na parlamentním korpusu

Po rozšíření hlubokých neuronových sítí došlo k pokusu o přepis pomocí systému na nich založeného. Hluboké neuronové sítě našly využití snad ve všech oblastech strojového učení, viz např. LeCun (2015)[74], Hinton (2012)[75]. Toto neminulo ani rozpoznávání řeči, konceptuálně již dávno před tím, viz Morgan (1995)[76]. Nezisková organizace Mozilla vydala vlastní svobodný nástroj pro rozpoznávání řeči DeepSpeech[62] založený na TensorFlow[77], který implementuje práci popsanou v podsekcí 5.4.5. Na rozdíl od článku[62] však implementuje rekurentní vrstvu pomocí LSTM upřednostňuje přesnost modelu před počítačnou efektivitou.

Dříve než popíšu použití neuronových sítí pro přepis Makoňova korpusu, pojednám o použití datové sady představené v kapitole 4 pro rozpoznávání řeči. Zmíněnou datovou sadu jsem použil pro natrénování rozpoznávače řeči pomocí systému DeepSpeech. Sadu jsem rozdělil na trénovací (train), ladící (dev) a testovací (test)<sup>9</sup> v poměru 18:1:1. Hyperparametry jsem nastavil následovně: learning rate 0,0001, dropout rate 0,2, zbytek ponechán defaultně. Ke konvergenci došlo po 12 epochách. S použitím pentagramového jazykového modelu natrénovaného na stenografických prepisech byla výsledná WER 8,40% před expanzí číslic a 7,89% po expanzi.

Ze zdrojů zmíněných v úvodu kapitoly 4 se bez větší námahy se získkem v řádu aspoň desítek hodin daly použít 1) Otázky Václava Moravce a 2) Vystadial. Krom toho jsem použil veřejně nepřístupné zdroje 3) CUCFN – Korpus Finančních zpráv Univerzity Karlovy, 4) Korpus reprezentativní mluvené češtiny Oral2013[78] a 5) amatérsky namluvenou bibli, která je dostupná na adrese [poslouchamebibli.cz](http://poslouchamebibli.cz), a u které není žádná explicitní licence.

Na všech sadách včetně manuálních prepisů nahrávek Karla Makoně jsem natrénoval model s použitím téhož systému Mozilla DeepSpeech, týchž hyperparametrů a téhož obecného jazykového modelu. Ten jsem natrénoval na datech z WMT 2019[69].

Nakonec jsem natrénoval akustický model na všech trénovacích sadách sloučených do jedné. Tabulka 5.4 shrnuje word error rate systémů trénovaných na jednotlivých částech testovaných jednak na testovací sadě z téhož zdroje a jednak na agregované testovací sadě sloučené ze všech dílčích. Z důvodu použití obecného

<sup>8</sup>K zamyšlení: je zde opravdu nutné vynechat testovací data?

<sup>9</sup>Toto označení je v souladu s konvencí pro systém DeepSpeech, ale ve skutečnosti jde po řadě o množiny train, heldout a dtest, jak zmiňuji v sekci 5.7.

jazykového modelu je u parlamentního korpusu vyšší chybovost než výše zmiňovaných 8,40% a u Makoňova korpusu taktéž vyšší chybovost než 19% uvedených v sekci 5.12.

zdroj	WER na sobě	WER na všech
bible	9,20%	94,7%
cucfn	31,6%	72,8%
makoň	30,4%	77,3%
oral2013	78,4%	60,7%
ovm	21,6%	72,9%
parlament s číslicemi	8,74%	39,7%
parlament ve slovech	7,89%	36,0%
vystadial	51,0%	74,0%
vše s číslicemi	28,4%	28,4%
vše ve slovech	26,0%	26,0%

Tabulka 5.4: Word error rate rozpoznávání řeči na jednotlivých korpusech a na jejich konkatenaci.

## 5.12 Přepis Makoňova korpusu pomocí neuronových sítí

Pro přepis Makoňova korpusu jsem natrénoval dva akustické modely pomocí DeepSpeech: 1) čistě na manuálních prepisech Makoňových nahrávek (v té době 100 hodin), 2) na agregované sadě popsané v sekci 5.11. Používal jsem parametr *automatic mixed precision* pro rychlejší trénink, *batch size* 50, *dropout* 0,2 a *learning rate*  $1^{-4}$ .

Model natrénovaný jen na Makoňových nahrávkách zkonvergoval po osmi epochách a dosáhl word error rate 19,2% na testovací sadě. Druhý model jsem vytvořil tak, že jsem nejprve trénoval na agregované trénovací sadě s použitím Makoňovy sady heldout pro validaci. Tato část zkonvergovala po 15 epochách a dosáhla WER 16,6%. Následně jsem pokračoval v tréninku na Makoňových nahrávkách. Tato část zkonvergovala po dvou epochách a konečné skóre bylo 13,0% WER.

Dokud jsem ale trénoval s číslicemi, viz sekci 4.2, dosáhl tento model výrazně nižší úspěšnosti s téměř dvojnásobnou chybovostí 27,3% resp. 23,5% WER. Na testovací sadě měl tedy vyšší úspěšnost model trénovaný jen na 100h Makoňových nahrávek se standardní abecedou, předče model trénovaný na 1500 hodinách na abecedě rozšířené o číslice.

Nejmarkantnější zlepšení přináší robustní model na velice poškozených nahrávkách, které jsem v době výběru testovací sady ještě neměl k dispozici, any byly digitalizovány mnohem později. Jedná se hlavně o nahrávky pořizované při nízké rychlosti převíjení pásky, zmiňované jednak v sekci 3.4 a jednak v sekcích 2.3 a 2.4.

Na minutovém úseku jedné z nejpoškozenějších nahrávek, který jsem přepsal, má první model WER 94,1%, zatímco robustní model má WER 75,8%. Porovnání na větším, asi pětiminutovém úseku je uvedeno v sekci 3.4.3.

Bohužel vinou přímého mapování z parametrizovaného audia na grafémy přicházíme o možnost zarovnání na úrovni hlásek, takže je nutno výstup nechat zarovnat v další iteraci, aby se umožnilo synchronní přehrávání ve webovém rozhraní.

## 5.13 OOV

*Out of vocabulary*<sup>10</sup>, tedy *mimo slovník*. Tak se zove jev, kdy výstupem rozpoznávání řeči je slovo, které jazykový model nezná. Může k tomu dojít ve dvou případech: Buď když je neznámé slovo součástí jazykového modelu nebo když je rozpoznávání řeči schopno vydávat i slova mimo slovník jazykového modelu.

Vzhledem k architektuře systému DeepSpeech je můj případ ten druhý. OOV tedy v jazykovém modelu nemám a slova mimo slovník se na výstupu objevily tehdy, když podle akustického modelu predikovaná posloupnost hlásek předčila všechny kandidáty, kterým jazykový model odhadoval nenulové pravděpodobnosti. DeepSpeech kombinuje odhad pravděpodobnosti podle akustického a jazykového modelu aditivně, takže nulová pravděpodobnost od jazykového modelu přijetí hypotézy neznemožňuje.

V praxi bylo v mém případě takových slov zanedbatelné množství: 49 z 8 milionů slov při pentagramovém modelu se slovníkem o velikosti 206 tisíc slov. Při trigramovém modelu se slovníkem o velikosti 140 tisíc slov byly mimoslovníkové predikce 194.

## 5.14 Úspěšnost

Poslední naměřená word error rate markovovského akustického modelu s výše zmíněným jazykovým modelem je 46,3%. Nejnižší dosažená WER s pentagramovým jazykovým modelem popsáním v sekci 5.6 je **10,3%**.

Jelikož jsem zpočátku práce neměl žádná přepsaná data, určil jsem fixní testovací sadu až v průběhu projektu. Chybí v ní přepisy těch nejméně srozumitelných nahrávek, které byly digitalizovány až po provedení většiny experimentů, a navíc k nim pořizovat ruční přepisy je velice nesnadné.

Tabulka 5.5 uvádí nejdůležitější milníky ve word error rate s použitím trigramového jazykového modelu natrénovaného na Makoňových spisech a manuálních přepisech. Je s podivem, že model dotrénovaný na Makoňových nahrávkách má na obecné testovací množině větší úspěšnost (v tabulce tučně) než výchozí model trénovaný na všech datech (viz tabulku 5.4). Dalo by se čekat, že když se po konvergenci na obecné trénovací množině aplikují ještě další trénovací epochy na jiných specifických datech, měla by úspěšnost stoupnout na oněch specifických datech a poklesnout na datech obecných. Příčinu tohoto jevu jsem dosud neměl příležitost prozkoumat.

Je úspěšnost nízká nebo vysoká? V roce 2016 publikovali Mizera et al.[79] sadu receptů na rozpoznávání řeči pro češtinu. Uvádějí jako state of the art techniky GMM-HMM a DNN-HMM a word error rate pro jednotlivé recepty mezi 8,49 a

<sup>10</sup>Častěji s pomlčkami *out-of-vocabulary*, což je správně jen je-li fráze rozvitím jiné jmenné fráze, jako např. ve výrazu *out-of-vocabulary words*, stejně jako např. *part of speech*, kteréžto výrazy mnozí mylně píší s pomlčkami vždy.

model	GMM	DNN trénovaný na Makoňovi	DNN trénovaný na různých sadách
standardní testovací množina (50min)	46,3%	19,2%	<b>13,0%</b>
5 minut přebuzených záznamů		45,0%	34,8%
5 minut záznamů pořízených nízkou rychlostí		68,5%	42,1%
1 minuta obzvláště nesrozumitelného záznamu		94,1%	75,9%
agregovaná testovací sada z různých dat		77,3%	<b>22,2%</b>

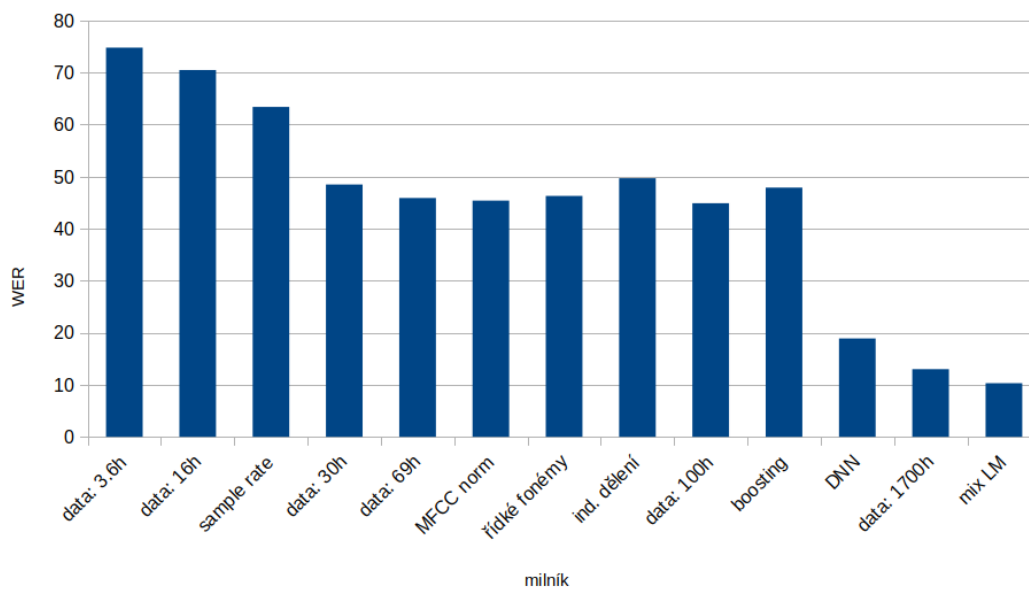
Tabulka 5.5: Chybovosti tří důležitých modelů na různých testovacích sadách s trigramovým jazykovým modelem trénovaným na Makoňových prepisech a spisech.

48,47 podle povahy dat. Úspěšnost na korpusu Karla Makoně v tomto porovnání je velmi dobrá, obzvláště s přihlédnutím k velké akustické variabilitě materiálu.

Na závěr kapitoly uvádím na obrázku 5.11 vývoj úspěšnosti automatického přepisu mluveného korpusu Karla Makoně od prvních experimentů po současnost.

- První sloupec odpovídá iniciálnímu přepisu modelem založeným na systému HTK natrénovaným na necelých čtyřech hodinách vlastních prepisů pořízených pomocí prototypu webové aplikace.
- Druhý, čtvrtý, pátý a devátý sloupec odpovídají milníkům v nárůstcích přepsaného materiálu použitého jako trénovací data.
- Třetí sloupec značí odstranění triviální chyby použití nesprávné vzorkovací frekvence, tedy přechod ze 44100 Hz na 16 kHz.
- Šestý sloupec koresponduje s experimentem keprstrální normalizace popsaným v sekci sec:mfcc-norm.
- Sedmý sloupec odpovídá pokusu o přidání řídkých hlásek:  $\underset{\sim}{a}$ ,  $\underset{\sim}{\eta}$ ,  $\underset{\sim}{o}$ : a  $\underset{\sim}{d}$ .
- Osmý sloupec odpovídá pokusu štěpit složky ve směsích modelů jednotlivých hlásek, nikoliv všechny najednou.
- Desátý sloupec odpovídá pokusu o rozšíření trénovací sady o automaticky přepsané pasáže mluveného korpusu Karla Makoně s velkou měrou jistoty predikce.
- Jedenáctý sloupec odpovídá prvnímu modelu na bázi DeepSpeech, kde k trénování bylo použito týchž sto hodin Makoňových nahrávek jako u předcházejícího modelu na bázi HTK v devátém sloupci.
- Dvanáctý sloupec odpovídá modelu trénovanému pomocí DeepSpeech na sedmnáctiset hodinové trénovací sadě z různých zdrojů.

- Třináctý sloupec odpovídá použití jazykového modelu natrénovaného na směsi Makoňových slov a výboru z korpusu WMT.



Obrázek 5.11: Vývoj úspěšnosti přepisu.

# 6. Webové rozhraní

## 6.1 Porovnání s jinými scénáři

Mým dílčím cílem je co nejlepší ortografický<sup>1</sup> a fonetický přepis tisícíhodinového korpusu jednoho mluvčího rozděleného do přibližně hodinových celků. Lidé, kteří se o tyto nahrávky zajímají a chtějí je studovat, představují potenciál, který mohu využít ke své práci, a zároveň cílovou skupinu, jimž bude produkt sloužit.

Webová aplikace by tedy měla skloubit tyto dva účely:

1. Sloužit uživatelům, aby mohli materiál co nejlépe konzumovat.
2. Navést uživatele, aby vydali co nejkvalitnější příspěvek.

Neznám žádný projekt se srovnatelným východiskem. Můžeme však porovnávat jednotlivé aspekty, vyskytující se v jiných aplikacích.

### 6.1.1 Programy pro přepis

Nástroje určené pro manuální přepis zvukového záznamu do textu představují typ aplikace, který je mému případu podobný a zároveň rozšířený. Porovnejme tyto dva úkoly a zpytujeme hlavní rozdíly. K porovnání vezměme

1. Transcriber<sup>2</sup>, klasický svobodný program napsaný v TCL,
2. oTranscribe<sup>3</sup>, svobodný moderní webový přepisovací nástroj a
3. Transcribe<sup>4</sup>, komerční webový přepisovací nástroj.

Každé číslo v seznamu níže označuje program, pro který platí ten který výrok. Ku příkladu jen Transcriber umožňuje anotaci mluvčích, proto u druhé položky stojí pouze číslo (1).

---

<sup>1</sup>Na pravopis jako takový se důraz neklade. Ortografickým přepisem myslím standardní zápis.

<sup>2</sup>trans.sourceforge.net

<sup>3</sup>otranscribe.com

<sup>4</sup>transcribe.wreally.com

- *přepisovací programy:*
- jsou optimalizované pro pořizování přepisu od nuly; (1,2,3)
- umožňují anotaci mluvčích; (1)
- nepotřebují kontrolu kvality: uživatel může přepisovat dle libosti a konečným měřítkem je jeho vlastní spokojenost; (1,2,3)
- zarovnávají na úrovni frází, pokud vůbec; (1)<sup>5</sup>
- točí se kolem uživatele: každý může přepisovat libovolná data; (1,2,3)
- předpokládají, že přepisovat je uživatelův záměr; (1,2,3)
- nesdílejí data mezi uživateli; (1,2)<sup>6</sup>
- *moje aplikace:*
- vždy vychází z existence předchozího přepisu;
- předpokládá, že všechna slova pocházejí od jednoho mluvčího;
- vyžaduje přesnost přepisu neboť tento je použit pro trénování statistických modelů;
- zarovnává na úrovni slov, interně na úrovni hlásek;
- točí se kolem dat: sbírka nahrávek je středobodem aplikace a přepisovat lze pouze ji;
- předpokládá, že uživatel chce poslouchat, vyhledávat nebo číst s poslechem, a k přepisu ho nutno motivovat;
- musí počítat s kolizemi, kdy více uživatelů upravuje týž úsek.

Navzdory těmto odlišnostem se v přepisovacím softwaru skrývá mnoho poučného. Snadnost provádění běžných úkonů, jako pozastavení a obnovení přehrávání či posun, jsou stěžejní pro uživatelský prožitek (*UX*) a tím i pro množství a kvalitu příspěvků. I způsob synchronního zobrazení textu se zvukem má velký dopad a v potenciálních přístupech je značná svoboda pro variaci.

## 6.1.2 Wiki

Kde se moje aplikace odchyluje od přepisovacích programů, tam do značné míry připomíná wiki: komunitní platformu, která slouží uživatelům včetně příspěvů, ale kde kvalita příspěvků je podstatná, zatímco samotná spokojenost příspěvovatele nemá takovou důležitost.

Jeden podstatný rozdíl oproti wiki je, že wiki je kreativní, kdežto náš úkol je mechanický. Uživatel má pramálo prostoru pro vlastní invenci: poskytnutí jiného než doslovného přepisu se vnímá jako chyba.

Populární wiki mají dobrá opatření pro konflikty v editacích, což je oblast, kde bych se mohl nechat poučit. Zatím k tomu ale nebyl důvod, protože pokud vždy použiji nejnovější verzi každého segmentu, výsledek zůstane konzistentní, i když segment od uživatele A padne do širšího přepisu od uživatele B.

Nově odeslaný segment přepisu vždy přepíše stávající verzi, ale každý lze vrátit zpět (*undo*), neboť všechny příspěvky udržuji v databázi. Lze také shlukovat

<sup>5</sup>Transcriber explicitně zarovnává text s mluveným slovem, oproti zbylým dvěma, které toliko umožňují přidání časových značek do přepisu.

<sup>6</sup>Transcribe podporuje kolaborativní přepis

příspěvky podle autora či jinak, ale zatím něčeho takového nebylo zapotřebí.

### 6.1.3 Korpusy

Tento projekt není prvním, který zahrnuje komunitní péči o korpus. Za zmínku stojí Manually annotated sub-corpus[80], kde se anotace různého druhu střádají od dobrovolníků. Dále Wikicorpus[81], korpus článků z Wikipedie s určitou úrovní lingvistické anotace. Můj projekt by s nimi mohl v budoucnu dosáhnout značné podobnosti, až se hlavní bod zájmu stočí od samotného přepisu k anotaci.

Je zde také CzEng[82], česko-anglický paralelní korpus, kde velká část překladů pochází od dobrovolníků. Podobnost ve východisku je zde pozoruhodná, neboť v obou projektech se z původního materiálu dojde k derivátu pomocí počítačového zpracování, přičemž chyby se pak komunitně opravují. V případě CzEngu jde o strojový překlad, v mém o strojový přepis. Nicméně specifika projektů přinášejí odlišné problémy a diktují odlišné přístupy.

Marge (2009)[83] zkoumá použití platformy Mechanical Turk k získání přepisů mluveného slova. Mihalcea (2004)[84] prezentuje webové rozhraní pro dezambiguaci významu slov (*word-sense disambiguation*) a zaměřuje se především na ošetření neshod mezi anotátory.

## 6.2 Popis webové aplikace

Webová aplikace sloužící k přehrávání a sběru přepisů existuje ve dvou verzích. Prototyp byl implementován hned za začátku projektu v roce 2012 a popisuje ho článek Krůza, Peterek (2012)[85]. Druhá, modernější verze začala vznikat na konci roku 2016 a poprvé byla nasazena na konci září 2017. Tu popisuje článek Krůza, Kuboň (2018)[86].

### 6.2.1 Prototyp

První implementace byla založena na přehrávači *jPlayer*, modulu pro knihovnu *jQuery*, který využívá standard HTML5 s jeho elementem `<audio>` a technologii *Adobe Flash*. Pro dynamickou odezvu zobrazených prvků na změny v datovém modelu jsem použil knihovnu *knockout*.

Aplikace měla formu jediné stránky s rozbalovatelným výběrem nahrávky, ovládacími prvky přehrávače a třemi řádky přepisu. Při označení části zobrazeného přepisu se stránka překryla rozhraním pro opravu přepisu, jež zvu *editačním okénkem*. V editačním okénku se zobrazilo vstupní pole (`<textarea>`) s předvyplněným současným přepisem, ovládací prvky pro přehrání odpovídající pasáže, odeslání opraveného přepisu a opuštění editačního okénka. Rozhraní prototypu ukazuje obrázek 6.1.

Nad rámec výše popsaných funkcionalit přibyly další na základě přání uživatelů a autorovy potřeby:

- indikace, do jaké míry je která nahrávka přepsána<sup>7</sup>,
- manuální posouvání hranic přepisovaného zvukového úseku<sup>7</sup>,

---

<sup>7</sup>Tato funkcionalita momentálně není implementována v nové verzi aplikace.





Bychom si měly vědomi toho, že i naše lidství je **prostředkem**, k to jen prostředkem a že je pro všechny lidsky bylo už je konec myšlenek a smysl než tím. Tak potom toho že my aniž si už je li i při prostředků **projevu toho lidství, tak by jsme mesměly zjistit že to všechno**

výskyt

prostředkem.

forma

prostředkem

výslovnost

prostř'etkem

pozice

0:04.14

přehrát

[manuál](#) data © 1960 - 1992 Karel Makoň; k dispozici pod licencí [CC BY-SA 3.0](#)

Obrázek 6.1: První verze webové aplikace.

- úprava zápisu slova s ponecháním výslovnosti,
- identifikace uživatelů včetně sezení, prohlížeče atp.,
- vyhledávání v prepisech.

Tato původní verze posloužila k přepsání asi 600 tisíc slov a běžela asi 5 let, než bylo nutné ji nahradit.

Pro kompletní přepis aplikace se postupně objevilo několik důvodů. Hlavním z nich bylo, že původní aplikace mohla kvůli své designové nevytříbenosti jen těžko sloužit pro širokou veřejnost. Dalším důvodem bylo, že některé kýžené funkce nebylo možné zprovoznit bez zásadních změn v provedení. Především šlo o ekvalizér, čili frekvenční korekci při poslechu. Akutním důvodem pak byl fakt, že všechny významné prohlížeče opouštěly podporu Flashe.

## 6.2.2 Základní rysy druhé verze

Pro novou verzi jsem zvolil technologie React + Redux[87] jako aplikační rámec, Web Audio API[88] jako platformu pro nakládání se zvukem a Twitter Bootstrap jako základ pro vzhled prvků. Zdrojový kód píšou v ECMAScript 6 a o kompilaci se stará webpack.

Aplikace sestává z několika *pohledů*<sup>8</sup>:

1. úvodní stránka se seznamem nahrávek, kde každý záznam odkazuje na detailní pohled,

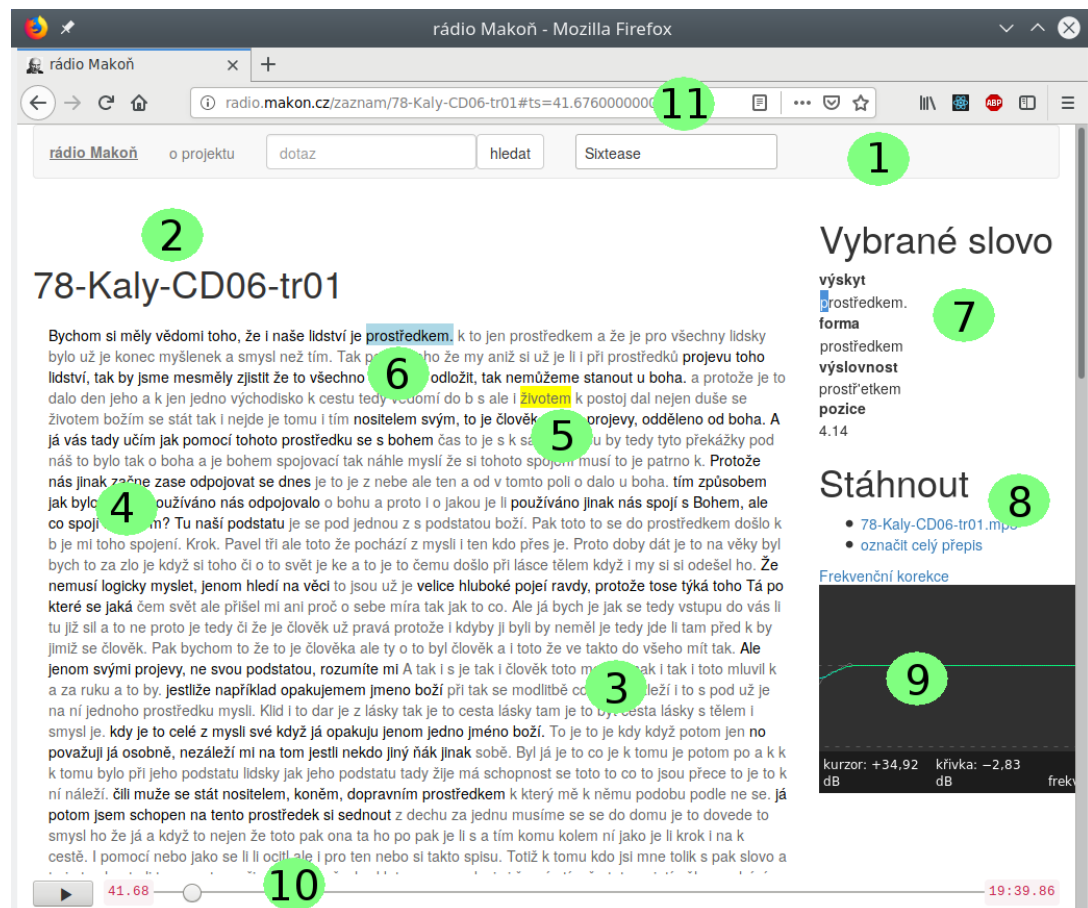
<sup>8</sup>Pohled ve smyslu *view* z architekturního přístupu *model - view - controller*. Podobně jako autoři frameworku Django, pojmem pohled (*view*) míním jednu stránku definovanou cestou v URL i s její funkcionalitou.

2. detail nahrávky, kde je možné přehrávání a zobrazuje se přepis, který se dá editovat,
3. výsledky vyhledávání, kde každý záznam obsahuje úryvek odpovídající vyhledávanému dotazu a odkazuje na příslušnou pasáž nahrávky v detailním pohledu,
4. různé statické podstránky s obecnými informacemi, manuálem atd.

Úvodní stránka má dvousloupcový formát, kde vlevo je rozbalovací seznam kategorií a vpravo lineární seznam nahrávek. Jednotlivé kategorie jsou pak skrolovacími odkazy do pravého sloupce a podle stupně skrolování se příslušná kategorie sama rozbalí (tzv. *scrollspy*).

Pro lepší přehlednost a v souladu s principem *separation of concerns* je seznam nahrávek pouze na úvodní stránce.

Podrobněji se budu zabývat pouze pohledem detailu nahrávky. Obrázek 6.2 ukazuje rozhraní v průběhu přehrávání. Obrázek 6.3 ukazuje rozhraní při editaci segmentu.



Obrázek 6.2: Webové rozhraní při přehrávání.

### Vysvětlivky k obrázku 6.2:

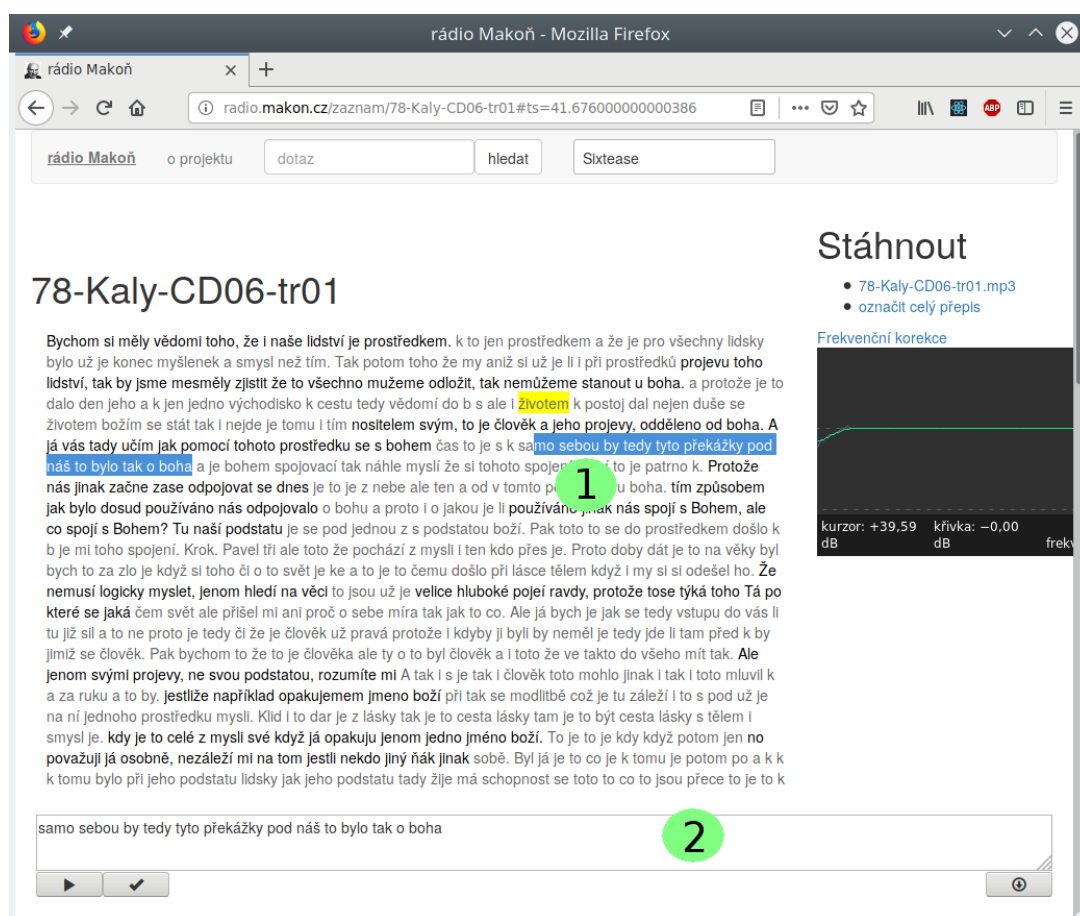
#### 1. Záhloví a v něm

- jméno aplikace odkazující na úvodní stránku,

- odkaz na informace o projektu,
  - vyhledávací políčko,
  - vstupní pole pro uživatelskou přezdívku.
2. Identifikátor nahrávky.
  3. Automaticky přepsané segmenty v šedi.
  4. Manuálně přepsané segmenty v černi.
  5. Právě přehrávané slovo zvýrazněné žlutým pozadím.
  6. Označené slovo zvýrazněné odstínem modři „st. regent“ na pozadí.
  7. Informace o označeném slově:
    - výskyt: slovo s kontextuálním velkým písmenem a interpunkcí, jak se nachází v textu (navíc právě editované, jak prozrazuje označené iniciální písmeno),
    - forma: normalizovaná slovní forma, jak se objevuje ve slovníku,
    - výslovnost: český fonetický zápis použité výslovnosti (viz podsekcí 6.5.3),
    - pozice: čas v sekundách od začátku nahrávky do začátku slova.
  8. Ukládání:
    - přímý odkaz k celé nahrávce ve formátu mp3,
    - označení celého přepisu pro snadné vložení (*copy-paste*).
  9. Grafický ekvalizér pro kompenzaci úzkopásmového šumu.
  10. Ovládací prvky přehrávání:
    - tlačítko pro pozastavení / pokračování,
    - současná pozice,
    - posuvník (*scrollbar*) přehrávání,
    - celková délka nahrávky.
  11. aktuální pozice reflektovaná v URL.

### Vysvětlivky k obrázku 6.3:

1. Označení textového úseku myší definuje segment k editaci tak, že označené části slov se doplní na celá;
2. Editační okénko a v něm:
  - textové pole (*textarea*) předvyplněné stávajícím přepisem,
  - tlačítko pro přehrání odpovídajícího segmentu,
  - tlačítko pro uložení,



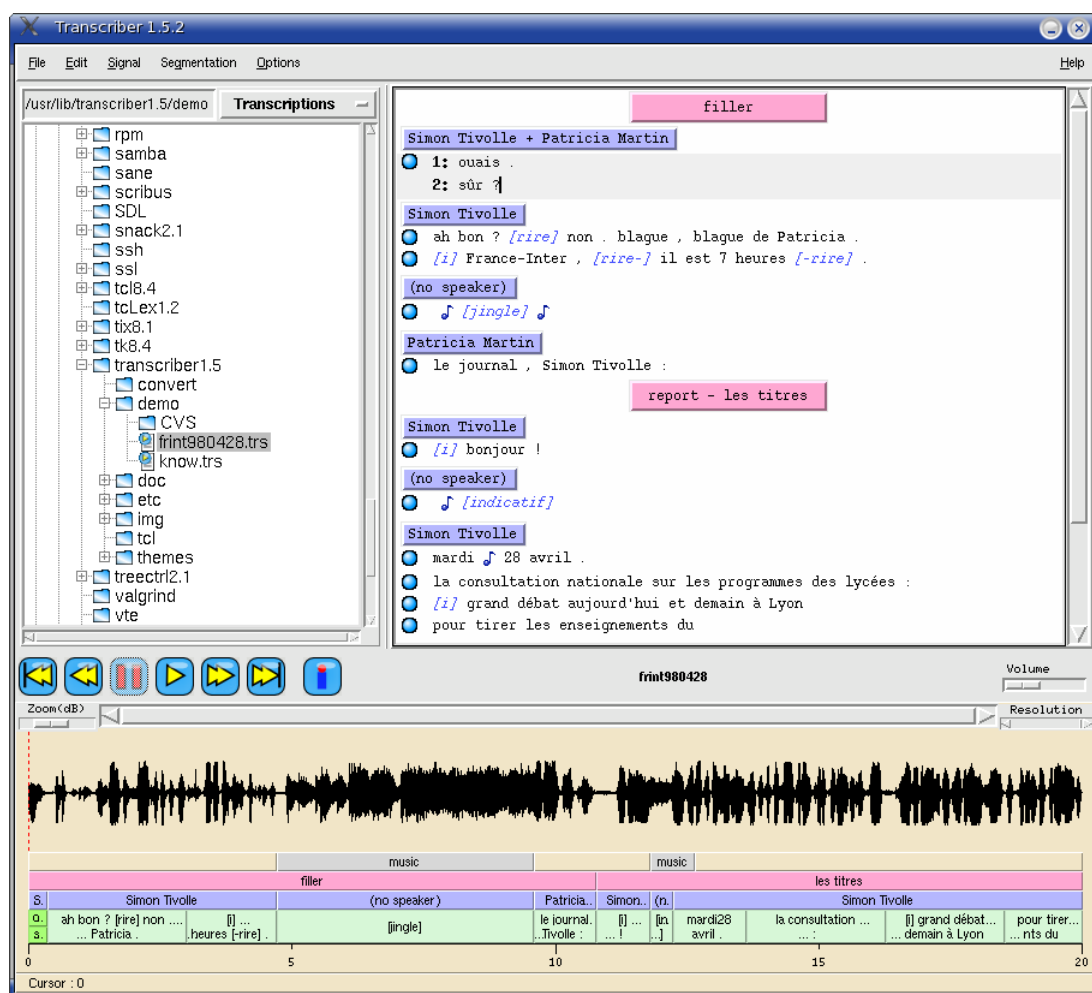
Obrázek 6.3: Rozhraní ve stavu editace segmentu.

- tlačítko pro stažení segmentu, které inicializuje operaci uložení souboru pro úsek audia odpovídající označenému textu. Syntéza uloženého souboru se odehrává v prohlížeči.

Nejčastější úkony mají klávesové zkratky: `ctrl+mezerník` pro přehrání / pozastavení a `ctrl+enter` pro uložení korekce.

### 6.2.3 Zobrazení přepisu

Mnohý program pro přepisování ukazuje transkript jako vertikální seznam vyřčených frází, viz obrázek 6.4 pro příklad z Transcriberu. To připisují faktu, že atomickými prvky přepisu jsou uživatelem definované fráze a jejich hranice jsou spolehlivé. Není na programu, aby je definoval nebo zpochybňoval. V mém případě jsou atomickými prvky slova. Ano, jsou zde i věty, ale segmentace na věty automatickým přepisovačem je velice nespolehlivá, takže je žádoucí, aby označení a přepsání segmentu, který přesahuje přes hranice věty, bylo přirozené a snadné.



Obrázek 6.4: Uživatelské rozhraní Transcriberu.

Toto je jeden z důvodů, proč zobrazují přepis v podstatě jako jeden zalomený rádek.

## 6.2.4 Problém s rychlostí

Na zobrazení přepisu byly kladeny tyto požadavky:

1. Právě přehrávané slovo aby bylo zvýrazněno.
2. Manuálně přepsané segmenty aby byly jasně odlišené od automaticky přepsaných.
3. Označení neprázdné množiny znaků (krom mezery) myší aby spustilo editační mód pro označený text doplněný na celá slova; při úspěšném uložení změny aby se tato vmísila do zobrazeného textu.
4. Kliknutí na slovo aby o něm vyvolalo zobrazení kontextových informací (toto zvu „*vybrané slovo*“, neb pojem „*označené slovo*“ je již obsazen).
5. Celý přepis aby byl viditelný najednou pro možnost vyhledávání.
6. Stránka aby byla responzivní.

Sklobit tyto požadavky je obtížnější, než by se mohlo zdát. Zejména responzivita se těžko slučuje s ostatními body. Proč?

Body 1 až 4 volají po tom, aby každé slovo bylo obaleno ve vlastním elementu. Bod 5 a medián počtu slov v nahrávce zvíci šesti tisíc dávají dohromady šest tisíc elementů `<span>` jen pro statické zobrazení textu.

Může se zdát, že to není takový problém, ale ovlivňuje to responzivitu a paměťovou náročnost stránky.

V původní verzi webové aplikace jsem toto vyřešil obětováním bodu 5. Zobrazovaly se jen tři řádky textu, přičemž právě přehrávané slovo se vždy drželo v tom prostředním. Ukazuje to obrázek 6.1, jen s tím rozdílem, že právě přehrávané slovo je v prvním řádku, poněvadž se přehrává začátek nahrávky. Díky pokroku ve webových standardech a jejich podpoře ze strany prohlížečů je nyní možné řešení.

## 6.2.5 Řešení

Můžeme využít šťastného faktu, že ručně přepsaná slova a automaticky přepsaná slova mají tendenci se shlukovat. Průměrný počet slov v jednom příspěvku je 7,9. Navíc drtivá většina takových segmentů bezprostředně navazuje na další ručně přepsané segmenty.<sup>9</sup> Z toho plyne, že obalení každého souvislého shluku manuálně či automaticky přepsaných slov do zvláštního HTML elementu nepředstavuje problém. Tím se řeší bod 2.

Bod 3 lze implementovat s použitím metody objektového modelu dokumentu (*DOM*) `document.selection` a objektů `Range`, které umožňují nalézt nejhlubší HTML elementy a pozice v jejich textu, kde začíná a končí označený úsek. Díky tomu, že délka slov je známa, mohu z označeného úseku dovodit odpovídající slova v přepisu.

---

<sup>9</sup>Medián počtu shluků je 1 (většina nahrávek nemá žádné ručně přepsané slovo), maximum je 1109. Medián pouze z nahrávek, které obsahují ruční korekce, je 8.

Body 1 a 4 se dá implementovat dvěma způsoby: Buďto zabalením přehrávaného a vybraného slova do zvláštního elementu anebo vykreslením zvýrazňujícího obdélníku na pozadí slova.

Výhodou obalení slov elementem by byla větší robustnost a menší náchylnost k chybám. Nicméně neustálé změny v DOMu při přehrávání s potenciálními častými operacemi *reflow*<sup>10</sup> hovoří proti tomuto řešení. Nalézt přesné pozice slova a vykreslení obdélníku přesně pod ním<sup>11</sup>, vyhnout se chybám v pozicování a udržet obdélník na správném místě i po změně velikosti okna či odskrolování, to je dozajista výzva, nicméně přesto jsem zvolil tuto cestu. Navýšení výkonnosti pro běžné používání převažuje potenciální chyby v okrajových případech, nanejvýš když eventuální chyby nejsou kritické a zmizí při dalším přehrávání.

Efektivitu repositionování obdélníku podporuje i fakt, že se dají dopředu spočítat souřadnice všech slov najednou a pak je přepočítat jen ve dvou případech: 1) při zřídka události změny velikosti okna a 2) když se opravený segment vkládá do zobrazeného přepisu a je tedy nutno souřadnice přepočítávat pouze pro slova, která jsou v dokumentu za vloženým segmentem.

Dalo by se optimalizovat dále a zastavit přepočet v momentě, kdy se nějakému slovu nezmění horizontální souřadnice. Pak by stačilo připočítat rozdíl ve vertikální souřadnici všem následujícím slovům. Jinými slovy, pokud jeden řádek zůstane stejný, pak i všechny pod ním.

## 6.2.6 Vizuální odlišení manuálního a automatického přepisu

Jak je vidět na obrázku 6.2, vykresluje se automatický přepis v šedé barvě a manuální v černé. Proč jsem zvolil rozlišení barvou a ne standardním versus tučným písmem? Za prvé, standardní písmo je optimalizované pro čtení. Tučné písmo je na to, aby bodově zvýrazňovalo úseky. Když se použije na dlouhé pasáže, působí těžkopádně. Automatický přepis obsahuje mnoho chyb, takže nedává smysl ho optimalizovat pro ideální četbu.

Je ještě jeden praktický důvod. Když se varianty písma liší pouze v barvě, nikoliv ve velikosti, a když segment automatického přepisu je beze změny odeslán jako zkorigovaný, pak jeho vložení do textu nezpůsobí reflow, což šetří výpočetní kapacitu a napomáhá responzivité. Může se to zdát jako okrajový případ, ale domnívám se, že identifikace správně automaticky přepsaných slov je legitimní způsob přispívání, tak proč ho neoptimalizovat?

Automaticky přepsané úseky balím i tak do elementů `<span>` a manuálně přepsané do elementů `<b>`, protože potom se rozlišení zachová při zkopírování textu z webové stránky do textového editoru podporujícího formátování.

## 6.2.7 Web Audio API

Přechod na tuto technologii umožnil některé pokročilé funkce, avšak za relativně vysokou cenu. Web Audio API je standard pro pokročilé zpracování zvukového signálu v prohlížeči. Základním konceptem je graf procesních uzlů, které mají vstup a výstup a mohou se libovolně propojovat. K dispozici jsou zdroje

<sup>10</sup>Při operaci reflow prohlížeč přepočítává pozice všech elementů a překresluje je.

<sup>11</sup>Pod ním na ose Z. Přes něho na osách X, Y.

zvuku jako oscilátory nebo přehrávače streamů, souborů (tag `<audio>`) a dat v paměti (`AudioBuffer`) a efekty jako zesílení, dynamická komprese, či mixování kanálů.

Velká výhoda Web Audio API oproti elementu `<audio>` je možnost přesného časování až k jednotlivým `samplem`. Přehrávání výseku odpovídajícího označenému textu se proto nemusí provádět pomocí časovače `setTimeout`, který je velice nepřesný.

Bez Web Audio API by také nebylo možné provádět frekvenční korekci při poslechu, čili mít tzv. *ekvalizér*. Ten je zapotřebí, protože některé nahrávky mají v určitém frekvenčním pásmu silný šum, jehož odstranění je s ekvalizérem snadné a komfort poslechu se tak razantně zvýší.

Další funkcí, kterou Web Audio API umožňuje, je stahování úseků. Označením přepsaného textu se definuje úsek nahrávky a ten je možné uložit bez dalšího síťového přenosu. Tato funkce však vyžaduje, aby nahrávka byla dekodovaná v paměti. Vzhledem k tomu, že nahrávky mají běžně i hodinu a půl, trvá její stažení a dekodování opravdu dlouho a navíc prohlížeč kvůli tomu spotřebuje přes gigabyte operační paměti.

Jsou plány na to, aby Web Audio API umožnila dekodovat jen část nahrávky<sup>12</sup>, avšak palčivost problému mne přiměla nečekat, viz následující sekci.

Díky tomu, že Web Audio API umožňuje přehrávání binárních dat z proměnné v paměti, nabízí se dekodovanou nahrávku uložit na persistentní úložiště uživatele počítače a při opětovné návštěvě stránky data místo stahování odsud nahrát.

Moderní prohlížeče poskytují několik bran k úložišti na místním disku. Nejtradičnějšími jsou bezesporu *cookies*, které jsou však pro ukládání objemnějších dat zcela nepoužitelné. Velice slibnou se jeví *localStorage*, umožňující ukládání párů klíč-hodnota. I zde však narážíme na příliš omezující kvóty. Kupříkladu Firefox ji má na 10MB, přičemž potřeba je asi 1GB. Dalším kandidátem je *File System API*. Tento standard pro izolovaný souborový systém k dispozici webové aplikaci je zcela ideálním řešením – dá se zde i explicitně požádat o konkrétní diskovou kvótu a uživatel tak má volbu bez nutnosti práce programátora webové aplikace. Kamenem úrazu je zde však podpora, která se momentálně omezuje pouze na prohlížeče na bázi Chromia.

Existuje ještě standard *IndexedDB API*, který má uspokojivou podporu a uložení gigabytu dat je s ním možné, byť ne zaručené. S využitím abstrahující knihovny *Dexie* jsem proto skrz tento standard ukládání implementoval. Pro uživatele, kteří delší dobu pracují na jedné a téže nahrávce, se tím přináší velká úspora času a přenesených dat. Nicméně s rozdělením nahrávek na segmenty (sekce 6.4) přestala být potřeba ukládat nahrávky aktuální.

## 6.3 Nucené zarovnání

Nucené zarovnání, anglicky *forced alignment*, je inherentní součástí systému rozpoznávání řeči na bázi skrytých markovovských modelů, jak je i zmiňuji v podsekci 5.5.2. V mém případě však hraje zásadnější roli, neboť nucené zarovnání využívám při sběru trénovacích vzorků (viz podsekci 6.5.1) a při zpracování korpusu

---

<sup>12</sup>[github.com/WebAudio/web-audio-api/issues/1305](https://github.com/WebAudio/web-audio-api/issues/1305)



(viz sekci 5.8 a podsekcí 6.4.2).

Až do května 2021 jsem používal pro nucené zarovnání nástroj `HVite` ze sady `HTK` s akustickým modelem natrénovaným zvláště pro tento účel. Jednalo se o monofónový model s gaußovskými směsmi. Toto bylo dlouho jednou z největších slabín systému a prioritním krokem v plánované práci. Hlavní důvody k nahrazení tohoto řešení byly dva: Jednak model nedokázal dobře postihnout data v jejich variabilitě, takže korektní přepis netypických nahrávek byl často odmítnut a jednak zde byla technologická závislost na zastaralém systému `HTK`, a to i pro samotný provoz webové aplikace.

Jako technologie pro následníka se nabízí Kaldi svojí moderností a přítomností aparátu pro skryté markovovské modely. Systém *Montreal Forced Aligner*[89] (*MFA*) poskytl hotové řešení využívající zkušeností s předchozími generacemi zarovnávačů, jmenovitě *Prosodylab aligner*[90] a *FAVE*[91].

*MFA* využívá trojfázového tréninku. Nejdříve se natrénuje monofónový model založený na `GMM`, poté se přejde k trifónům a nakonec se provádí adaptace na jednotlivé mluvčí pomocí `fMLLR`.

*MFA* jsem se pokusil natrénovat na celém agregovaném korpusu (viz sekci 5.11). To se bohužel ukázalo jako výpočetně příliš náročné, proto jsem omezil trénovací data na 60 tisíc vzorků (vět), z původních 820 tisíc. Vzorky jsem vybíral tak, aby každý zdrojový korpus byl zastoupen přibližně stejně. Konfiguraci jsem ponechal tovární. Kvůli chybějící anotaci mluvčích u velké části dat a také pro obrovské množství jednotlivých mluvčích jsem adaptaci dělal po jednotlivých zdrojových korpusech. Jelikož korpus Karla Makoně je korpus jediného mluvčího, je identifikován jako jeden z mluvčích a mohu využít natrénování na jeho hlas.

*MFA* se osvědčil i jednoduchostí v natrénování a použití a též robustností výsledného systému – úspěšně zarovná i pětatřicetiminutový úsek. Zarovnávač je k dispozici pro libovolná data v češtině jako služba na Lindatu: `lindat.mff.cuni.cz/services/aligner`.

## 6.4 Rozdělení nahrávek na úseky

Vzhledem k tomu, že v roce 2019 nebyl kurzorový přístup ke zvukovým datům skrze `Web Audio API` v dohlednu (a není ani v lednu 2021), a jak odrazující dopad má nutnost stahovat a dekódovat celou nahrávku aspoň při jejím prvním načtení, nezbylo mi, než změnit způsob, jakým jsou nahrávky uloženy.

Nahrávky jsou uloženy v několika instancích pro různé účely:

1. na backendovém serveru ve formátu `MFCC` pro nucené zarovnávání,
2. v repozitáři `LINDAT` ve formátu `FLAC` za účelem archivace a bádání,
3. na `CDN`<sup>13</sup> ve formátu `mp3` za účelem přímého stažení uživatelem,
4. taktéž na `CDN` ve formátech `OGG/Vorbis` a `mp3` pro webové rozhraní.

Pouze poslední jmenovanou instanci je žádoucí ukládat tak, aby každý soubor byl jen tak velký, aby jeho stažení a dekódování trvalo únosně dlouho. V ostatních případech je lépe zachovat uložení, kde jedna nahrávka odpovídá většinou jedné

---

<sup>13</sup>content delivery network

straně kazety či jednomu průchodu pásky z kotouče na kotouč. Třetí a čtvrtá instance však navzdory rozdílnému účelu sdílejí tatáž data. Bylo proto nutné je duplikovat.

### 6.4.1 Délka segmentů

Délka úseků, na které nahrávky rozdělují, ovlivňuje, jak dlouho se každý segment bude stahovat a dekódovat. Čas stahování a dekódování segmentu, který obsahuje slovo, na němž je kurzor při prvním požadavku o přehrávání, je roven zpoždění od uživatelské akce k začátku přehrávání. Podle internetového periodika UXMovement[92], začíná uživatel po čtyřech sekundách čekání upouštět od předchozího záměru. Podle článku Nielsen Norman Group[93] je hranice únosnosti 10 sekund.

Pokud budou úseky příliš dlouhé, jejich stahování a dekódování zabere příliš mnoho času. Na druhou stranu s každým předělem vnášíme do přehrávání bod, kde se úseky nalepují a může tam vyvstat artefakt. Také s každým segmentem se pojí extra HTTP request s nezanedbatelnou režíí.

Jako vhodný kompromis se jeví segmenty o délce 30 - 120 sekund. Velikost dvouminutového segmentu je v komprimovaném jednokanálovém formátu při vzorkovací frekvenci 24kHz kolem 0,6MB a na Intel Core2 o 2,5GHz se dekóduje asi 1,6 sekundy.

### 6.4.2 Metody hledání bodů předělu

Vhodným výběrem bodů předělu můžeme omezit dopad případných artefaktů způsobených nepřesným navázáním. Ideálním by bylo dělit nahrávky v momentech ticha. Ne vždy jsou momenty ticha každé dvě minuty, proto z momentů ticha ustupme k požadavku pauzy v řeči. Hovořit dvě minuty bez nádechu hraničí s nemožností. Potýkáme se tedy s úlohou nalézt pauzy v řeči. Jednak je třeba ujasnit, podle jakého klíče budeme pauzy vybírat, a jednak, jak je budeme přesně hledat.

Hledat pauzy v řeči lze různými způsoby. Nejspolehlivější a nejnáročnější je manuální označování pauz. Pokoušel jsem se o to sám a dosáhl jsem rychlosti přibližně čtyřnásobku rychlosti přehrávání, tedy jeden zapsaný bod předělu za třicet sekund.

Další velice spolehlivou metodou je hledání podle predikovaných pseudofónů ticha v zarovnaném přepisu. Tuto metodu jsem mohl namnoze použít, neboť k většině nahrávek mám automatický nebo i manuální přepis.

Tam, kde pořízení přepisu nebo jeho automatické zarovnání selhalo, lze použít detekci ticha prostou akustickou analýzou. Tato metoda je velice náchylná k chybám v případě nahrávek s malým poměrem signálu k šumu, a těch je v korpusu Karla Makoně mnoho<sup>14</sup>.

Kde nepomůže ani metoda detekce ticha, což se pozná podle toho, že detekované pauzy jsou příliš daleko od sebe nebo naopak zabírají valnou část nahrávky, nezbyvá, než určit body předělu ve fixních intervalech, nehledě na to, že jich mnoho padne doprostřed slova.

---

<sup>14</sup>Později jsem se dozvěděl o existenci WebRTC-VAD, který funguje relativně spolehlivě a plánuju ho použít jako fallback pro nepřepsané nahrávky.

Pokusy dvě metody vyloučily: Manuální hledání bylo příliš neefektivní. Kromě mne se dalších asi pět dobrovolných anotátorů o tento úkol pokusilo a došla jim trpělivost po nule až deseti minutách označovaného materiálu. I na některé nahrávky, u nichž přepis selhal, šlo detekci pomocí zarovnaného přepisu použít. Rozdělily se na menší části, tyto se přepsaly, zpravidla s malou úspěšností. Tento přepis opět v některých případech selhal, ale většina takové nahrávky byla nějakým přepisem pokryta. A jakkoliv nekvalitní takový přepis byl, právě dlouhé mezery mezi slovy se našly s uspokojivou přesností. Krátké úseky, na nichž selhalo rozpoznávání řeči, byly pak příliš obtížné i pro detekci pomocí ticha. Jednalo se o úseky bez řečových událostí, nebo s extrémním šumem.

Celkový počet různě získaných bodů předělu shrnuje tabulka 6.1<sup>15</sup>.

metoda získání	počet použitých
manuálně	0
podle zarovnaného přepisu	60424
podle detekce ticha	0
fixní délkou	22043
celkem	82467

Tabulka 6.1: Počet bodů předělu podle metody jejich získání.

### 6.4.3 Výběr bodů předělu

V metodě určování bodů předělu pomocí fixního intervalu jsem zvolil délku šedesáti sekund. Výše rozvádím, že to je délka přijatelná, a další optimalizací tohoto parametru jsem se nezabýval.

Zajímavější je situace u hledání pomocí zarovnaného přepisu. Zde se jedná o programátorský úkol, kde na vstupu máme posloupnost slov vyskytujících se v přepisu nahrávky, z nichž každé s sebou krom své formy a výslovnosti nese informaci, kde začíná, a pokud obsahuje na konci ticho, pak kde začíná pseudofón ticha a jak je dlouhý. Vstup tedy můžeme redukovat na posloupnost párů čísel, kde první vždy udává počátek ticha a druhé jeho konec. Na výstupu očekáváme posloupnost časových pozic, které rozdělují nahrávku na úseky o délce nejméně 30 sekund, nejvýše 120 sekund, a které jsou uprostřed co nejdelších tich.

Povšimněme si, že úloha nemá řešení, pokud je nahrávka kratší třiceti sekund. To ovšem v mluveném korpusu Karla Makoně nenastává a ani v opačném případě by to nevadilo, protože takovou nahrávku bychom nechali v jednom souboru.

Úloha nemá řešení ani v případě, kdy mezi dvěma sousedními tichy je rozestup větší než 120 sekund. Takový případ nastává, když je samotné detekované ticho velmi dlouhé. Tyto případy jsem řešil manuální úpravou.

Hledaný algoritmus se zdá být typickým příkladem pro dynamické programování: Nalezneme ideální rozdělení nahrávky, která obsahuje jen první slovo, a poté přidáváme slova, načež na základě dosavadního řešení a nového slova řešení rozšiřujeme.

<sup>15</sup>Vysoký počet předělů fixní délkou je způsoben tím, že k některým nahrávkám jsem doposud nepořídil zarovnaný přepis s vyznačením délky ticha.

Je ale i jednodušší varianta: Začneme s množinou všech tich a iterujeme přes ně od nejkratšího po nejdelší. Ticho z množiny odebereme, pokud sloučením sousedních segmentů nevznikne segment delší než 60 sekund. Přes vybraná ticha znova iterujeme a ticho odebereme, jestliže jeden z jeho sousedů má méně než 30 sekund.

Zbývá množina tich splňuje počáteční podmínky, pokud to je vzhledem ke vstupním datům možné. Algoritmus je jednoduchý na naprogramování a má milou lineární složitost. Pseudokód algoritmu:

```
const maxSegmentLength := 60
const minSegmentLength := 30

// vstupní posloupnost tich daných délkou a pozicí středu
silences := [
  { 'center': 1.23, 'length': 0.5 },
  { 'center': 3.45, 'length': 0.7 },
  ...
]

// přidej odkazy na sousedy
forEach (i in 0 .. input.length - 2) {
  silences[i + 1]['left'] := silences[i]
  silences[i]['right'] := silences[i + 1]
}
silences[0]['left'] := { center: 0 }
silences[-1]['right'] := { center: recordingLength }

// seříd od nejkratší
toFilter := silences.sort((a,b) => a['length'] <=> b['length'])

function drop(silence) {
  silence['left']['right'] := silence['right']
  silence['right']['left'] := silence['left']
}

// zahoď ticho, pokud nevznikne příliš dlouhý segment
filtered := []
forEach (silence in toFilter) {
  postMergeLength
    := silence['right']['center'] - silence['left']['center']
  if (postMergeLength > maxSegmentLength) {
    filtered.push(silence)
  } else {
    drop(silence)
  }
}

// zahoď ticho, pokud je segment příliš krátký
toFilter := filtered
filtered := []
forEach (silence in toFilter) {
```

```

leftLength := silence['center'] - silence['left']['center']
rightLength := silence['right']['center'] - silence['center']
if (min(leftLength, rightLength) < minSegmentLength) {
  drop(silence)
} else {
  filtered.push(silence)
}
}

return filtered.map(a => a['center']).sort()

```

#### 6.4.4 Pojmenování souborů

Jsou-li vybrány body předělu, mohou se nahrávky podle nich rozdělit a výsledné segmenty uložit na disk. Zde vyvstává otázka, jak rozdělit soubory do adresářů a jak je pojmenovat. Způsob uložení souborů hraje svou roli, jak dokládá i Reppen (2010)[94]. Zvolil jsem tento formát:

*ID/format/ID-from-ZACATEK-to-KONEC.PRIPONA*

tedy například

88-04A/ogg/88-04A-from-1155.27-to-1211.53.ogg.

Důvody jsou tyto: Není praktické z důvodu omezení mnoha souborových systémů mít příliš mnoho souborů v jednom adresáři. Používám proto rozdělení do adresářů podle identifikátorů nahrávek. Že formát je právě podadresář identifikátoru a ne třeba nadadresář, je arbitrární. Obojí by bylo možné, stejně jako mít soubory všech formátů v jednom adresáři a odlišovat je jen příponou.

Zopakovat identifikátor nahrávky i v názvu souboru jsem se rozhodl proto, aby případný zatoulaný soubor mohl být snáze identifikován. Díky tomu, že se do názvu souboru uvede začátek i konec úseku v rámci nahrávky, je zajištěno, že název souboru přesně popisuje jeho obsah. Oproti tomu např. lineární číslování by při změně bodů předělu vedlo k tomu, že jeden název souboru by byl totožný pro různé úseky v různých verzích korpusu. To by mohlo vést k problémům s uchováním. Odvozování identifikátorů na základě binárního obsahu souboru, např. pomocí kontrolních součtů, by vedlo k nutnosti změnit identifikátory při každé změně komprese apod., ačkoliv slyšitelný rozdíl by třeba nebyl žádný.

Pokud by při současném řešení došlo ke změně bodů předělu, by se nové i staré úseky musely uchovávat v jednom adresáři, a až by všechny reference na staré úseky byly vyhlazeny z paměti cache všech klientů, mohly by se staré smazat. Jedinou nevýhodou je, že staré a nové úseky by byly pomíchané v jednom adresáři, a proto by se musely při mazání explicitně vyjmenovat.

#### 6.4.5 Překryv úseků

Při testování se ukázalo, že vyřízneme-li pomocí programu `sox` úsek zvukového souboru, výsledný soubor skončí přehrávání o několik desetin sekundy dříve. Jako by chybělo posledních několik set samplů. Příčinu tohoto fenoménu zatím neznám. Kompenzoval jsem jej tím, že jsem každý úsek prodloužil o půl sekundy. Následkem toho bylo potřeba upravit přehrávání tak, aby každý úsek skončil

tehdy, až dohraje jeho metadaty daná délka, nikoliv až do skutečného vyčerpání zvukových dat.

## 6.5 Použití aplikace

Aplikace znamená použití. Použitelnost je tedy klíčovým faktorem pro její hodnocení.

### 6.5.1 Expertíza uživatelů

Přepis, který pořizují, je na hranici toho, co se dá nazvat lingvistickou anotací dat. V naší pozeňnané části světa, kde podíl analfabetů je zanedbatelný, můžeme přepis mluveného slova stěží nazvat odbornou prací. Na druhou stranu zajistit, aby přepis přesně odpovídal mluvenému projevu

- jakožto vyjádření vyřčených slov a jejich významu,
- na fonetické úrovni písmeno na hlásku
- a na časové ose

je za hranicemi toho, co se dá očekávat od nevyškoleného uživatele.

Lingvistická anotace dat obecně vyžaduje zaškolené pracovníky. Podíváme-li se např. na Pražský závislostní korpus, můžeme si povšimnout, že od anotátorů vzešla taková úroveň expertízy, že se stali spoluautory[95].

*Crowdsourcing*, přístup založený na komunitní spolupráci nebo zapojování dobrovolníků, nabývá na popularitě při získávání hodnot, které by jinak byly neúnosně drahé, viz podsekcí 6.1.3. Nicméně např. Maekawa (2000)[96] popisuje tvorbu mluveného korpusu spontánní japonštiny s využitím placených anotátorů.

Ve většině případů je kvalita pro anotaci dat velmi důležitá, proto je aspoň nějaká kontrola nezbytná, ať už je odbornost anotátorů jakkoliv vysoká. Je zřejmé, že čím méně expertízy na straně anotátorů, tím silnější kontroly je zapotřebí.

Běžnou metodou kontroly kvality je mezianotátorská shoda. To má obrovskou nevýhodu v tom, že každá část dat musí být anotována aspoň dvakrát, což snižuje výtěžnost nejméně o 50%.

Ještě jeden důvod hovoří proti jejímu použití v případě tohoto projektu. Webová aplikace je dělaná pro lidi, kteří chtějí poslouchat Makoňovy nahrávky z vlastního zájmu a jejich přínos pro kvalitu přepisu je spíše vedlejším produktem. Nebylo by snadné přesvědčit je, aby si vybrali právě nahrávku, kterou už někdo jiný přepsal.

Naštěstí lze implementovat automatický mechanismus, který uživatelům dopomůže k vyšší kvalitě příspěvků.

Webová aplikace vychází z předpokladu, že a-priori existuje nějaký přepis ke každé nahrávce, takže uživatelův příspěvek je vlastně korekcí. Každý příspěvek má formu nahrazení textového segmentu jiným. Jelikož přepisy jsou zarovnané s audiem na časové ose, víme také, jakému přesně úseku nahrávky daný text odpovídá.

Dále se vychází z existence akustického modelu pro nahrávky, viz kapitolu 5.

Díky těmto dvěma prvkům mohou provést nucené zarovnání textového úseku s audiem. V případě selhání zarovnání můžeme předpokládat, že úsek byl přepsán chybně, příspěvek odmítnout a dát tím uživateli zpětnou vazbu. Jelikož jednotlivé úseky odpovídají akustickému modelu v různé míře, dochází k falešně pozitivním i negativním vyhodnocením.

Falešně pozitivní případ (když systém přijme chybný přepis) představuje skutečný problém, protože chyba vstoupí do trénovacích dat. Falešně negativní případy mohou uživatelé často obejít tím, že správný, leč odmítnutý přepis, pošlou znova, rozdělený do kratších částí. Touto metodou by se pochopitelně mohlo také podařit vnutit systému nesprávný přepis. Nepředpokládám však na straně uživatelů zlou vůli.

Krom zachycení chybného přepisu slouží nucené zarovnání k přesné synchronizaci na časové ose. Tento prvek zcela chybí prakticky ve všech programech pro přepis, viz porovnání v podsekcí 6.1.1.

## 6.5.2 Pořízení fonetického přepisu

Fonetický přepis je nezbytný pro trénování akustického modelu. Pořizuje se provedením nuceného zarovnání na každý *ortograficky* manuálně přepsaný segment. Pokud je více výslovnostních variant, automaticky se zvolí ta, která lépe odpovídá akustickému modelu. Na to je potřeba pořídit výslovnostní varianty každého slova. Používám kombinaci pravidlového převodníku inspirovaného Psutkou et al. (2004)[97] a dynamického výslovnostního slovníku. Dynamický výslovnostní slovník je seznam alternativních výslovností každého slova, který se rozšiřuje s používáním aplikace.

Manuál k aplikaci vyzývá uživatele, aby text přepisovali podle standardního českého pravopisu, ale při zachování maximální věrnosti vyřčených slov, tedy aby nekorigovali /ɲa:k/ na *ňjak*, nýbrž přepsali doslova jako *ňák*. Fonetický slovník obsahuje časté výslovnostní varianty, např. počáteční /v/ ve slovech začínajících na /o/, tedy /vopitsi vobludnou/ mohou přepsat jako *opici obludnou*.

V případě slov s nestandardní výslovností, tedy primárně cizích slov, se od uživatelů žádá, aby slovo přepsali foneticky. Teprve po úspěšném zarovnání a integraci do přepisu mohou slovu nastavit kýženou pravopisnou formu. Toto je jeden z mála případů, kdy se od uživatele chce něco nekonvenčního.

Když je pravopisně chybný, fonetický zápis poslán, pak pokud projde fází nuceného zarovnání, integruje se do zobrazeného přepisu. Datová reprezentace každého slova sestává z těchto prvků:

1. Výskyt: slovo, jak se vyskytuje v textu, včetně zachování velkých a malých písmen a přílehlé interpunkce.
2. Slovní forma: slovo, jak je zaneseno v jazykovém modelu a ve výslovnostním slovníku. (Slovní forma se odvozuje algoritmicky z výskytu převedením do malých písmen a odstraněním neabecedních znaků. Z toho plyne, že interpunkce a všechny neabecední znaky jsou vždy součástí přílehlého slova a nikdy netvoří token samy o sobě.)
3. Výslovnost: posloupnost hlásek.

4. Časová značka: vzdálenost počátku slova od počátku nahrávky v sekundách s přesností na dvě desetinná místa.
5. (Délka: časová pozice od začátku slova, kde končí poslední vlastní hláska (tedy kde začíná ticho nebo následující slovo).
6. Manuálně přepsané: pravdivostní hodnota odlišující manuálně přepsaná slova od automaticky přepsaných.
7. Confidence measure<sup>16</sup>: míra jistoty, se kterou bylo slovo predikováno (týká se pouze automaticky přepsaných slov).
8. Ticho<sup>17</sup>: začátek a délka ticha, pokud se za slovem vyskytuje.

Jakmile je slovo součástí přepisu, lze upravit jeho *výskyt*, tedy jak se jeví v textu. Nyní může uživatel vložit správnou ortografickou formu odchyloující se od českých pravidel výslovnosti.

To má za následek přidání dvojice *slovní forma - výslovnost* do dynamického výslovnostního slovníku. Tento úkon je proto nutné provést jen jednou pro každé slovo. Pokaždé, když na toto slovo libovolný uživatel narazí znova, stačí zadat jeho ortografickou formu a správná výslovnost se dovodí automaticky.

Například při prvním setkání se jménem *George* je potřeba je přepsat jako *džordž*. Poté lze tomuto slovu změnit výskyt na *George*, čímž se do dynamického výslovnostního slovníku tento pár zápis - výskyt přidá. Všechny další výskyty tohoto jména se pak mohou přepisovat jako *George* a správnou výslovnost systém aplikuje automaticky.

### 6.5.3 Fonetický zápis

Přese všechny výhody reprezentace hlásek podle systému PACal se nejedná o praktický zápis výslovnosti pro laické Čechy. Díky jednoduchému, povětšinou deterministickému mapování mezi fonémy a grafémy je fonetický zápis, nebo jak se tento mechanismus označuje anglicky, *pronunciation respelling*, v češtině něčím přirozeným a spolehlivým. Není ani potřeba explicitního dělení slabik, jako tomu je u angličtiny (Wikipedie<sup>18</sup> udává příklad “*Diarrhoea*“ *is pronounced DYE-uh-REE-a*). Že tato technika je přirozenou pro všechny rodilé mluvčí češtiny se základním vzděláním, postulují jako fakt bez podpůrného výzkumu a zakládám to čistě na vlastní zkušenosti.

Převod z fonetického zápisu do PACal obstarává zmíněný převodník, viz podsekcí 6.5.2. Je ale zapotřebí i opačného směru, aby se uživateli mohla dát možnost zkontrolovat, zda slovo, které přepsal, se uložilo se správnou výslovností.

Za tímto účelem jsem vytvořil javascriptový modul pro převod mezi seznamem fonémů a českým fonetickým zápisem. Popsán je v článku Krůza (2018)[98].

Algoritmus je jednoduchý. Ve většině případů jeden foném odpovídá jednoznačně jednomu písmenu ve fonetickém přepisu. Výjimky jsou tyto:

1. Foném *x* se píše *ch*.

<sup>16</sup>Od přechodu na zarovnávaní založené na Montreal Forced Aligner se již nepoužívá.

<sup>17</sup>Zastaralý údaj nahrazený délkou slova.

<sup>18</sup>[https://en.wikipedia.org/wiki/Pronunciation\\_respelling](https://en.wikipedia.org/wiki/Pronunciation_respelling)



2. Fonémy *dz*, *dzh* se píší *dz*, *dž*.
3. Dvojhlasiky *aw*, *ew*, *ow* se píší *au*, *eu*, *ou*.
4. Sekvence *ch*, *ou*, *au*, *eu*, *dz*, *dzh* se píší *c'h*, *o'u*, *a'u*, *e'u*, *d'z*, *d'ž*. Budiž však poznamenáno, že sekvence *ch* je ryze hypotetická, ana porušuje spodobu znělosti.
5. Neznělou zvýšenou alveolární vibrantu označuji *ř'*.
6. Palatální a labiodentální nazála se píší *n'*, *m'*.
7. Ticho na konci slova ve fonetickém přepise nevyznačuji.

Modul umožňuje obousměrný převod, ačkoliv v aplikaci je zapotřebí jen směr ze seznamu fonémů do fonetického zápisu určeného pro člověka. Uživatel sice může explicitně vyznačit neznělé *eř* oproti znělému, či posloupnost hlásek *o*, *u* oproti dvojhlasce *ou* pomocí apostrofu. Za osm let provozu však tohoto nebylo ani jednou potřeba.

Podotýkám, že ve výstupu převodníku do fonetického zápisu se nikdy nevykytují sekvence *di*, *ti*, *ni*, *dě*, *tě*, *ně*. Palatální souhlásky jsou vždy vyjádřeny explicitně háčkem a např. sekvence *ni* se vždy vyjádří jako *ny*.

V tabulce 6.2 uvádím několik příkadů slov s jejich výslovností a fonetickým zápisem, jak jej produkuje algoritmus, pokud na vstup dostane příslušnou výslovnost ve formátu PACal.

slovo	výslovnost v IPA	výsl. v PACal	fonetický zápis
nic	nɪts	nj i c	ňic
kdo	gdɔ	g d o	gdo
disk	disk	d i s k	dysk
dřít	dʁit	d rzh ii t	dřít
třít	tʁit	t rsh ii t	třít
auto	awto	aw t o	auto
nauka	naʊka	n a u k a	na'uka
džbán	dʒba:n	dzh b aa n	džbán
odžít	odʒit	o d zh ii t	od'žít
odznak	odznak	o dz n a k	odznak
podzemí	podzemi:	p o d z e m ii	pod'zemí
noc	nots	n o c	noc
tento	tɛnto	t e n t o	tento
hangár	ɦaŋgar	h a ng g aa r	han'gár
samba	samba	s a m b a	samba
tonfa	tomfa	t o mg f a	tom'fa

Tabulka 6.2: Příklady algoritmicky získaného fonetického zápisu.

Použití apostrofu pro rozlišení víceznačností a zvláštností není stoprocentně intuitivní a představuje další bod, kde je zapotřebí uživatele zaškolit, aby tuto funkcionalitu dokázal patřičně využívat.

## 6.5.4 Vyhodnocení kvality přepisů

Jak stojí výše, webová aplikace krom jiného slouží pro získání kvalitního zarovnaného přepisu od laických uživatelů. Pokusím se vyhodnotit, do jaké míry se to podařilo.

Vyhodnocení kvality přepisů proběhlo ve zvláštním procesu, kde tři dobrovolníci vyhodnocovali náhodně přidělené záznamy, v nichž žádný příspěvek nebyl od nich samých. Úkolem auditorů bylo opravovat chyby v ruční transkripci. Zkontrolovalo se celkem 74 968 přepsaných slov celkem v 11 hodinách záznamu. Za jednu chybu se počítá chybějící, přebývající, zaměněné nebo přesmyknuté slovo, tedy „ale je“ místo „je ale“ jsem bral za jednu chybu. Celkové množství nalezených chyb bylo 291, tedy 0,39%. Je ovšem nutno podotknout, že někteří přispěvatelé dosahují chybovosti i 0,02%.

Další z věcí, které můžeme posoudit, jsou přijetí a odmítnutí příspěvků zarovnávačem. Celkem z 109640 pokusů o zarovnání jich 3419 bylo odmítnuto, což je 3,12%. Manuálně jsem prošel 20 náhodně vybraných odmítnutých pokusů a přišel jsem k těmto číslům:

- V 11 případech se jednalo o falešná negativa, u nichž přepis byl správný a měl být přijat,
- ve 4 případech byly příčinou odmítnutí akustické nedostatky jako např. šum,
- ve 4 případech se jednalo o pravdivá negativa způsobená chybně zvolenými hranicemi segmentu a
- v 1 případě se jednalo o pravdivé negativum způsobené chybným přepisem.

Ve 25% tohoto malého vzorku tedy zarovnávač splnil svoje validační poslání, předšed tomu, aby se do trénovacích dat dostal chybný vzorek. V 55% případů selhal a byl jen překážkou v práci a ve zbývajících 20% případů sice odmítl validní přepis, ale zabránil tomu, aby se do trénovacích dat dostal defektní vzorek, na což se dá dívat v pozitivním světle.

Dá se také vyhodnotit scénář s nestandardní výslovností. Za tím účelem jsem z dynamického výslovnostního slovníku vybral 4 nadějně záznamy a prohlédl si příspěvky, které je obsahují. Tabulka 6.3 uvádí pro každý z nich správnou ortografickou formu, chybnou výslovnost získanou převodníkem, správnou výslovnost a konečně možný fonetický zápis. Ke každému údaji je uvedeno, kolikrát se objevil v manuálně přepsaných datech.

Z tabulky 6.4 je patrné, že většina případů je správně jak po stránce fonetické, tak po stránce pravopisné. Pouze asi ve 13% případů je uchována pravopisně nesprávná forma. To připisuji tomu, že uživatelé, kteří jsou si této problematiky vědomi, většinou celý proces dokončí a formu upraví.

Na druhou stranu téměř třetina případů vykazuje ponechání chybné fonetické reprezentace. To představuje závažný problém alespoň z dvou úhlů pohledu: Dokazuje se tím, že nucené zarovnání selhává při zachycení zcela odlišné výslovnosti, a zároveň se touto cestou dostávají do trénovacích dat špatné vzorky.

Jednou z patrných příčin je, že dynamický slovník rozpoznává pouze exaktně shodná slova. V jednom souboru je například vidět, jak všechny výskyty slova

psaná forma	#	chybná výslovnost	#	správná počestělá výslovnost	#	fonetický zápis	#
Moody	2	moʔodi	0	mu:di	4	múdy, mûdy	2
Descartes	2	dɛst̪sartɛs	0	dɛka:rt	4	dekárt	2
Weinfurter	30	vɛmfɔrtɛr	13	vajnɔrtʁ	19	vajnɔrtʁ	2
Michelangelo	6	mixelaŋɟelo	2	mixelaŋdʒɛlo	4	mikelandželo	0

Tabulka 6.3: Příklady nestandardní výslovnosti v manuálních prepisech.

	foneticky správně	foneticky chybně
ortograficky správně	25	15
ortograficky chybně	6	0

Tabulka 6.4: Správnost fonetické a ortografické reprezentace cizích slov na základě tabulky 6.3.

*Weinfurter* mají výslovnost správně, zatímco ostatní formy, jako např. *Weinfurterovi*, chybně.

Krom toho jistě budou hrát roli neinformovanost a roztržitost uživatelů, což se jim dá mít těžko za zlé, vzhledem k tomu, jak náročná činnost na soustředění se od nich chce.

Případ nesprávné ortografické formy oproti tomu nepředstavuje tak závažný problém. Může sice ztížit vyhledávání, ale to lze provést na výslovnosti, už nyní manuálně, a v případě potřeby automatizovat.

Čtvrtá kombinace fonetického zápisu a špatné výslovnosti se pochopitelně nevyskytuje.

## 6.6 Backend

Popsaná webová aplikace, která je uživatelským rozhraním, spoléhá na aplikační rozhraní (*API*), odkud dostává aktuální prepisy, kam posílá příspěvky od uživatelů, a na hosting, odkud stahuje zvukové soubory.

### 6.6.1 API

Backendová aplikace má formu HTTP serveru s následujícími koncovými body.

#### 1. Odeslání manuálního prepisu segmentu

- cesta: `/subsubmit`
- metoda: `POST`
- parametry:
  - `trans` (řetězec): prepis jak jej vložil uživatel,

- `filestem` (řetězec): identifikátor nahrávky,
  - `start` (desetinné číslo): pozice začátku přepsaného segmentu od začátku nahrávky v sekundách,
  - `end` (desetinné číslo): pozice počátku přepsaného segmentu, ditto,
  - `subs`: seznam zarovnaných slov,
- odpověď při úspěchu:
 

```
{
  success: 1,
  filestem: řetězec,
  start: desetinné číslo,
  end: desetinné číslo,
  data: seznam zarovnaných slov
}
```
  - odpověď při selhání: { `message`: řetězec }

Seznam zarovnaných slov je v tomto formátu:

```
[
  {
    fonet: řetězec,
    wordform: řetězec,
    occurrence: řetězec,
    humanic: 1, (znamená, že je manuálně přepsané)
    timestamp: desetinné číslo, (pozice začátku slova)
    slen: desetinné číslo, (délka ticha, jen když > 0)
  },
  ...
]
```

## 2. Požadavek na seznam přepisů

- cesta: `/init`
- metoda: `GET`
- odpověď:
 

```
jsonp_init({ subversions => { id => verze, ... } })
```

Verze se inkrementuje při každém příspěvku. Slouží k tomu, aby se mohly cachovat přepisy, ale aby se cache nepoužila, pokud někdo přepis změnil.

## 3. Inicializace sezení

- cesta: `/req`
- metoda: `POST`
- parametry:
  - `username` (řetězec),
  - `session` (řetězec, nepovinný),

- odpověď: { status: "OK" }

Slouží k detekci začátku práce na prepisech pro účely sledování času potřebného k přepisům.

#### 4. Požadavek statistiky, z jaké části je která nahrávka přepsána

- cesta: /humpart
- metoda: GET
- odpověď:
 

```
{
  identifikátor:
    human: celé číslo - počet manuálně přepsaných slov
    total: celé číslo - celkový počet slov
  ...
}
```

Tento endpoint momentálně nová verze webového rozhraní nepoužívá, byl zamýšlen jako vodítko pro uživatele při výběru nahrávky pro přepis a pro navození soutěživého ducha.

#### 5. Změna atributů zarovnaného slova v přepisu

- cesta: /saveword
- metoda: POST
- parametry:
  - **wordform** (řetězec): slovo malými písmeny bez interpunkce,
  - **occurrence** (řetězec): slovo, jak se vyskytuje v textu,
  - **fonet** (řetězec): hlásky oddělené mezerou, zavržený parametr, používá se endpoint `subsubmit`,
  - **timestamp** (desetinné číslo): pozice začátku slova od začátku nahrávky v sekundách,
  - **stem** (řetězec): identifikátor nahrávky,
- odpověď: { success: 1 }

Editace slova v přepisu. Používá se, když slovu, které je foneticky správně přepsané a zarovnané, je potřeba změnit ortografickou formu, např. u cizích slov, doplnit interpunkci atp.

Veškerá komunikace je kódována v UTF-8. Koncový bod `subsubmit` pro přijetí (nebo odmítnutí) přepisu úseku nahrávky, provede na straně serveru nucené zarovnání přijatého přepisu s odpovídajícím úsekem audia. Aparát pro nucené zarovnání zahrnuje spustitelné binární soubory z balíčku HTK, akustický model a uložené nahrávky v parametrizovaném formátu MFCC. Na jaře 2021 jsem implementoval zarovnávaní modernějším nástrojem Montreal Forced Aligner a nasadil jako separátní webovou službu. Ta běží v rámci Lindatu a přijímá requesty metodou POST se dvěma parametry: audiem v libovolném formátu, který sox dokáže

dekódovat, a přepisem ve formě prostého textu. Odpovědí je zarovnání ve formátu TextGrid<sup>19</sup>. Od té doby `subsubmit` přijímá zarovnaný text jako parametr, zatímco dříve bylo zarovnání jeho hlavní úlohou. O převod z formátu TextGrid do seznamu zarovnaných slov se stará front-end.

## 6.6.2 Ukládání dat

API používá databázi PostgreSQL pro ukládání příspěvků, metadat k nim a nepravdělných výslovností. Ke každému příspěvku se ukládá

- samotný přepis,
- identifikátor nahrávky,
- časové rozmezí odpovídajícího úseku nahrávky,
- zda byl přepis přijat,
- datum a čas přispění,
- přezdívka autora, pokud ji vyplnil,
- identifikátor sezení,
- verze prohlížeče.

Každé uživatelské sezení má taktéž svůj záznam a ukládá se proň totéž, co pro příspěvek, krom příspěvku samotného. Dále se v databázi ukládají verze přepisů, které se inkrementují při každém příspěvku. Poslední věcí v databázi je fonetický slovník. Ten slouží ke sběru fonetických reprezentací slov s nestandardní výslovností, jejichž výslovnost a psanou formu poskytují uživatelé.

Přepisy se ukládají trojitě. Primárně na disku serveru v souborech ve formátu *JSONP*<sup>20</sup>. Tyto se při každé změně zálohují na externí cloudové úložiště. Do třetice se denně a na požádání exportují do HTML, které je přístupné z CDN. CDN slouží též k servírování samotných nahrávek.

## 6.7 Budoucí práce

Webovou aplikaci je potřeba dále zdokonalovat s ohledem na UX. Rád bych snížil potřebu zaškolení uživatele zřehledňováním ovládacích prvků a integrací vhodného systému nápovědy. Dále chci odstranit nutnost označovat úsek, aby ho člověk mohl opravit. Metody zvýšení použitelnosti webu lze čerpat např. z knihy Jana Řezáče *Web ostrý jako břitva*[99].

---

<sup>19</sup>[https://www.fon.hum.uva.nl/praat/manual/TextGrid\\_file\\_formats.html](https://www.fon.hum.uva.nl/praat/manual/TextGrid_file_formats.html)

<sup>20</sup>JSON with padding – JSON obalený do JavaScriptové funkce.

## 7. Vyhledávání

Možnost vyhledávat v nahrávkách byl pro mne jeden z hlavních cílů od začátku projektu. Se získáním přepisů nahrávek, byť kolísavé kvality, bylo možné vyhledávání realizovat. Fulltextové vyhledávání jsem implementoval pomocí nástroje Elastic.

Elastic je svobodný vyhledávač napsaný v Javě, který umožňuje fulltextové vyhledávání v dokumentech. Dokumenty se rozumí datové struktury, které se vyhledávači poskytnou ve formátu JSON. Elastic má mnoho funkcionalit, z nichž pro mne je klíčové rozhraní na základě HTTP naplňující konvence REST, automatický stemming, zvýrazňování nalezených pasáží a možnost vyhledávat v libovolných položkách dokumentu.

Aby bylo možné každý nalezený výsledek proměnit v odkaz na příslušnou pasáž v nahrávce, zvolil jsem za jednotlivé dokumenty nikoliv celé nahrávky, nýbrž věty.

Ke každému dokumentu se ukládá

- textová reprezentace,
- posloupnost hlásek,
- stupeň manuálního přepisu, tedy zda je přepis pořízen zcela automaticky, zcela manuálně nebo kombinací obého
- a také vektor confidence measure jednotlivých automaticky přepsaných slov.

Pro skloňování je použito pravidlového stemmingu, který je dodáván s distribucí Elastic. Momentálně je vyhledávač nainstalován na téměř serveru jako API a je dostupný z webové aplikace. Důležitým bodem budoucí práce je automatizace indexování manuálních oprav, jak přicházejí. Dále pak zakomponování automatického přepisu pořízeného bez použití jazykového modelu, jak se diskutuje v podsekcí 7.1.1.

### 7.1 Kvantitativní vyhodnocení

Do jaké míry lze současný přepis použít k vyhledávání v korpusu? To jsem se pokusil odhadnout jednak na testovací sadě a jednak namátkou.

Na testovací sadě jsem vyhodnocení provedl tak, že jsem vzal všechna podstatná jména vyskytující se v ní a vyhledal je v ručním i v automatickém přepisu. Za úspěch jsem považoval, byla-li daná *věta* mezi výsledky v obou případech.

Podstatná jména jsem převedl do regulárních výrazů reflektujících skloňování. Celkem byly 344. Vět bylo celkem 376 s 5511 tokeny. Každé slovo se samozřejmě mohlo vyskytnout v mnoha větách. Zásahů mezi větami v automatickém přepisu bylo 840, mezi větami v ručním 880 a úspěšných zásahů bylo 813, což znamená **precision 96,79% a recall 92,39%**.

### 7.1.1 Identifikace témat

Provedl jsem dva pokusy o automatickou identifikaci témat v mluveném korpusu, které jsou popsány v článku Krůza (2019)[100]

První z nich využívá toho, že pojmenované entity a některé další výrazy se běžně nevyskytují v řeči, pokud se nemluví právě o nich. Vybral jsem namátkou klíčová slova

1. Lazar,
2. Mithra, mithraismus,
3. Satan,
4. svatá Terezie,
5. pohádka.

Pohádku jsem vybral jakožto opakující se téma, které není vázané na pojmenovanou entitu, ačkoliv by bývalo šlo použít jméno Honza, které se u Makoňe vyskytuje obvykle právě ve spojitosti s pohádkou.

V přepisu získaném pomocí neuronových sítí trénovaných na manuálním přepisu Makoňova korpusu jsem vyhledal kmeny výše zmíněných slov. Výsledek hledání shrnuje tabulka 7.1. U výsledků jsem manuálně zkontroloval, je-li tématem skutečně hledaný výraz. Pokud bylo výsledků hledání více než 20, zkontroloval jsem toto u náhodných dvaceti výsledků.

téma	dotaz	výsledků v automat. přepisu	očekávaný počet výsl. v aut. přep.	výsledků v ručním přepisu	precision
Lazar	lazar.*	113	157	14	11/20
Mithra	mith?ra.*	0	0	0	n/a
Satan	satan.*	1659	1741	155	14/20
Sv. Tereza	terez.*	1906	1752	156	14/20
pohádka	pohádk.*	911	640	57	16/20
celková precision					68,75%

Tabulka 7.1: Výsledky textového vyhledávání v přepisu v prosinci 2020. Precision počítána pouze z automatických přepisů.

Precision 69% poukazuje na fakt, že kvalita přepisu umožňuje v korpusu rozumně vyhledávat. Navíc pouze polovina chyb byla způsobena špatným přepisem. Zbylou polovinu tvořily případy, kdy se dané slovo skutečně v mluvě vyskytlo, ale jen jako letmá zmínka.

Celkem 8,9% slov v korpusu bylo v době tohoto experimentu manuálně přepsáno. Sloupec „očekávaný počet výsledků v automatickém přepisu“ obsahuje interpolaci toho, kolik výsledků by mělo být obsaženo, pokud by frekvence byla stejná jako v manuálním přepisu. Je vidět, že očekávaný a nalezený počet výskytů se příliš neliší, což napovídá, že recall je není vyloženě žalostný.



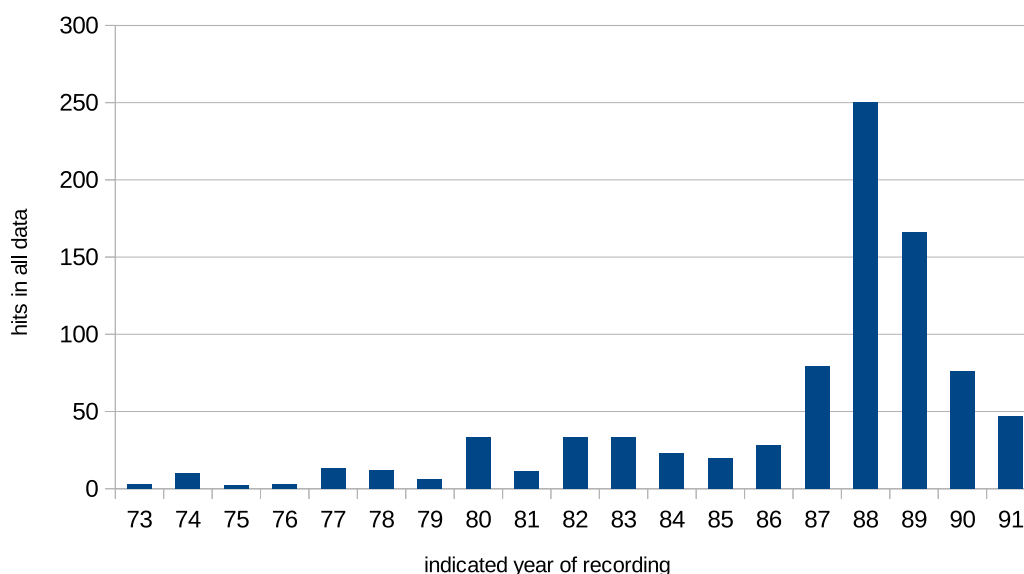
Jeden falešný výsledek hesla „*lazar*“ spočíval v použití tohoto slova pro označení nemocného člověka, tedy v přeneseném významu. Zde se tedy odrazila omezenost této metody pro hledání tématu, nikoliv nedokonalost přepisu. Při zkoumání výsledků vyhledávání hesla „*tereza*“ v podobě regulárního výrazu `tere*.*` jsem dospěl ke kurióznímu zjištění: Když nalezený výskyt byl formou slova „*Terezie*“, výsledek byl vždy správný, nenarazil jsem na žádnou výjimku. Ovšem byl-li výsledek „*Tereza*“, šlo ve více než polovině případů o chybně přepsaná slova „*které za*“. Forma „*Terezie*“ je v přepisu pětadvacetkrát častější, takže na výsledku tohoto experimentu se tato chyba odrazila jen nepatrně.

Díky tomu, že mithraismus se vyskutekuje v ručních indexech k nahrávkám, lze se přesvědčit, v čem tkví nulový výskyt tohoto hesla v přepisech. Automatický přepis zde skutečně selhává, vole slova jako „*mistrů*“ místo „*Mithrův*“ nebo „*my trasu*“ místo „*mithraismus*“. Pokusil jsem se několik frází obsahujících nějaké slovo odvozené od jména „*Mithra*“ rozpoznat bez použití jazykového modelu. Např. pro frázi „*se v jejich kultu mithraistickém znázorňovala*“ se bez jazykového modelu dojde k predikci „*te jejich kultu mitra jistickém znázorňovala*“.

Zdá se proto, že by bylo užitečné mít k dispozici pro vyhledávání i přepis získaný bez použití jazykového modelu. Toto je předmětem budoucí práce.

## 7.1.2 Korelace témat mezi nahrávkami a knihami

Pro vyhledávání se dá využít i psaných Makoňových děl. Věc se zakládá na domněnce, že v době, kdy Makoň o určitém tématu psal knihu, o něm bude pravděpodobně také mluvit na přednáškách. Pro ověření domněnky jsem se zaměřil na téma svaté Terezie z Avily. V roce 1988 Makoň překládal její dílo *Hrad nitra (El castillo interior)*. Vzhledem k tomu, že u většiny nahrávek známe rok jejich pořízení, lze zjistit distribuci výskytů vyhledávání podle roku. Pro dotaz na svatou Terezii tuto distribuci ukazuje obrázek 7.1.



Obrázek 7.1: Počet výsledků vyhledání dotazu `tere*.*` v přepisech podle udaného roku nahrávky.

Elevace kolem roku 1988 v grafu domněnku korelace mezi tématem právě psané knihy a tématem přednášek podporuje.

## 7.2 Případová studie

Vyhledávání v prepisech mluveného korpusu našlo využití v kompilaci referátů o určitých tématech, kterým se Karel Makoň věnuje. V průběhu let 2018 až 2020 vznikly alespoň čtyři takové, a to na témata

- karma,
- převtělování,
- Otčenáš,
- relativní dobro a zlo.

Každé téma bylo zpracováno do formy souboru krátkých úseků nahrávek, které se prezentovaly sekvenčním přehráním s prepisy jako vizuálním vodítkem.

Autor referátu o tématu relativního dobra a zla dohledal poznámkový aparát k tvorbě a rekonstruoval svůj postup, který zde popíšu jako příklad použití prepisů, z něhož lze usoudit na efektivitu práce.

Téma relativního dobra a zla bylo předem zamýšleno a bylo vybráno pro autorův zájem a nikoliv s ohledem na to, jak snadné bude pro vyhledání. Dopředu byl dán časový rámec výsledku na cca. dvě hodiny zvukového záznamu. Kvalitativními kritérii bylo 1) důsledné přidržení se tématu, 2) aby se téma zpracovalo z různých úhlů pohledu, s čímž korelují různé ročníky zahrnutých zdrojových nahrávek, 3) aby výsledná kompilace tvořila koherentní výpověď bez mnoha významově shodných vyjádření.

Autorova metodika byla následovná: vyhledal frázi „relativní dobro a zlo“ a výsledky procházel ve výchozím relevantním řazení nástroje Elastic. Prošel prvních sto z celkových 7379 výsledků. U každého posoudil, zda se jedná o pasáž, skutečně o tématu pojednávající, nebo jen o letmou zmínku či vůbec falešný zásah, popřípadě o duplicitní výskyt.

Po odstranění duplicit a jednoznačně irelevantních výsledků prošel autor výsledky opět po řadě, a vybíral z této užší množiny s ohledem na ročník zdrojové nahrávky, aby byl v kompilátu zastoupen průřez vývoje Makoňova myšlení, výjimečně podle návaznosti či shrnujícího charakteru výpovědi pro závěrečnou část kompilace.

Vznikla tak výsledná množina 25 úseků, které pocházely z prvních 53 výsledků vyhledávání. Do konečného se tak dostala zhruba polovina procházené množiny. V první stovce výsledků vyhledávání autor identifikoval 16 duplicit. Celkový čas strávený tvorbou kompilace autor odhaduje řádově na desítky hodin, v čemž je zahrnuta i ostatní práce, nejen vyhodnocování relevance výsledků.

## 8. Závěr

Tématem disertační práce je iterativní zdokonalování přepisu zvukových nahrávek s využitím zpětné vazby posluchačů. Hlavním předmětem snažení tedy bylo vytvoření systému, pomocí kterého se pro existující soubor záznamů opatří co nejdokonalejší přepis pomocí počítačnělingvistických metod a zapojení laické komunity.

Práce zasahuje do několika odvětví. Za prvé jde o digitalizaci a uchování fondu nahrávek a tím spadá do archivnictví. Za druhé jde o představení nového korpusu a tím spadá do korpusové lingvistiky. Za třetí jde o pořízení přepisu automatickými metodami a tím spadá do oblasti rozpoznávání řeči. Za čtvrté jde o vývoj nového typu aplikace a tím spadá do oblasti softwarového inženýrství. Krom toho se dotýká obsahu mluveného korpusu Karla Makoně, čímž se tento projekt dotýká i některých odvětví věd humanitních. Z hlediska obsahu textů se jedná o filosofii, teologii a religionistiku. Z hlediska téměř zapomenutého odkazu ing. Karla Makoně jde i o téma české historie.

### 8.1 Výsledky disertační práce

Zastřešujícím a hlavním výsledkem této práce je návrh, realizace a vyhodnocení metody pořízení kvalitního zarovnaného přepisu velkého množství dat se zapojením pouze malého počtu laických přispěvatelů. Taková metoda, pokud je mi známo, nebyla dosud realizována. Přitom ve své podstatě je univerzální a vhodná pro nasazení všude tam, kde 1) existuje soubor velkého množství nahrávek, u nichž je kýžený přepis např. pro vyhledávání a 2) je přítomna třeba i malá skupinka lidí, kteří projektu jsou ochotni věnovat čas.

Důležitým bodem je zde pořízení kompletního přepisu zkoumaného mluveného korpusu. Přepis umožňuje další zkoumání obsahu této ojedinělé sbírky, čímž poskytuje materiál pro bádání v různých oborech. Dosažení tohoto cíle nebylo snadné z důvodu specifické mluvy, slovní zásoby, akustické kvality a rozsahu materiálu.

Souvisejícím výsledkem je vyvinutí webové aplikace, která umožňuje synchronní konzumaci mluveného projevu a jeho přepisu a sběr oprav přepisu od laických uživatelů použitelných jako trénovací data pro strojové učení. Tato aplikace se odlišuje od všech mně známých jiných svým *use-case*, takže může sloužit jako prototyp pro daný scénář. Svým provedením spojuje potřebnou funkcionalitu s plynulostí a komfortem webové platformy, čímž skýtá měřítko pro budoucí obdobná řešení.

Vedlejším produktem, který však představuje jeden z nejvýznamnějších výsledků práce, je trénovací sada pro rozpoznávání češtiny nezávisle na mluvčím, konkrétně nový tisícihodinový korpus svobodně k dispozici všem badatelům. V porovnání s ostatními dostupnými korpusy představuje materiál pro trénink nejrobustnějšího akustického modelu, viz tabulku 5.4.

Dalšími výsledky jsou umožnění fulltextového vyhledávání v mluveném korpusu, částečné zmapování jeho obsahu a několik výsledků negativních: keprstránní normalizace na izolovaných řečových událostech, použití standardní CycleGAN pro snížení chybovosti rozpoznávání řeči a získávání specifických trénovacích dat

pro aktivní učení od dobrovolných anotátorů.

Korpus je k dispozici v repozitáři Lindat:

<https://lindat.mff.cuni.cz/repository/xmlui/handle/11372/LRT-1455>.

Webová aplikace sídlí na adrese <http://radio.makon.cz>.

Všechny zdrojové kódy jsou k dispozici na GitHubu [github.com/Sixtease](https://github.com/Sixtease).

Relevantní repozitáře:

- Evadevi skripty pro rozpoznávání řeči nezávislé na datech,
- `cz-parliament-speech-corpus` kompilace záznamů jednání parlamentu ČR pro trénování ASR,
- `Lingua-CS-Num2Words` rozpis čísel do číslovek (modul pro Perl),
- `MakonASR` automatický přepis Makoňova korpusu pomocí Evadevi / HTK,
- `DsMakonASR` automatický přepis Makoňova korpusu pomocí DeepSpeech,
- `MakonFM` backendová aplikace a prototyp front-endu,
- `MakonReact` front-endová aplikace,
- `SpeechBack` mikroslužba pro nucené zarovnávání,
- `CorpusMakoni` nástroje pro obsluhu dat a index k záznamům,
- `Disertace` tato disertační práce.

## 8.2 Budoucí práce

Během práce na tomto projektu se postupně odhalila další možná témata, která určitě stojí za povšimnutí. Při řešení úkolů, které si tato práce kladla za cíl, se ukázaly nové možnosti vylepšení celého přístupu. Budoucí témata jsem již zmínil již výše, tak jak mě napadala během práce. Zde jsou shrnuty v bodech:

- automatizovat indexaci manuálních oprav do vyhledávače,
- integrovat vysvětlivky, aby se dále redukovala potřeba zaškolení,
- umožnit editaci bez nutnosti předchozího označování.

Kromě toho bych rád zařadil integraci tematických anotací do webové aplikace. Stávající data tohoto druhu jsou k dispozici ve velkém množství, proto mohou být k užitku lidem, kteří hledají pasáže týkající se konkrétního tématu. Zároveň je možné využít zapojení uživatelů i jiným způsobem než pro sběr manuálních prepisů, a to tím spíše, že s rostoucím množstvím jejich přínos pro akustické a jazykové modelování klesá.

Pochopitelně bych také rád pokračoval v akustickém čištění poškozených záznamů, aby vzrostla přesnost jejich přepisu i srozumitelnost lidskému uchu.

Jistě bych uvítal, kdyby se mnou vyvinutá technologie mohla použít i na jiné sady nahrávek. Jsem přesvědčen, že by se přepis mluveného korpusu a jeho další aplikace daly využít například v historii či jiných humanitních vědách. Jakékoli

velké soubory audio nahrávek s komunitou příznivců by tak mohly být přepsány a dále podrobněji zpracovávány metou, kterou jsem v této práci představil. Ke zrealizování tohoto cíle v současnosti (léto 2021) pracuji na uzpůsobení front-endu k použití na libovolná data v rámci Lindatu.

## 8.3 Poděkování

*(bez zvláštního pořadí)*

Děkuji svému školiteli doc. RNDr. Vladislavu Kuboňovi, Ph.D. za vřelou záštitu a svobodu při práci, doc. RNDr. Markétě Lopatkové, Ph.D. za pomoc a podporu při účasti na konferencích, Mgr. Nino Peterkovi, Ph.D. za přípravu v akustickém modelování, Mgr. Davidu Klusáčkovi, Ph.D. za mnoho rad a pomoc při akustických úpravách, doc. RNDr. Ondřeji Bojarovi, Ph.D. a RNDr. Zdeňku Morávkovi, Ph.D. za nápady a konzultace, doc. Ing. Zdeňkovi Žabokrtskému, Ph.D. za významnou pomoc i upřímnou zpětnou vazbu a vážené paní Libuše Brdičkové a ing. Daniele Rejnušové za jejich neúnavnou ochotu a vstřícnost.

Děkuji váženému MUDr. Vítu Elgrovi, Ing. Milanu Tulachovi, Mgr. Lence Vinklerové, Alence Valentové, a dalším za zprostředkování díla Karla Makoně, za zapůjčení nahrávek, spolupráci, testování a používání aplikace i za zpětnou vazbu.

Děkuji Mgr. Petru Kazdovi z Konicy Minolty, že mi umožnil skloubit zaměstnání s prací na disertaci.

# Seznam použité literatury

- [1] Steve Crowdy. Spoken corpus design. *Literary and Linguistic Computing*, 8(4):259–265, 1993.
- [2] Jurik Hájek. Český mystik Karel Makoň. *Dingir*, 2007/4:142–143, 2007.
- [3] Radmila Müllerová. *Vyprávění o utrpení: Formování identity a důvěryhodnosti mystika Karla Makoně*. PhD thesis, Masarykova univerzita, Filozofická fakulta, 2019.
- [4] Swami Vivekananda. *Bhakti Yoga*. Celephaïs Press, Leeds, UK, 2003.
- [5] Sri Aurobindo. *The Synthesis of Yoga*. Sri Aurobindo Ashram Publication Department, 1999.
- [6] Teresa de Ávila. *El Castillo Interior*. San Pablo, 2011.
- [7] Adolphe-Alfred Tanquerey. *Précis de Théologie Ascétique et Mystique*. Société de Saint Jean l’Evangéliste, Desclée et Cie, Tournay - Alphonse Picard, Paris, 1928.
- [8] Karel Makoň. *Umění následovat Krista*. Psychoenergetická skupina ČSVTS, 1992.
- [9] Karel Makoň. *Odkrytá moudrost starých pravd*. Onyx, 1992.
- [10] Karel Makoň. *Utrpení a láska*. Psychoenergetická společnost Praha, 1995.
- [11] Karel Makoň. *Mystická koncentrace a příprava k ní*. Psychoenergetická společnost Praha, 1995.
- [12] Karel Makoň. *Světlo na cestu*. Česká psychoanalytická společnost, 1999.
- [13] Karel Makoň. *Blahoslavenství*. Onyx, 2000.
- [14] Karel Makoň. *Duchovní úlohy*. Onyx, 2002.
- [15] Karel Makoň. *Základní kurs nadživotnosti pro ty, kteří si myslí, že nevěří, a základní kurs náboženství pro ty, kteří si myslí, že věří, neboť jsou si všichni rovni, pokud umírají, aniž by se během života znovu narodili*. Onyx, 2005.
- [16] Lucie Skorkovská, Pavel Ircing, Aleš Pražák, and Jan Lehečka. Automatic topic identification for large scale language modeling data filtering. In *International Conference on Text, Speech and Dialogue*, pages 64–71. Springer, 2011.
- [17] Michael I Mandel and Daniel PW Ellis. Song-level features and support vector machines for music classification. 2005.
- [18] Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. Using mutual proximity to improve content-based audio similarity. In *ISMIR*, volume 11, pages 79–84, 2011.

- [19] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [20] B. W. Gillespie and L. E. Atlas. Acoustic diversity for improved speech recognition in reverberant environments. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–557–I–560, 2002.
- [21] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann. Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Processing Magazine*, 29(6):114–126, 2012.
- [22] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224, 2017.
- [23] M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402, 2013.
- [24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [25] Deniz Engin, Anil Genc, and Hazim Kemal Ekenel. Cycle-dehaze: Enhanced cyclegan for single image dehazing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- [26] Xiaohan Jin, Ye Qi, and Shangxuan Wu. Cyclegan face-off. *arXiv preprint arXiv:1712.03451*, 2017.
- [27] Heran Yang, Jian Sun, Aaron Carass, Can Zhao, Junghoon Lee, Zongben Xu, and Jerry Prince. Unpaired brain mr-to-ct synthesis using a structure-constrained cyclegan. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 174–182. Springer, 2018.
- [28] Takuhiro Kaneko and Hirokazu Kameoka. Parallel-data-free voice conversion using cycle-consistent adversarial networks. *arXiv preprint arXiv:1711.11293*, 2017.
- [29] Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation. *arXiv preprint arXiv:1804.00522*, 2018.
- [30] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

- [31] Matěj Korvas, Ondřej Plátek, Ondřej Dušek, Lukáš Žilka, and Filip Jurčíček. Free english and czech telephone speech corpus. 2014.
- [32] Ondřej Plátek, Ondřej Dušek, and Filip Jurčíček. Vystadial 2016 – czech data, 2016. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [33] Marie Mikulová, Jiří Mírovský, Anja Nedoluzhko, Petr Pajas, Jan Štěpánek, and Jan Hajič. Pdtsc 2.0-spoken corpus with rich multi-layer structural annotation. In *International Conference on Text, Speech, and Dialogue*, pages 129–137. Springer, 2017.
- [34] Jan Hajič, Petr Pajas, Pavel Ircing, Jan Romportl, Nino Peterek, Miroslav Spousta, Marie Mikulová, Martin Grüber, and Milan Legát. Prague DaTabase of spoken czech 1.0, 2017. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [35] Martin Grüber. Czech senior COMPANION expressive speech corpus, 2014. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [36] Luboš Šmídl and Aleš Pražák. OVM – otázky václava moravce, 2013. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [37] Luboš Šmídl, Petr Stanislav, and Vlasta Radová. STAZKA – speech recordings from vehicles, 2015. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [38] Aleš Pražák and Luboš Šmídl. Czech parliament meetings, 2012. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [39] Jan Oldřich Krůza. Czech parliament meeting recordings as asr training data. In Maria Ganzha, Leszek Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*, volume 21 of *Annals of Computer Science and Information Systems*, pages 185–188. IEEE, 2020.
- [40] Jonas Kratochvil, Peter Polak, and Ondřej Bojar. Large corpus of czech parliament plenary hearings. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6363–6367, 2020.
- [41] Matyáš Kopp, Vladislav Stankov, Jan Krůza, Pavel Straňák, and Ondřej Bojar. Parczech 3.0: A large czech speech corpus with rich metadata. *Lecture Notes in Computer Science*, 2021.



- [42] Pedro J Moreno, Chris Joerg, Jean-Manuel Van Thong, and Oren Glickman. A recursive algorithm for the forced alignment of very long audio segments. In *Fifth International Conference on Spoken Language Processing*, 1998.
- [43] Timothy J Hazen. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Ninth International Conference on Spoken Language Processing*, 2006.
- [44] A Nagórski, Lou Boves, and Herman Steeneken. In search of optimal data selection for training of automatic speech recognition systems. In *2003 IEEE workshop on automatic speech recognition and understanding (IEEE Cat. No. 03EX721)*, pages 67–72. IEEE, 2003.
- [45] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.
- [46] Balázs Csanád Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Eötvös Loránd University, Hungary*, 24(48):7, 2001.
- [47] Ken H Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.
- [48] Xuedong Huang, James Baker, and Raj Reddy. A historical perspective of speech recognition. *Communications of the ACM*, 57(1):94–103, 2014.
- [49] H Sakoe and Chiba S. A dynamic programming approach to continuous speech recognition. In *Proceedings of International Congress on Acoustics Budapest, Hungary*, pages Paper 20C–13, 1971.
- [50] Alan V Oppenheim. Speech analysis-synthesis system based on homomorphic filtering. *The Journal of the Acoustical Society of America*, 45(2):458–465, 1969.
- [51] Stanley Smith Stevens, John Volkman, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.
- [52] Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.
- [53] Pavel Ircing, Pavel Krbec, Jan Hajic, Josef Psutka, Sanjeev Khudanpur, Frederick Jelinek, and William Byrne. On large vocabulary continuous speech recognition of highly inflectional language-czech. In *Seventh European Conference on Speech Communication and Technology*, 2001.
- [54] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.
- [55] Josef Psutka, Pavel Ircing, Josef V Psutka, Vlasta Radová, William J Byrne, Jan Hajic, Jiri Mirovsky, and Samuel Gustman. Large vocabulary asr for spontaneous czech in the malach project. In *Eighth European Conference on Speech Communication and Technology*, 2003.

- [56] Josef Psutka, Pavel Ircing, Josef V Psutka, Vlasta Radová, William J Byrne, Jan Hajič, Samuel Gustman, and Bhuvana Ramabhadran. Automatic transcription of czech language oral history in the malach project: Resources and initial experiments. In *International Conference on Text, Speech and Dialogue*, pages 253–260. Springer, 2002.
- [57] Josef Psutka, Pavel Ircing, Josef Psutka, Jan Hajič, William Byrne, and Jiří Mírovský. Automatic transcription of czech, russian, and slovak spontaneous speech in the MALACH project. In *Proceedings of Eurospeech 2005*, pages 1349–1352, Lisboa, Portugal, 2005. ISCA, ISCA.
- [58] Steve Renals, Nelson Morgan, Hervé Bourlard, Michael Cohen, and Horacio Franco. Connectionist probability estimators in hmm speech recognition. *IEEE transactions on speech and audio processing*, 2(1):161–174, 1994.
- [59] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.
- [60] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.
- [61] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [62] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.
- [63] Z Palková. Fonetika a fonologie Češtiny univerzita karlova, 1992.
- [64] Jan Nouza, Josef Psutka, and Jan Uhlír. Phonetic alphabet for speech recognition of czech. *Radioengineering*, 6(4):16–20, 1997.
- [65] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [66] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius—an open source real-time large vocabulary recognition engine. 2001.
- [67] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.
- [68] Fred Jelinek. Self-organized language modeling for speech recognition. *Readings in speech recognition*, pages 450–506, 1990.

- [69] Loïc Barrault, Ondřej Bojar, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, 2019.
- [70] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics, 2011.
- [71] Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.
- [72] Olli Viikki and Kari Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1-3):133–147, 1998.
- [73] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.
- [74] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [75] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.
- [76] Nelson Morgan and Herve A Bourlard. Neural networks for statistical recognition of continuous speech. *Proceedings of the IEEE*, 83(5):742–772, 1995.
- [77] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [78] Lucie Benešová, Michal Křen, and Martina Waclawičová. Korpus spontánní mluvené češtiny oral2013. *Časopis pro moderní filologii (Journal for Modern Philology)*, 1(97):42–50, 2015.
- [79] Petr Mizera, Jiří Fiala, Aleš Brich, and Petr Pollak. Kaldi recipes for the czech speech recognition under various conditions. In *International Conference on Text, Speech, and Dialogue*, pages 391–399. Springer, 2016.
- [80] Nancy Ide, Christiane Fellbaum, Collin Baker, and Rebecca Passonneau. The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 conference short papers*, pages 68–73. Association for Computational Linguistics, 2010.

- [81] Samuel Reese, Gemma Boleda, Montse Cuadros, and German Rigau. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus. 2010.
- [82] Ondřej Bojar, Miroslav Janíček, Pavel Češka, Peter Beňa, et al. Czeg 0.7: Parallel corpus with community-supplied translations. *LREC 2008*, 2008.
- [83] M. Marge, S. Banerjee, and A. I. Rudnicky. Using the amazon mechanical turk for transcription of spoken language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5270–5273, March 2010.
- [84] Rada Mihalcea and Timothy Chklovski. Building sense tagged corpora with volunteer contributions over the web. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 260:357, 2004.
- [85] Oldřich Krůza and Nino Peterek. Making community and asr join forces in web environment. In *International Conference on Text, Speech and Dialogue*, pages 415–421. Springer, 2012.
- [86] Oldřich Krůza and Vladislav Kuboň. Second-generation web interface to correcting ASR output. In Kohei Arai, Rahul Bhatia, and Supriya Kapoor, editors, *Proceedings of the Future Technologies Conference (FTC) 2018*, number 1, pages 749–762, Cham, Switzerland, 2018. Science and Information Organization, Springer-Verlag.
- [87] D Abramov. Redux. *React Community, c*, 2015.
- [88] Paul Adenot, Chris Wilson, and Chris Rogers. Web audio api. *W3C, October*, 10, 2013.
- [89] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kald. In *Interspeech*, volume 2017, pages 498–502, 2017.
- [90] Kyle Gorman, Jonathan Howell, and Michael Wagner. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193, 2011.
- [91] Ingrid Rosenfelder, Josef Fruehwald, Keelan Evanini, and Jiahong Yuan. Fave (forced alignment and vowel extraction) program suite. *URL <http://fave.ling.upenn.edu>*, 2011.
- [92] Anthony Tseng. Progress bars vs. spinners: When to use which, 2016. Accessed: 2019-01-09.
- [93] Jakob Nielsen. Website response times, 2010. Accessed: 2019-01-09.
- [94] Randi Reppen. Building a corpus: what are the key considerations? In *The Routledge handbook of corpus linguistics*, pages 59–65. Routledge, 2010.
- [95] Jan Hajič. Complex corpus annotation: The prague dependency treebank. *Insight into Slovak and Czech Corpus Linguistics. Veda Bratislava*, 2005:54–73, 2005.

- [96] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara. Spontaneous speech corpus of japanese. In *LREC*. Citeseer, 2000.
- [97] Josef Psutka, Jan Hajic, and William Byrne. The development of asr for slavic languages in the malach project. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–749. IEEE, 2004.
- [98] Oldřich Krůza. Phonetic transcription by untrained annotators. In Stanislav Krajčí, editor, *Proceedings of the 18th conference ITAT 2018: Slovenskočeský NLP workshop (SloNLP 2018)*, volume 2203 of *CEUR Workshop Proceedings*, pages 35–40, Košice, Slovakia, 2018. Šafárik University, Košice, CreateSpace Independent Publishing Platform.
- [99] Jan Řezáč. *Web ostrý jako břitva: návrh fungujícího webu pro webdesignery a zadavatele projektů*. House of Řezáč, Brno, second edition, 2016.
- [100] Jan Oldřich Krůza. Spoken corpus of karel makoň. In *Book of Abstracts XI International Conference on Corpus Linguistics*, pages 189–190. ADEIT - Fundación Universidad-Empresa de la Universitat de València, 2019.

# Seznam obrázků

1.1	Schéma architektury systému. . . . .	6
2.1	Index přiložený k jednomu z kotoučů. . . . .	14
2.2	Ručně psaný index. . . . .	15
2.3	Přibývání celkového času oprav v hodinách. . . . .	18
2.4	Distribuce automaticky (bílá) a manuálně (černá) pořízených přepisů. . . . .	19
3.1	Kvalitní záznam bez zjevných defektů. <a href="http://radio.makon.cz/zaznam/90-02A#ts=673.14">http://radio.makon.cz/zaznam/90-02A#ts=673.14</a> . . . . .	21
3.2	Výrazné echo. <a href="http://radio.makon.cz/zaznam/90-24A-24.4.90#ts=664.33">http://radio.makon.cz/zaznam/90-24A-24.4.90#ts=664.33</a> . . . . .	22
3.3	Širokopásmový šum. <a href="http://radio.makon.cz/zaznam/92-04A#ts=691.37">http://radio.makon.cz/zaznam/92-04A#ts=691.37</a> . . . . .	23
3.4	Úzkopásmový šum. <a href="http://radio.makon.cz/zaznam/92-03B#ts=664.43">http://radio.makon.cz/zaznam/92-03B#ts=664.43</a> . . . . .	24
3.5	Absence vysokých frekvencí. <a href="http://radio.makon.cz/zaznam/88-04A#ts=678.94">http://radio.makon.cz/zaznam/88-04A#ts=678.94</a> . . . . .	25
3.6	Zrychlený záznam způsobený zpomalením převíjení pásky při nahrávání. <a href="http://radio.makon.cz/zaznam/90-18A-XX-zrychlene#ts=2473.56">http://radio.makon.cz/zaznam/90-18A-XX-zrychlene#ts=2473.56</a> . . . . .	26
3.7	Silně degradovaná nahrávka pořízená rychlostí 2,38 cm/s. <a href="http://radio.makon.cz/zaznam/kotouc-plzen-neident01-a#ts=660">http://radio.makon.cz/zaznam/kotouc-plzen-neident01-a#ts=660</a> . . . . .	27
3.8	Pomalá mluva. <a href="http://radio.makon.cz/zaznam/76-04A-Kaly-7-IEQUA#ts=13.79">http://radio.makon.cz/zaznam/76-04A-Kaly-7-IEQUA#ts=13.79</a> . . . . .	28
3.9	Rychlá mluva. <a href="http://radio.makon.cz/zaznam/89-11B#ts=203.17">http://radio.makon.cz/zaznam/89-11B#ts=203.17</a> . . . . .	29
3.10	Velikosti clusterů během hierarchického shlukování. . . . .	31
3.11	Průběh signálu (nahore) a spektrogram (dole) nahrávky pořízené rychlostí 2,38 cm/s před doménovým transferem (vlevo) a po něm (vpravo). . . . .	34
3.12	Průběh signálu (nahore) a spektrogram (dole) přebuzené nahrávky před doménovým transferem (vlevo) a po něm (vpravo). . . . .	34
4.1	Apriorní zarovnání a překryv zvukových záznamů k přepisům. Vyobrazen je záznam z 12. února 2020 kolem 10. hodiny. Přepis záznamu vlevo nahore pokrývá pozice od 01:34 do 11:24. Vpravo dole pak od 01:24 do 12:00. . . . .	36
4.2	Schéma zarovnání zvukových záznamů ke stenografickým přepisům na úrovni slov. . . . .	37
5.1	Ilustrace rozpoznávání řeči pomocí šablon technikou dynamic time warp. . . . .	41
5.2	Vzorkování signálu. $t$ je čas, $S(t)$ je průběh signálu, $T$ je vzorkovací perioda, $S(i)$ je hodnota $i$ -tého vzorku. . . . .	42

5.3	Schéma překrývajících se oken při převodu z vlnového průběhu do melfrekvenčních kepstrálních koeficientů. $A$ je šířka okna, $B$ je rozestup mezi okny. . . . .	43
5.4	Schéma aplikace filtrů do pásem podle stupnice mel. . . . .	44
5.5	Schéma použití markovovského řetězce jako generativního modelu v rozpoznávání řeči. . . . .	45
5.6	Schéma rekurentní neuronové sítě. Vertikálně je znázorněn průběh času a horizontálně jednotlivé vrstvy sítě. . . . .	49
5.7	Schéma buňky LSTM. . . . .	50
5.8	Schéma obousměrné rekurentní neuronové sítě . . . . .	50
5.9	Architektura systému DeepSpeech. . . . .	51
5.10	Hledání optimální podmnožiny obecného korpusu pro jazykové modelování. Na ose X je počet přidaných vět, na ose Y WER. Žlutá čára s trojúhelníčky reprezentuje první pokus (řádky 2-4 v tabulce 5.3); Červená čára s čtverečky na hrotech druhý pokus (řádky 5-8); modrá čára s čtverečky na stranách třetí pokus (řádky 9-16). . . . .	58
5.11	Vývoj úspěšnosti přepisu. . . . .	65
6.1	První verze webové aplikace. . . . .	69
6.2	Webové rozhraní při přehrávání. . . . .	70
6.3	Rozhraní ve stavu editace segmentu. . . . .	72
6.4	Uživatelské rozhraní Transcriberu. . . . .	73
7.1	Počet výsledků vyhledání dotazu <code>terez.*</code> v přepisech podle udaného roku nahrávky. . . . .	93

# Seznam tabulek

3.1	Word error rate při trénovacích i testovacích datech před redukcí šumu a po ní. . . . .	32
3.2	Word error rate u dvou skupin poškozených nahrávek před doménovým transferem a po něm. . . . .	33
5.1	Použité hlásky: IPA, PACal a nejčastější odpovídající grafém. . .	53
5.2	Použité záměny hlásek; hvězdičkou jsou vyznačeny záměny použité ještě v době psaní textu. . . . .	53
5.3	Úspěšnost rozpoznávání s použitím různých částí obecného korpusu. Kritérium rozhoduje o zařazení věty do jazykového modelu. Proměnná $m$ je pravděpodobnost věty podle unigramového modelu z Makoňových přepisů a spisů, vážená počtem slov. Proměnná $w$ je totéž podle modelu z obecných českých textů. Počet přidanych vět je uveden v celkovém počtu, v procentech celkové velikosti obecného korpusu a v násobcích velikosti reprezentativního korpusu. .	58
5.4	Word error rate rozpoznávání řeči na jednotlivých korpusech a na jejich konkatenaci. . . . .	62
5.5	Chybovosti tří důležitých modelů na různých testovacích sadách s trigramovým jazykovým modelem trénovaným na Makoňových přepisech a spisech. . . . .	64
6.1	Počet bodů předělu podle metody jejich získání. . . . .	79
6.2	Příklady algoritmicky získaného fonetického zápisu. . . . .	85
6.3	Příklady nestandardní výslovnosti v manuálních přepisech. . . . .	87
6.4	Správnost fonetické a ortografické reprezentace cizích slov na základě tabulky 6.3. . . . .	87
7.1	Výsledky textového vyhledávání v přepisu v prosinci 2020. Precision počítána pouze z automatických přepisů. . . . .	92



# Seznam publikací

1. Oldřich Krůza and Nino Peterek. Making Community and ASR Join Forces in Web Environment. In *International Conference on Text, Speech and Dialogue*, pages 415–421. Springer, 2012.
2. Oldřich Krůza and Vladislav Kuboň. Second-Generation Web Interface to Correcting ASR Output. In Kohei Arai, Rahul Bhatia and Supriya Kapoor, editors, *Proceedings of the Future Technologies Conference (FTC) 2018*, number 1, pages 749–762, Cham, Switzerland, 2018. Science and Information Organization, Springer-Verlag.
3. Oldřich Krůza. Phonetic Transcription by Untrained Annotators. In Stanislav Krajčí, editor, *Proceedings of the 18th conference ITAT 2018: Slovenskočeský NLP workshop (SloNLP 2018)*, volume 2203 of *CEUR Workshop Proceedings*, pages 35–40, Košice, Slovakia, 2018. Šafárik University, Košice, CreateSpace Independent Publishing Platform.
4. Jan Oldřich Krůza. Spoken Corpus of Karel Makoň. In *Book of Abstracts XI International Conference on Corpus Linguistics*, pages 189–190. ADEIT - Fundació Universidad-Empresa de la Universitat de València, 2019. [https://adeit-estaticos.econgres.es/19\\_CILC/book\\_abstracts.pdf](https://adeit-estaticos.econgres.es/19_CILC/book_abstracts.pdf)
5. Jan Oldřich Krůza. Czech parliament meeting recordings as ASR training data. In *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 185–188. IEEE. [https://annals-csis.org/Volume\\_21/drp/119.html](https://annals-csis.org/Volume_21/drp/119.html)
6. Matyáš Kopp and Vladislav Stankov and Jan Oldřich Krůza and Pavel Straňák and Ondřej Bojar. ParCzech 3.0: A Large Czech Speech Corpus with Rich Metadata. In *International Conference on Text, Speech and Dialogue*, Springer, 2021.