Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Automatická detekce a vizualizace chyb konkatenační syntézy řeči

PLZEŇ, 2013 **JAKUB VÍT**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: **2012/2013**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jakub VÍT**
Osobní číslo:	**A11N0076P**
Studijní program:	**N3918 Aplikované vědy a informatika**
Studijní obor:	**Kybernetika a řídicí technika**
Název tématu:	**Automatická detekce a vizualizace chyb konkatenační syntézy řeči**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování:

1. Seznamte se s problematikou konkatenační syntézy řeči, zejména pak se systémem syntézy řeči z textu ARTIC vyvíjeným na KKY.
2. Systémem ARTIC vygenerujte dostatečně reprezentativní množinu syntetizovaných promluv. Promluvy analyzujte, nalezněte v nich slyšitelné artefakty a vizualizujte je.
3. Navrhněte metodiku pro označování a kategorizaci slyšitelných artefaktů v syntetické řeči (např. špatná segmentace řečové jednotky ve zdrojové promluvě, nespojitost v průběhu základního hlasivkového tónu, neadekvátní trvání či změna tempa řečových jednotek, nespojitosti ve spektru, apod.).
4. Navrhněte algoritmus pro automatickou detekci artefaktů v syntetické řeči. Na základě analýzy syntetických promluv a metodiky kategorizace artefaktů navrhněte vhodná pravidla, popř. příznaky pro automatickou klasifikaci artefaktů.
5. Navržený algoritmus vyhodnoťte a získané výsledky vhodně vizualizujte.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 23. května 2013

.....
vlastnoruční podpis

Poděkování

Chtěl bych poděkovat Doc. Ing. Jindřichu Matouškovi, Ph.D., vedoucímu mé diplomové práce, za jeho odborné vedení, cenné rady a podporu.

Abstrakt

Syntéza řeči je již v dnešní době schopna vytvářet syntetickou řeč vysoké kvality. Konkatenační metoda s výběrem jednotek je známa právě pro svou schopnost vytvářet kvalitní syntetickou řeč, která je velmi přirozená. Nevýhodou této metody je fakt, že špatné napojení jednotek může vést k náhlému propadu kvality řeči, což je velmi rušivé. V této práci je představen návrh automatického systému, který by tyto jevy dokázal automaticky detekovat. Nejdříve je představen program, který dokáže analyzovat syntetickou řeč. Poté jsou na základě poslechových testů sestavena objektivní označení rušivých úseků. Z nich je trénován SVM klasifikátor. Ten je poté schopen označovat podobná místa v nových syntetických promluvách s velkou úspěšností. V provedených experimentech se úspěšnost pohybovala v rozmezí 70 až 80 %.

Klíčová slova: syntéza řeči, metoda výběru jednotek, detekce chyb, klasifikace

Nowadays, speech synthesis is able to produce high-quality synthetic speech. Unit selection method produces very natural speech, but it may suffer from sudden quality drops at concatenation points, which is very disturbing. In this thesis, an automatic speech synthesis error detection system is presented. Firstly, a program for speech analysis is introduced. Then, based on data gathered during listening tests, an SVM is trained. Using this classifier, errors can be labeled automaticaly in any new synthetic speech. Finnally, few experiments were carried out with a success rate between 70-80 %.

Keywords: speech synthesis, unit selection, error detection, classification

Obsah

Kapitola 1

Úvod

Výzkum a vývoj syntézy řeči probíhá již velmi dlouho. Za tu dobu se podařilo posunout kvalitu syntetické řeči o velký kus dopředu. V počátcích zněla syntetická řeč velmi uměle, dnes však již syntéza řeči dosahuje velmi vysoké kvality a srozumitelnosti.

Syntéza řeči je proces umělého vytváření řeči. Jejím cílem je vytvořit syntetickou řeč, která bude znít nejlépe identicky s tou lidskou. Řečový signál je ale velmi pestrý a komplikovaný, a tak se přes všechno úsilí o docílení co nejlepší kvality, občas vyskytnou v syntéze lokální chyby, které působí velmi rušivě a často znehodnotí dojem z celé věty. Pokud se jedná o lokální problém, hovoří se o tzv. „artefaktu“.

I když za poslední desetiletí došlo k výraznému zlepšení kvality syntetické řeči, problém vzniku artefaktů v syntetických promluvách, ač v menší míře, stále přetrvává. Právě na odhalování řečových artefaktů v syntetické řeči je zaměřena tato diplomová práce.

1.1 Cíle práce

Hlavní motivací řešení této úlohy je zlepšit kvalitu syntetické řeči produkované systémem syntézy řeči. Právě již zmíněný výskyt lokálních nespojitostí (artefaktů) je jednou z hlavních příčin propadu kvality syntetické řeči. Primárním cílem práce je tedy navrhnout automatický systém detekce řečových artefaktů. S použitím systému by bylo možné označit artefakty nejen v syntetické promluvě, ale rovněž by bylo možné těmto artefaktům předcházet. Systém detekce chyb by měl automaticky odhalit artefakt v syntetické řeči. K tomu by měl použít dostupné parametry ze systému syntézy řeči či jiné parametry, které budou snadno dostupné.

Obvyklý postup v úlohách zpracování řečového signálu je provedení poslechových testů. Vnímání lidské řeči je velmi subjektivní záležitost. Proto se reakce na syntetickou řeč liší. Poslechových testů se účastní více posluchačů právě proto, aby byla data reprezentativní a objektivní. Obyčejně se poslechové testy používají pro porovnání kvality syntetické řeči z původního a vylepšeného systému syntézy. V této práci byly poslechové testy využity pro jiné účely. Pro sestavení systému detekce řečových artefaktů je nutné mít k dispozici referenční data, která by reprezentovala množinu míst, ve kterých se vyskytuje řečový artefakt. Tato data však nejsou k dispozici. Vnímání chyb v syntetické řeči je také velmi subjektivní záležitost. Poslechový test byl v této úloze použit právě pro sestavení takovýchto referenčních

dat. Jeho sestavení je tedy druhý cíl práce. Získané „objektivní“[1] označení míst, kde se vyskytují artefakty, sloužilo pro trénování i zhodnocení systému automatické detekce řečových artefaktů.

Pro pochopení příčin vzniku artefaktu je třeba analyzovat velké množství syntetických promluv. V nich je třeba studovat průběh řečového signálu a také průběhy ostatních parametrů a spektra. Na všechny tyto funkce existují programy nebo jiné nástroje. Neexistuje však žádný program, který by všechny tyto funkce dokázal sjednotit a napojit na systém ARTIC[2]. Třetí cíl práce je vytvoření programu, který umožňuje vizualizovat a analyzovat proces syntézy řeči.

1.2 Obsah práce

V druhé kapitole je čtenář seznámen se základními pojmy, které souvisí s danou problematikou. Jedním z nich je téma syntézy řeči s důrazem na konkatenační metodu s výběrem jednotek. Na tu je totiž systém detekce artefaktů primárně cílen. Zbývající část kapitoly se soustřeďuje na problém klasifikace, konkrétně na SVM klasifikátor, který je v této práci využíván.

Třetí kapitola popisuje program pro analýzu syntetické řeči. Ten byl vyvinut pro vizualizaci syntetických promluv v okolí míst artefaktů. Později byl rozšířen i o další funkce, jako je například úprava segmentace řečového korpusu nebo napojení na systém ARTIC. Jeho vývoji se věnuje třetí kapitola.

Čtvrtá kapitola provádí analýzu problému vzniku artefaktů. K tomu hojně využívá program vyvinutý ve třetí kapitole. Příčiny vzniku artefaktů jsou kategorizovány a popsány. Dále je zde představen návrh systému pro automatickou detekci artefaktů. Je zde také objasněn důvod provedení poslechových testů.

Pátá kapitola se věnuje návrhu a realizaci získání referenčních dat pomocí poslechového testu. Je zde popsaná metodika předkládání vět posluchačům. Také je zde popsán způsob extrakce a selekce dat pro další použití v klasifikaci. U výběru dat je zmíněn způsob nastavení vah, které jsou také využity pro trénování klasifikátoru.

Šestá kapitola zakončuje práci samotným procesem trénování klasifikátoru, který tvoří základ systému pro automatickou detekci artefaktů. V této kapitole jsou představeny a diskutovány příznaky, podle kterých klasifikátor určuje, zda a na jakém místě syntetické promluvy se vyskytuje artefakt. Je zde rovněž popsán proces samotného trénování a přípravy systému. Závěrem kapitoly jsou prezentovány výsledky úspěšnosti klasifikace a také výsledky několika experimentů. Výsledky rovněž porovnávají přínos vážení příznaků během procesu klasifikace.

Poslední kapitola rekapituluje cíle, provedení a výsledky práce. Je zde proveden rozbor výsledků a analýza splnění zadání. V této kapitole jsou diskutována možná rozšíření a návrhy budoucí práce, které by mohly problém automatické detekce příznaků dále rozvíjet.

[1] Jako téměř objektivní lze považovat místa v syntetické řeči, kde více posluchačů označilo chybu.

[2] ARTIC je systém syntézy řeči vyvíjený na Katedře kybernetiky fakulty aplikovaných věd Západočeské univerzity v Plzni [6].

Kapitola 2

Základní pojmy

2.1 Syntéza řeči

Syntéza řeči je jednou z oblastí zpracování řečového signálu. Jedná se o proces, při němž se vytváří umělá řeč. Dnes se tak děje výhradně pomocí číslicového počítače. Cílem je vytvořit řeč co nejpřirozenější a nejsrozumitelnější. V dnešní době již srozumitelnost bývá považována za dostatečnou, a tak se hlavní proud výzkumu udává směrem co největší přirozenosti umělé řeči. Syntéza řeči tak může například produkovat emotivně zabarvenou řeč.

Metod syntézy řeči existuje několik. Každá metoda má své výhody i nevýhody, proto zatím neexistuje jeden nejlepší způsob syntézy řeči. Mezi nejpoužívanější se nicméně řadí *HMM syntéza* a konkatenační metoda s výběrem jednotek (*unit selection*).

V následující části jsou stručně shrnuty metody syntézy řeči a také problémy, které souvisí obecně s převodem psaného textu na řeč. Základní úlohou syntézy řeči bývá převedení vstupního textu na akustický signál řeči. Tento postup bývá označován jako *text to speech* (TTS). V této úloze se musí, kromě samotné syntézy řeči, řešit další dílčí problémy.

2.1.1 Normalizace textu

V první řadě je nutné text převést do psané formy. Zpracování textu, jinak tzv. *normalizace textu*, má za úkol nahradit ve vstupním textu elementy, které mají jinou formu zápisu psaní a výslovnosti. Jedná se například o číslovky psané pomocí číslic (1 → jedna), zkratky (HIV → há í vé), datumy a časy (12:00 → dvanáct nula nula), matematické symboly (1+1 → jedna plus jedna) či jinak uživatelem definovaná pravidla. Český jazyk (patřící mezi tzv. flexní jazyky) díky své složitosti vytváří problémy, které se například v angličtině nevyskytují. Například skloňování číslovek znesnadňuje převod číslic na text, neboť o tvaru slova je nutné rozhodnout na základě sémantického kontextu věty (2 muži a 2 ženy → dva muži a dvě ženy). Normalizace a porozumění textu je důležitá úloha, jejíž význam v dnešní době plné zpracování textových informací stoupá.

2.1.2 Fonetická transkripce

Po zpracování textu je nutné převést text do fonetické formy, kde jsou jednotlivé znaky reprezentovány symboly z fonetické abecedy. Tomuto procesu se říká *fonetická transkripce*. Fonetická transkripce češtiny je z jedné strany jednoduší než u například angličtiny díky faktu,

že se hlásky čtou stejně tak, jak se píší. Na druhou stranu zde existují výjimky (například spodoba znělosti), které vyžadují nasazení kontextových pravidel pro správný přepis textu.

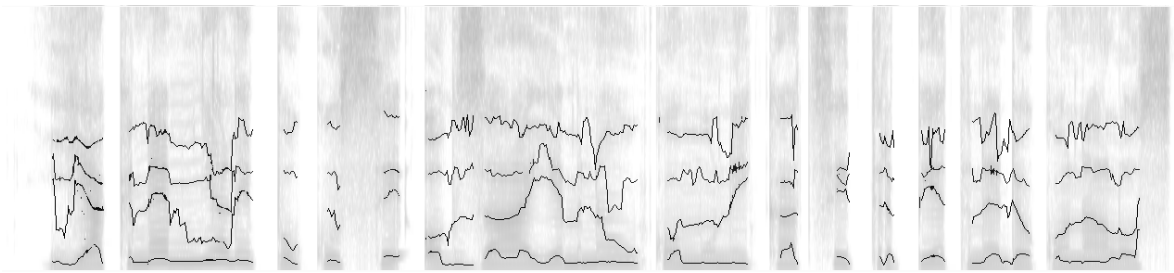
2.1.3 Počátky syntézy

První syntetizér byl postaven na mechanickém principu. Tento „mluvící stroj“ byl sestaven tak, aby napodoboval lidské orgány. Pomocí měchů, které se daly nastavit pomocí pák a ventilu, vyluzoval zvuky podobné lidské řeči. Stroj byl sestrojen v roce 1791 Wolfgangem von Kempelenem. Na začátku 20. století pak začaly vznikat první elektronické syntetizéry, které pomocí rezonančních obvodů dokázali generovat bzučení podobné lidské řeči. Po nástupu číslicových počítačů se objevily digitální verze syntetizérů.

2.1.4 Formantová syntéza

Princip těchto syntetizérů využívá i tzv. *formantová syntéza*. Tato metoda syntézy řeči je založena na akustickém principu tvorby řeči. Snaží se modelovat hlasový trakt pomocí formantů. Formant je v akustice oblast lokálního maxima ve spektru. Vzniká rezonancí v hlasovém traktu. Při vytváření lidského hlasu vytvoří hlasivky svým kmitáním spektrálně bohatý tón. Ten nese základní frekvenci, která se běžně označuje jako frekvence $F0$. Při šíření dutinami vznikají rezonance, kde některé frekvence jsou zesíleny (formanty) a některé naopak zeslabeny (antiformanty). První dva formanty jsou důležité pro rozlišení samohlásek.

Právě tento princip se snaží formantová syntéza napodobit. Syntetizér je vlastně filtr, který modeluje jednotlivé formanty. Vstupem filtru je buzení. Ten je v případě znělé hlásky tvořen signálem se základní frekvenci $F0$ a v případě neznělých hlásek je tvořen šumem. V některých případech se oba zdroje kombinují. Při správném nastavení (mnoha) parametrů generuje syntetizér celkem srozumitelnou řeč.



Obrázek 2.1: Spektrogram signálu ukázkové věty (0-8000Hz). Vyznačené černé čáry zobrazují hodnoty prvních 4 formantů. (Obrázek pochází z výstupu programu představeného v druhé kapitole.)

Formantová syntéza byla v minulosti velmi používaná. Mezi její výhody patří jednoduchost modelu, snadná změna prozodických vlastností, konstantní kvalita a plynulost řeči. Bylo zde i mnoho nevýhod, kvůli kterým se dnes již prakticky nepoužívá. Mezi nevýhody patří hlavně pracné hledání pravidel a obtížné nastavování parametrů, kdy změna jednoho ovlivní nastavení ostatních. Hlavní nevýhodou je však nízká přirozenost řeči.

2.1.5 HMM syntéza

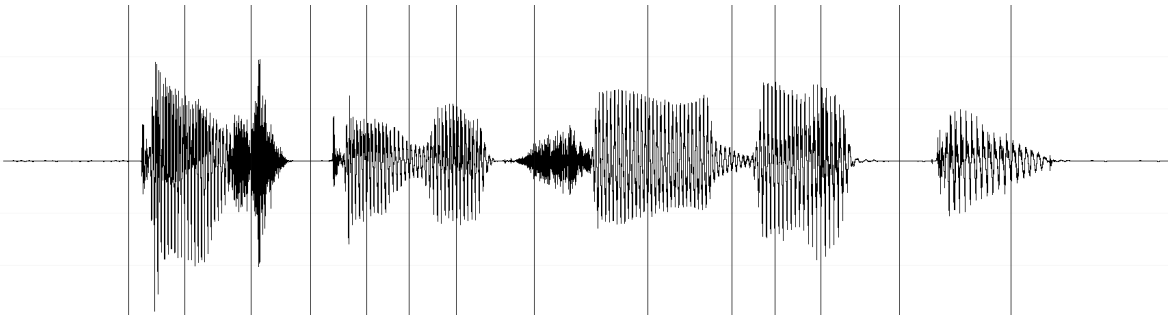
Formantová metoda je založena na modelovém přístupu syntézy řeči. Mnohem modernější metodou, založeném též na modelovém přístupu, je *HMM syntéza*. Ta využívá principu skrytých Markovských modelů. Statistické nastavení parametrů odráží konfiguraci hlasového ústrojí. Tyto parametry však nelze ze signálu přímo pozorovat, proto se jedná o skrytý model a jednotlivé stavy jsou vybírány na základně metody maximální věrohodnosti. Pro generování výsledné řeči je použit opačný směr, kde je z posloupnosti stavů, které jsou dány vstupní posloupností fónů, generován akustický signál na základě parametrů konkrétního stavu opět pomocí zdroje buzení.

HMM se dnes používá velice často. Její výhody jsou konstantní a dobrá kvalita řeči, možnost modelovat různé emotivní či jiné změny hlasu, možnost plynulé změny parametrů syntézy (rychlost, výška, tempo, barva hlasu). HMM syntéza také umožňuje provádět adaptaci hlasu na konkrétního řečníka. K tomu je třeba pouze malé množství nahraných promluv od nového řečníka. Lze tak snadno rozšiřovat databázi hlasů. Mezi další výhody patří nízká výpočetní náročnost (při syntéze) a malé paměťové nároky. Toho lze využít v zařízeních s omezenou výpočetní kapacitou, jako je například mobilní telefon či vestavěný mikropočítač.

HMM syntéza však trpí neduhem přílišného vyhlazení signálu, neboť parametry jsou reprezentovány statistickým průměrem všech natrénovaných dat. Díky tomu zní výstup HMM syntézy trochu plechově a nepřirozeně. HMM je syntéza v současné době nejvíce zkoumanou metodou syntézy řeči.

2.1.6 Konkatenační syntéza

Kromě HMM syntézy se dnes používá také *konkatenační syntéza*. Principem konkatenační metody je spojování úseků, které pochází z již nahraných vět od řečníka, kterého chce syntéza napodobovat. Tyto úseky se vhodně spojí a vzniká tak syntetická řeč. Jako úsek se obvykle používá fón nebo difón (od středu fónu ke středu dalšího fónu).



Obrázek 2.2: Konkatenační syntéza: spojováním signálů z různých vět vzniká syntetická řeč. Svislé čáry znázorňují konkatenační místa.

Difónová syntéza

Konkatenační metoda se dále dělí podle toho, kolik kandidátů je pro daný fón k dispozici. Je možné použít pouze jednoho kandidáta. Tato metoda se anglicky nazývá *diphone synthesis*.

Každý difón jakožto jednotka, je pečlivě vybrán tak, aby šel co nejlépe napojovat na další úseky, a aby reprezentoval průměrnou jednotku.

Při syntéze se z fonetického zápisu vytvoří posloupnost jednotek. Při jejich spojování dochází ke spektrálním modifikacím tak, aby výstupní řeč kopírovala předem definovanou trajektorii frekvence $F0$ a rychlosti řeči. Trajektorie frekvence $F0$ je modelována pomocí sady pravidel prozodie, které se snaží generovat výšku hlasu tak, jak je obvyklé u běžné řeči (klesání hlasu na konci oznamovací věty nebo naopak stoupání na konci otázky).

Modifikace signálu probíhá například pomocí metody PSOLA (*pitch synchronous overlap and add*). Její princip spočívá v rozdělení signálu na malé úseky (obvykle jedna perioda frekvence $F0$). Posunem, přidáním anebo odebráním těchto úseků, dochází ke změně trvání a výšky výsledné řeči. Metoda obvykle pracuje v časové oblasti. Existují však i alternativy, které pracují ve frekvenční oblasti.

Výhodou metody je právě možnost modelovat trajektorii prozódie a rychlosti řeči. Dále malé paměťové a datové nároky. Výhodou je rovněž rychlost syntézy, která probíhá v lineárním čase. Oproti tomu kvalita výsledné řeči není nejlepší. Ve zvuku jsou slyšet ony modifikace signálů, které degradují značně kvalitu a přirozenost.

Unit selection

Druhou konkatenační metodou je tzv. *Unit Selection*. Ta je představitel tzv. *korpusově orientované syntézy řeči*. Této metodě je věnován odstavec 2.2.

2.1.7 Artikulační syntéza

Poslední metoda, která stojí za zmínku je artikulační syntéza. Ta by se mohla jednou v budoucnu stát nejvěrnějším napodobením lidského hlasu. Její princip totiž spočívá v tom, že se simuluje funkčnost jednotlivých orgánů a celkový proces vytváření řeči od základu. Simulovat lze chvění hlasivek, následné šíření vlnění a jeho částečné absorpce či odraz v tkáních zvukového ústrojí. Této metodě se díky své složitosti zatím nedostává velké pozornosti.

2.1.8 Reference

Podrobnější a mnohem ucelenější přehled metod syntézy řeči i celkovou problematiku řečových technologií včetně další témat, jako například rozpoznávání řeči, lze dohledat například v [9] nebo v [10].

2.2 Syntéza s výběrem jednotek

Metoda *Unit Selection* je představitel tzv. *korpusově orientované syntézy řeči*. Zde má každá jednotka velké množství kandidátů, jež pochází z mnoha namluvených vět řečníkem. Tyto jednotky jsou uloženy v databázi jednotek (*řečový korpus*). Z této databáze se vybírají nejlepší kandidáti, které se potom spojí do výstupní věty. Při spojování se provádí minimum signálových modifikací. Signál proto zachovává přirozenost a původní parametry řečníka. Algoritmus výběru jednotek se anglicky nazývá *unit selection*. Touto metodou se zabývá tato práce a je i primárně využívána i v systému ARTIC.

Algoritmus výběru jednotek

Pro vyjádření míry vhodnosti spojení dvou jednotek je nadefinována tzv. hodnotící funkce (*cost function*). Ta se obvykle skládá z dvou částí. Cena cíle (*target cost*) je veličina udávající kontextuální shodnost dvou po sobě jdoucích jednotek. Správné kontextové okolí a hlavně pozice v původní větě, vedou k nízké ceně cíle. Druhou složkou cenové funkce je cena spojení (*join cost*). Na tu lze nahlížet jako na vzdálenost akustických parametrů signálu v místě spojení. V ideálním spojení se shoduje základní frekvence hlasu $F0$ i tendence jejich průběhů v okolí a také spektrum signálu (popisované například pomocí MFCC koeficientů) v daném místě spojení. Algoritmus výběru jednotek hledá nejlepší posloupnost jednotek z řečové databáze tak, aby kumulativní součet hodnotící funkce na jednotlivých spojích byl co nejmenší.

Prohledávání stavového prostoru všech kombinací jednotek je velice časově náročná úloha. Pomocí určitých optimalizací a heuristik lze však náročnost značně snížit. Jako základ řešení prohledávání prostoru kombinací se obvykle používá *Viterbiův algoritmus*.

Inventář řečových jednotek

Řečové jednotky jsou uloženy v inventáři. V této databázi jsou uloženy zdrojové promluvy a z nich vypočtené parametry: průběhy $F0$, energie a mfcc. Zdrojové promluvy jsou rovněž nasegmentované, tj. jsou rozděleny podle jednotlivých hlásek. Vytváření inventáře řečových jednotek probíhá zcela automaticky. Díky tomu lze vytvářet obrovské inventáře a tím tak zajistit dostatečný počet kandidátů pro každou jednotku.

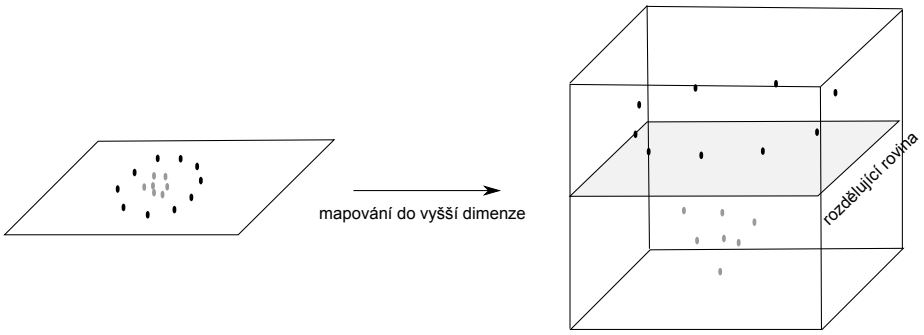
Výhody a nevýhody

Řeč produkovaná metodou unit selection vyniká vynikající kvalitou a přirozeností. Je to dáno tím, že se signál nijak nemodifikuje, ale pouze se vhodně skládá. Generování neznělých a šumivých tónu zde není problém, tak jako je tomu u modelových metod. V místě spojování vzniká bohužel potenciální místo problému. Při špatném zřetězení jednotek se v jinak bezchybné řeči vyskytne lokální propad v kvalitě, což působí velmi rušivě a kazí to celkový dojem z celé věty. Výskytů a předcházení tomuto jevu (dále označován jako *řečový artefakt*) se věnuje tato práce podrobněji v dalších kapitolách.

Občasné chyby v řetězení jsou velká nevýhoda metody unit selection. Mezi další nevýhody patří velké paměťové, datové a výpočetní nároky, neboť se musí procházet mnoho stavů s mnoha jednotkami, kde se s každou jednotkou váže i odpovídající zvukový signál. Pomocí unit selection se také obtížně vytváří jiná než neutrální řeč. Mnoho experimentů se snaží použít unit selection na vytváření emotivně zabarvené, či jinak upravené řeči, ale tím vznikají další problémy a velmi stoupá množství jednotek v řečovém korpusu.

Použití

Metoda unit selection se dnes používá v případech, kde je požadavek na neutrální, nijak emotivně nezabarvenou, co nejvíce kvalitní a přirozenou řeč. A také tak, kde není problém zajistit vyšší výpočetní a paměťové zdroje.



Obrázek 2.3: Mapování do vyšších dimenzí, kde již lze jednotlivé množiny oddělit

2.3 Support Vector Machines

Support vector machines (SVM) je poměrně mladá metoda strojového učení. Poprvé byla metoda představena tak, jak je dnes používána, v roce 1995 v práci [2]. SVM je v základní verzi lineární binární klasifikátor (tj. rozděluje vstupní data do dvou tříd na základě rozhodnutí založeném na hodnotě lineární kombinace příznaků), který pracuje na principu učení s učitelem.

Základní lineární klasifikátory jsou obvykle postaveny na jednoduchém principu a díky tomu jsou i algoritmy výpočtu velmi efektivní. Mezi takové metody lze zařadit například jednovrstvá neuronová síť. Tyto klasifikátory jsou však omezené na klasifikaci v lineárně separabilních oblastech.

Naproti tomu existují klasifikátory složitější, které dokáží rozdělit oblast i pomocí obecně nelineárních funkcí. Mezi takové patří například vícevrstvá neuronová síť se sigmoidální aktivační funkcí, kde se pro učení používá algoritmus zpětného šíření (backpropagation). Jejich nevýhodou je však obtížné a hlavně pomalé učení velkého počtu parametrů a riziko konvergence do lokálního minima.

SVM kombinuje výhody obou skupin. Data jsou nejdříve převedena do prostoru vyšší dimenze, ve které jsou již data lineárně oddělitelná. Pro rozdělení nadroviny jsou v trénovacích datech hledány vektory, které leží blízko oddělující roviny. Tyto vektory jsou poté použity pro popsání rozdělující plochy. Jsou označovány jako tzv. podpůrné vektory, odtud pochází původ názvu klasifikátoru.

2.3.1 Princip klasifikátoru

Pro převod do vyšších dimenzí využívá SVM *jádrovou transformaci* (kernel transformation). Pro zachování jednoduchosti je algoritmus navržen tak, aby výpočet pracoval pouze se skalárním součinem transformačních funkcí. Ten lze určit pomocí tzv. jádrové funkce (kernel function), která se počítá stále v původním prostoru hodnot. Tímto lze pracovat v prostoru vyšších dimenzí bez potřebného mapování příznaků do těchto prostorů.

Jádrových funkcí $K(x_i, x_j)$ existuje velké množství. Tím získává SVM klasifikátor na obecnosti, kdy použití různé jádrové funkce vede k různým vlastnostem. Při znalosti konkrétní úlohy lze dokonce určit vlastní jádrovou funkci, která lépe popíše daný problém.

Základní jádrové transformace pro SVM klasifikátor jsou:

- Lineární: $K(x_i, x_j) = x_i^T x_j$
- Polynomiální: $K(x_i, x_j) = (x_i x_j + 1)^d$
- S radiálním základem (RBF): $K(x_i, x_j) = e^{-\gamma \|(x_i - x_j)\|^2}$
- Sigmoidální: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

SVM hledá minimum následujícího výrazu:

$$\min_{\mathbf{w},\xi,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \right\} \tag{2.1}$$

kde w a b jsou parametry rozdělující roviny a ξ_i je volný parametr, který udává míru chybné klasifikace pro vstup x_i. Tento parametr byl zaveden v [2] pro popis tzv. hladkého rozpětí, což je rozšíření lineárního SVM které připouští existenci špatně klasifikovaných vzorků. Pro tento parametr platí vztah:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \tag{2.2}$$

kde $y_i \in \{-1, 1\}$ je požadovaný výstup a $x_i \in \mathbb{R}^n$ je vstupní vektor.

Při bližším pohledu na vztah 2.1 lze pozorovat, že rozšířená verze SVM obsahuje ve výrazu součet dvou hodnot. První hodnota $\frac{1}{2}\|\mathbf{w}\|^2$ udává míru příspěvku jednotlivých vzorků do výsledné roviny, kde nízká hodnota představuje hladký tvar, kdežto vysoká hodnota velmi kolísavý a prudce se měnící tvar. Druhá hodnota $\sum_{i=1}^{n} \xi_i$ minimalizuje míru chybné klasifikace. Nízká hodnota znamená velkou přesnost klasifikátoru na učících datech. Preference jedné či druhé hodnoty určuje konstanta C. Její hodnotou lze nastolit správný poměr mezi rozdělující nadrovinou s plynulým průběhem a mírou chybné klasifikace.

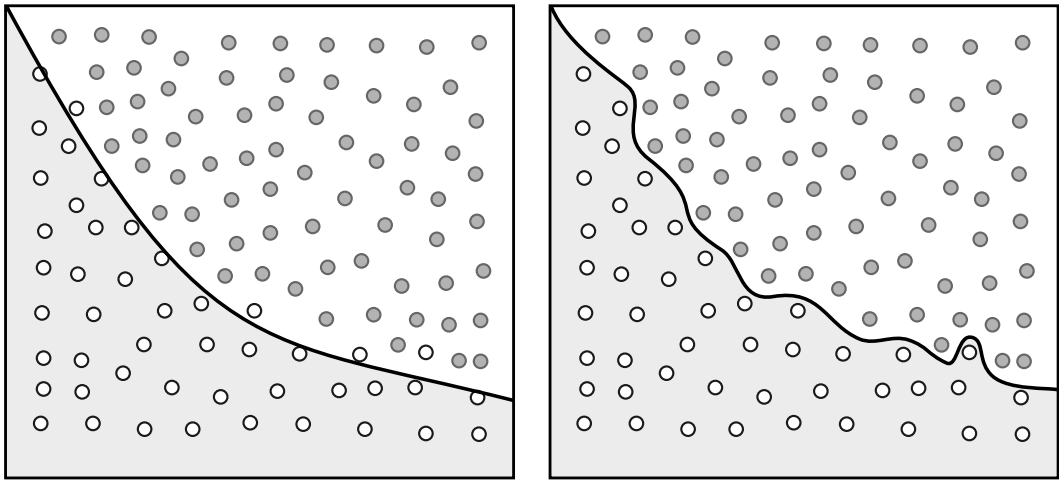
2.3.2 Trénování

Správné nastavení parametru C, spolu s parametry jádrové transformace, se obvykle provádí pomocí metody *grid search*. Tato metoda pracuje na principu prohledávání hrubou silou, kde se postupně zkouší všechny n-tice z kartézského součinu množin kombinací všech parametrů. Aby byl výpočet rychlý, provádí se nejprve hledání na hrubém nástřelu parametrů s malým počtem kombinací parametrů a teprve poté se provádí jemnější nastavování parametrů v místě, které bylo hrubým hledáním označeno za nejlepší.

Při trénování klasifikátoru je velmi důležité dodržovat několik podstatných věcí. Při jejich nedodržení totiž velmi rychle klesá úspěšnost klasifikace. Jedním z takových pravidel je dodržení vyvážení trénovací množiny. Obě třídy vzorků by měli být reprezentovány nejlépe stejným počtem vzorků. Tento problém lze částečně vyřešit použitím vah u tříd. Při vyvážené trénovací množině však podává klasifikátor nejlepší výsledky.

Druhá věc, která může způsobit velmi vážné propady úspěšnosti v klasifikaci (dle [3]) je správné škálování rozsahu příznaků. Všechny hodnoty příznaků musí mít totiž stejný interval rozpětí. Obvykle se volí -1 až +1. Pro tento rozsah byl totiž klasifikátor původně navržen.

Je důležité dodržovat striktní oddělení trénovacích a testovacích dat. Při hodnocení úspěšnosti klasifikátoru se musí používat jiná data, než která byla použita pro učení klasifikátoru. Jinak je velké riziko tzv. přetrénování klasifikátoru na danou trénovací množinu.



Obrázek 2.4: Příklad přetrénování klasifikátoru. Správné natrénování (vlevo) sice občas neklasifikuje správně, ale kopíruje obecnou skutečnost rozdělení příznaků. Naproti tomu přetrénovaný klasifikátor (vpravo) má pro konkrétní testovací data větší úspěšnost, při jiném výběru dat by však nedosahoval tak dobrých výsledků.

Přetrénování (overfitting) znamená, že díky omezenému počtu trénovacích vzorků se klasifikátor neučí obecné souvislosti, ale konkrétní rozložení trénovacích vzorků. Pro trénovací data pak má velkou úspěšnost klasifikace, při použití na jiných datech však úspěšnost prudce klesá. Problém přetrénování je znázorněn na obrázku 2.4. SVM je znám pro svou schopnost být odolný vůči přetrénování (narozdíl například od trénování pomocí nejbližšího souseda), přesto je však nutné tomuto jevu věnovat pozornost.

2.3.3 Klasifikace

Poté co je klasifikátor naučen na trénovacích datech, nastává druhá fáze: fáze klasifikace. V tomto režimu jsou klasifikátoru předkládány nová data. Úkolem klasifikátoru je zařadit každý vzorek do jedné ze tříd. Využívá k tomu znalosti, které získal z trénovacích dat.

V případě SVM klasifikátoru to vyžaduje převedení vektoru příznaků do vyšší dimenze pomocí jádrové transformace a dosazení do rovnice nadroviny. Z toho lze určit, do jaké třídy vektor příznaků patří. Tento krok není náročný na výpočet. Fáze klasifikace je proto velmi rychlá.

2.3.4 Validace

Při trénování klasifikátoru se velmi často používá tzv. *křížová validace* (cross validation). Tato metoda slouží pro validaci natrénovaného modelu, kde se testuje jeho nezávislost na trénovacích datech. Princip metody je v postupném rozdělování množiny trénovacích vzorků na trénovací a testovací část, kde je vždy klasifikátor natrénován na trénovací množině a vyhodnocen na testovací množině. Tento postup se opakuje několikrát po sobě. Při každém kroku je zaznamenána úspěšnost a celkový výsledek je poté určen průměrem hodnot spolu se směrodatnou odchylkou. Data lze rozdělovat pomocí vzoru (např k-násobná validace) nebo zcela náhodně v určeném poměru (obvykle 70% k 30%).

2.3.5 Nástroje

Klasifikátor SVM je implementován v širokém spektru nástrojů. Je součástí matematických programů (jako např Matlab), stejně tak v knihovnách pro věděcké výpočty, které jsou dostupné pro různé programovací jazyky (např. Scikit-learn [8] pro Python). Základní implementací pro většinu z nich je však knihovna LIBSVM [1]. Je napsaná v programovacím jazyce C++, ale existuje napojení na téměř všechny ostatní jazyky. Sestává se ze tří základních nástrojů: svm-scale, svm-train a svm-predict pomocí nichž lze škálovat vstupní data, trénovat a ověřovat SVM klasifikátor. Jako vstup je používán jednoduchý textový formát.

2.4 Hodnocení úspěšnosti klasifikátoru

Pro hodnocení úspěšnosti klasifikátoru jsou používány následující standardní míry úspěšnosti. Testování úspěšnosti se provádí obvykle na datech, které nebyly použity při procesu učení klasifikátoru. Pro vyhodnocení je zaznamenán počet správných a špatných odpovědí. Správná odpověď nastane v případě, že klasifikátor přiřadí testovací vzorek do stejné třídy, ze které vzorek pochází. Celkem jsou z jedné testovací sady vyhodnocena čtyři čísla. Ty udávají četnosti typů odpovědí podle hodnoty predikované a skutečné. Pro názornost jsou vypsány do tabulky:

		Skutečná hodnota	
		Ano	Ne
Predikce	Ano	*true positive* (tp) správně klasifikovaný pozitivní vzorek	*false positive* (fp) špatně klasifikovaný negativní vzorek
	Ne	*false negative* (fn) špatně klasifikovaný pozitivní vzorek	*true negative* (tn) správně klasifikovaný negativní vzorek

Z těchto čísel jsou vypočtené standardní míry úspěšnosti:

Přesnost (*Precision*)

$$P = \frac{tp}{tp + fp} \tag{2.3}$$

Přesnost je pravděpodobnost, že náhodný vzorek, který klasifikátor označil pozitivně, je ve skutečnosti opravdu pozitivní vzorek.

Úplnost (*Recall*)

$$R = \frac{tp}{tp + fn} \tag{2.4}$$

Úplnost je pravděpodobnost, že náhodný vzorek, který je doopravdy pozitivní, bude správně označen klasifikátorem jako pozitivní.

F1-hodnota

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.5}$$

Pro porovnání klasifikátoru jedním číslem se používá F1 hodnota. Ta kombinuje přesnost a úplnost do jednoho čísla. Jedná se o harmonický průměr těchto hodnot. Výše této hodnoty zpravidla vypovídá o celkové úspěšnosti trénování klasifikátoru neboť F1 hodnota je nejvyšší, pokud je jak přesnost, tak i úplnost vysoká.

Celková úspěšnost (*Accuracy*)

$$A = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.6}$$

Celková úspěšnost je pravděpodobnost, že klasifikátor náhodný vzorek klasifikuje správně.

Preference trénování

V některých případech lze při trénování klasifikátoru určit, zda se má spíše maximalizovat přesnost nebo úplnost. Obecně je však snaha při procesu trénování maximalizovat obě hodnoty, což znamená maximalizaci $F1$, v důsledku i A.

2.5 Míry shody posluchačů

2.5.1 Cohenova Kappa

Cohenova kappa neboli míra souhlasu je statistická míra vyjádření kvalitativní shody dvou hodnocení (to může vytvářet například anotátor nebo i klasifikátor). Oproti prostému procentuálnímu vyjádření shodných odpovědí, Cohenova kappa započítává do výsledku i pravděpodobnost náhodné shody a je tak proto více robustnější.

Výsledek míry souhlasu je číslo v intervalu 0 až 1, kde 1 znamená absolutní souhlas a 0 absolutní nesouhlas. Cohenova Kappa je používává v kapitole 5 pro porovnávání dvou posluchačů v poslechových testech.

Výpočet

Vzoreček výpočtu je:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{2.7}$$

kde $\Pr(a)$ je poměr shodných odpovědí k celkovému počtu a $\Pr(e)$ je pravděpodobnost náhodné shody odpovědí.

Příklad

Mějme 2 anotace stejných dat. V datech se nachází n vzorků. Pro celý vzorek spočteme četnost případů, kdy obě odpovědi jsou kladné, záporné a nebo různé. Četnosti lze pro přehlednost zapsat do následující tabulky:

	Ano	Ne
Ano	n_{11}	n_{12}
Ne	n_{21}	n_{22}

Potom poměr shodných odpovědí je:

$$\Pr(a) = \frac{n_{11} + n_{22}}{n}$$

a pravděpodobnost náhodné shody je:

$$\Pr(e) = \frac{(n_{11} + n_{12}) \cdot (n_{11} + n_{21}) + (n_{22} + n_{12}) \cdot (n_{22} + n_{21})}{n^2}$$

2.5.2 Fleissova Kappa

Flesissova kappa je podobná míra jako Cohenova kappa. Na rozdíl od ní dokáže však vyjádřit míru shody nad libovolným počtem hodnocení. Důležitý rozdíl je však v tom, že Cohenova kappa předpokládá, že hodnocení vždy pochází od stejných hodnotitelů. Naproti tomu Fleissova kappa to nepředpokládá a vyžaduje vždy jen stejný počet hodnotitelů. Princip výpočtu je podobný. Opět se bere v potaz pravděpodobnost náhodné shody. Vzoreček výpočtu je:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{2.8}$$

$$\bar{P} = \frac{1}{Nn(n-1)}(\sum_{i=1}^{N}\sum_{j=1}^{k} n_{ij}^2 - Nn) \tag{2.9}$$

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2 \tag{2.10}$$

$$p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij} \tag{2.11}$$

N vyjadřuje počet vzorků, na které se odpovídá a n je počet hodnotitelů, k je počet tříd do kterých se klasifikuje. $n_{i,j}$ udává počet hodnocení, které přiřadily i-tý vzorek do j-té třídy.

Kapitola 3

Program pro analýzu syntetické řeči

V rámci této práce, byl vytvořen program pro analýzu syntézy řeči. Ten nejprve sloužil jako jednoduchá grafická vizualizace sekvence jednotek z výstupu syntézy řeči. Postupně však program získával nové a nové funkce. V aktuálním stavu disponuje širokým spektrem funkcí jak pro analýzu procesu syntézy, tak i odhalování příčin vzniku problému. Zároveň s tím obsahuje i nástroje, jak danou příčinu rovnou vyřešit (oprava segmentace a její uložení zpátky do řečového korpusu).

Tento program byl použit pro analýzu rušivých míst (artefaktů) v syntetických promluvách, kde sloužil jako nástroj pro rychlé otevření konkrétního místa v syntetické promluvě a jeho zkoumání z hlediska průběhů audio vlny a parametrů. S jeho pomocí byly artefakty kategorizovány a analyzovány. Tato kapitola se věnuje představení tohoto programu a jeho funkcím.

3.1 O programu

Program byl vyvíjen pod kódovým označením *Prokus*[1]. Jeho zadání a vývoj byl veden ve spolupráci s katedrou kybernetiky FAV ZČU v Plzni. Nové funkce tak vznikaly z reálných požadavků na řešení problémů, primárně tak vývoj nesměřoval k požadavkům této práce, ale držel se dlouhodobějších plánů.

Jeho vývoj začal v lednu 2012. Je naprogramován v jazyce C++ a je postaven nad knihovnou *Qt*. Díky této kombinaci je možné program sestavit jak pro operační systém Windows tak i pro Linux.

Díky tomu, že je program napsán v C++, bylo velmi jednoduché provést napojení na současný systém syntézy řeči vyvíjený na katedry kybernetiky ARTIC [6]. Jsou do něj rovněž zakomponované moduly fonetické transkripce a předzpracovaní textu.

Program je zapojen do řetězu automatického sestavování a kompilace. Při úpravě zdrojových kódů či změně knihovny ARTIC je automaticky sestaven na nejnovější verzi. Zdrojové kódy jsou uloženy v SVN repositáři, který spravuje katedra kybernetiky FAV ZČU.

[1] Název naznačuje nejčastější činnost uživatele: prokousávání se syntézou.

3.2 Funkce programu

Následující text obsahuje stručný výčet vlastností programu.

3.2.1 Zobrazení zvukového signálu

Jak promluva, která se nachází v korpusu systému syntézy řeči, tak i syntetická řeč se sestává z audio signálu, který je zobrazen ve audio vlny. Program dokáže zobrazit audio záznam v libovolném přiblížení.

Přehrávání

Audio signál lze libovolně označit a nechat si přehrát. Pokud data obsahují segmentace, lze výběr provádět buď po celých jednotkách nebo bez omezení.

Export do souboru

Celý audio signál nebo jen označený úsek lze vyexportovat na disk do audio souboru. Ten je ve formátu wave.

Spektrum

Spolu s pohledem na audio vlnu je ve výchozím pohledu zobrazeno i frekvenční spektrum signálu. To hraje velmi významnou roli při analýze řečového signálu. Pro jeho výpočet se používá STFT (*short-time Fourier transform*). Parametry FFT (počet vzorků, krok, velikost okna) lze konfigurovat. Nastavení parametrů FFT přes grafické rozhraní lze pozorovat na obrázku 3.5. Samotné spektru včetně průběhu formantů je ukázáno na obrázku 3.6.

Průběh parametrů

Program rovněž umožňuje sledovat průběhy hodnot energie, $F0$ a formantů. Tyto hodnoty jsou načítány z dodatečných souborů a jsou zakresleny do grafu (viz. obrázek 3.4).

Časová osa

Pod signálem je zobrazena časová osa. Ta umožňuje uživateli získat přehled o aktuálním měřítku a také kurzorem odměřovat čas úseku. Umožňuje také rychlý přesun na konkrétní čas.

3.2.2 Editace segmentace v řečovém korpusu[2]

V programu je možné otevřít a upravovat libovolnou větu z řečového korpusu. Ten je nejdříve nutno nakonfigurovat dodáním cest k datovým souborům. Takto lze v programu vytvořit databázi všech dostupných hlasů.

[2] Vysvětlení pojmů segmentace a řečový korpus lze dohledat v kapitole 2.

Otevření věty z řečového korpusu

Při načtení korpusu je uživateli vypsán seznam vět z aktuálního korpusu. Uživatel může vyhledat nebo vybrat větu, kterou chce zobrazit (viz. obrázek 3.1 a 3.2).

Posun hranic jednotek

Při větším přiblížení se přes aktuální pohled zobrazí hranice jednotek. Hranice lze pomocí kurzoru myši libovolně posouvat (viz. obrázek 3.3).

Přidání, úprava a smazání jednotek

Do sekvence jednotek lze provést rovněž úpravy. Vložení či smazání lze provést pomocí vybrání místa a akce z kontextového menu.

Segmentace pro systém ARTIC obsahuje jak dělení na skutečná slova tak i na prozodická slova, kde například předložka je součástí následujícího slova, neboť se vyslovuje spolu s ním. Program tak umožňuje editovat i prozodická slova.

Export hranic zpět do systému

Všechny provedené změny lze uložit a změny přenést zpátky do datových souborů systému (viz. obrázek 3.12). Pomocí programu je tak možné provádět veškeré ruční úpravy segmentace v grafickém rozhraní bez nutnosti spouštět externí nástroje.

3.2.3 Syntéza řeči

Druhá hlavní funkce programu je přímá integrace se systémem syntézy řeči ARTIC. Ten je zakomponován přímo do nitra programu a využívá tak naplno veškeré jeho funkce. Uživatel může spustit syntézu řeči a upravit parametry v grafickém rozhraní.

Jako vstup lze zadat buď fonetický zápis fráze nebo prostý text (viz. 3.7). Program posléze zajistí spuštění všech procesů pro vygenerování syntetické věty. Výstup syntézy je poté zobrazen na další kartě (viz. 3.8).

Alternativy fonetické transkripce

Text může mít více variant přepisu fonetické transkripce. Lišit se může například učebnicová výslovnost od té běžně používané (dětský → [ďecký], [ďetskí]). Program umožňuje procházet všechny varianty přepisu vstupní fráze.

Náhled vstupu a výstupu modulu zpracování textu

V programu je rovněž obsažen modul pro zpracování textu. Ten umožňuje například převést číslovky do textové podoby. Normalizace textu se provádí automaticky.

Konfigurace parametrů syntézy

Před spuštěním syntézy je možno konfigurovat všechny parametry syntézy. Mezi ně patří například parametry prořezávání, které redukují množství kandidátu při procesu výběru jednotek [11]. Vyšší prořezávání snižuje čas potřebný pro syntézu věty. Může ale způsobit, že se vybere jiná posloupnost jednotek. Zbylé parametry se vztahují na popis vstupní věty: kontextové okolí a typ věty.

3.2.4 Analýza syntetizované věty

Po syntéze řeči se v programu otevře nová karta, ve které se nachází podobný obsah jako když byla otevřena věta z korpusu (viz. obrázek 3.11). Jsou zde však navíc některé informace. Ve větě jsou vyznačeny hranice jednotek, ze kterých syntetická věta vzešla.

Proklik jednotky do originální věty

U každé jednotky je interně uložena informace, z jaké zdrojové promluvy z řečového korpusu pochází. Program umožňuje uživateli prokliknout danou jednotku do její originální věty. Lze se tak velmi rychle dostat ke zdroji problému. Tato funkce zjednodušuje opravy anotace.

Znázornění průběhů parametrů ze zdrojových promluv

Podobně jako u věty z korpusu, program zobrazuje průběh energie, $F0$ a formantu. Kolem místa spojení jsou navíc zobrazeny i průběhy z okolních jednotek, tak jak by pokračovaly, kdyby nedošlo ke konkatenaci. Z překryvu je tak možné pozorovat, jestli průběhy hodnot sledují správný trend (viz. obrázek 3.9).

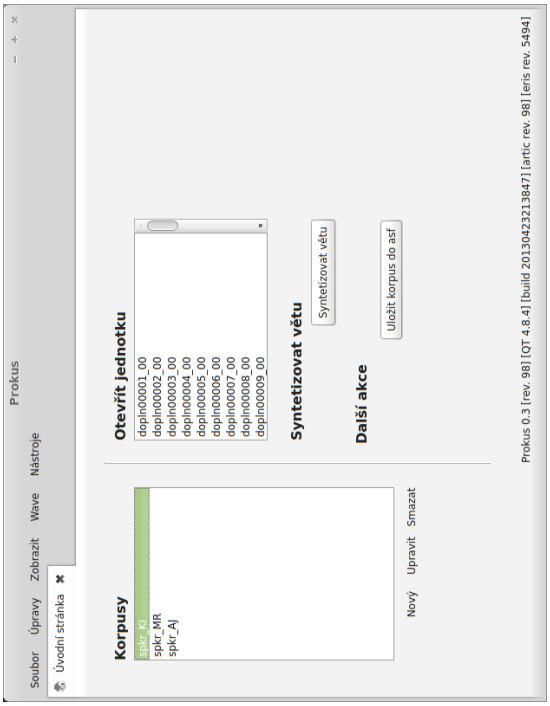
Alternativy jednotky

Tato velmi důležitá funkce umožňuje zkoušet různé sekvence jednotek, a tím snáze identifikovat jednotku, která způsobuje problém. Pro každou jednotku může uživatel vybrat ze seznamu kandidátu libovolnou alternativu (viz. obrázek 3.10). Tou může nahradit původní jednotku, která ale byla vybrána systémem syntézy řeči jako nejlepší.

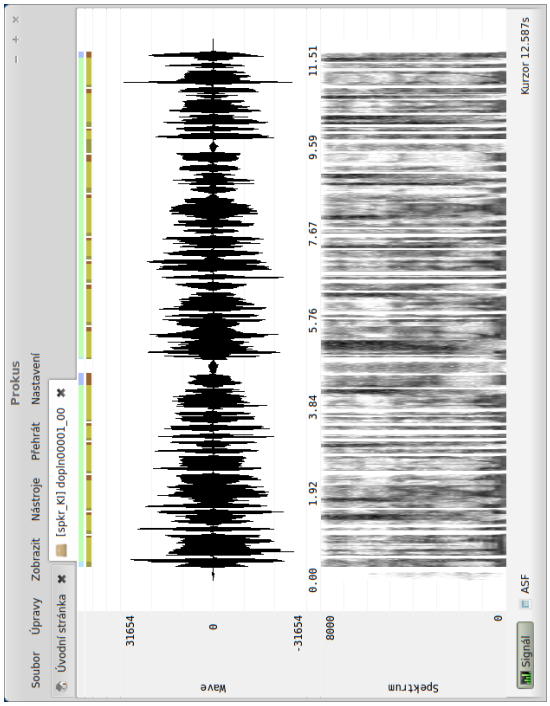
Použití lokální segmentace

Program vnitřně upravuje systém ARTIC tak, aby bylo možné při syntéze použít aktuální segmentaci jednotek namísto té, která se nachází v originálních datových souborech. Uživatel tak může ihned sledovat, zda se změna segmentace projeví na výstupu syntézy konkrétní fráze.

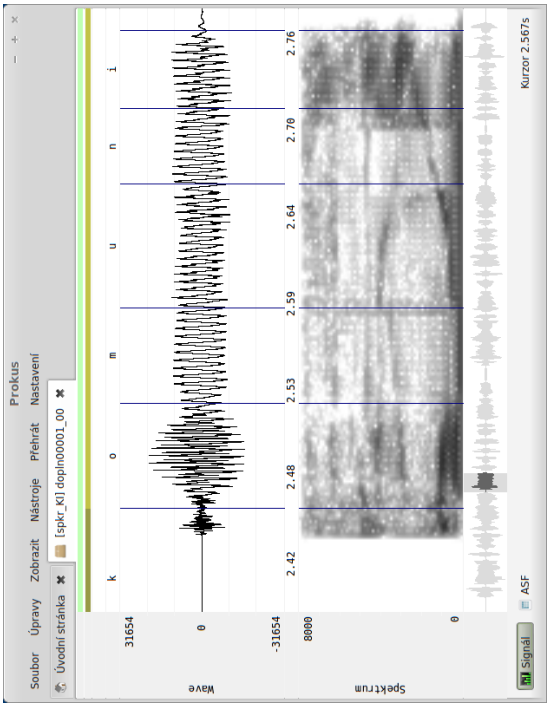
3.3 Grafické rozhraní



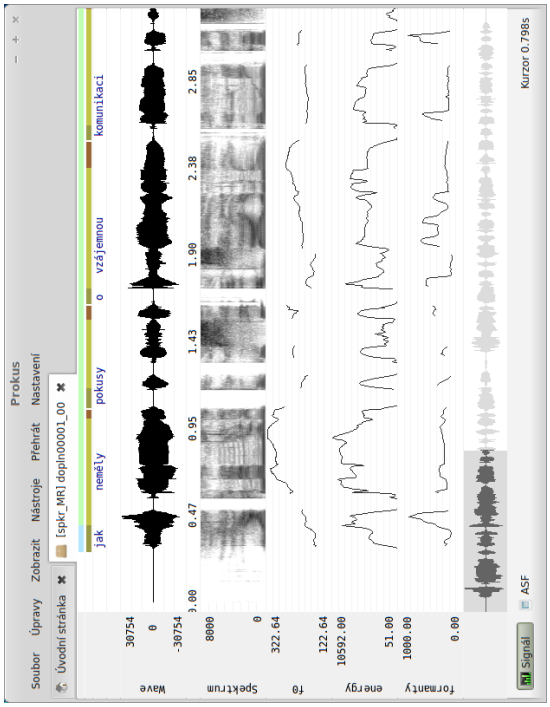
Obrázek 3.1: Rozcestník



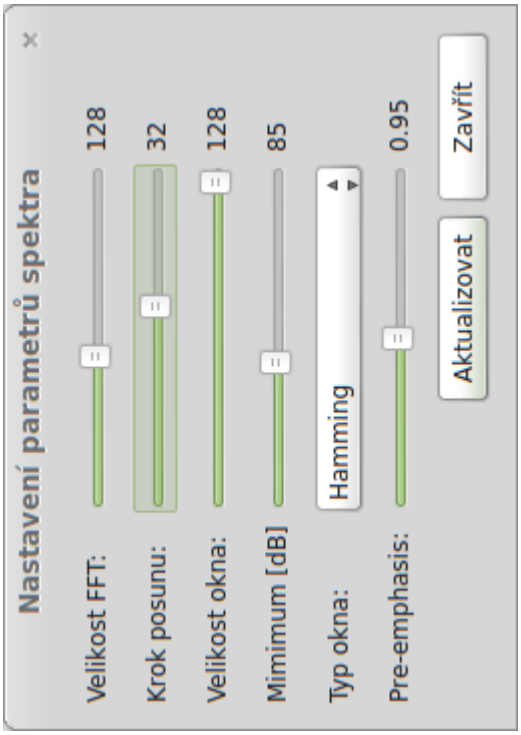
Obrázek 3.2: Otevření věty z řečového korpusu



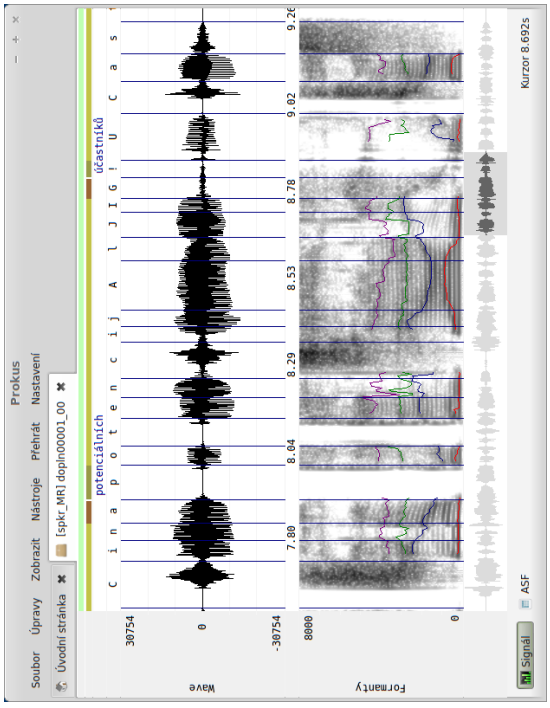
Obrázek 3.3: Editace segmentace jednotek



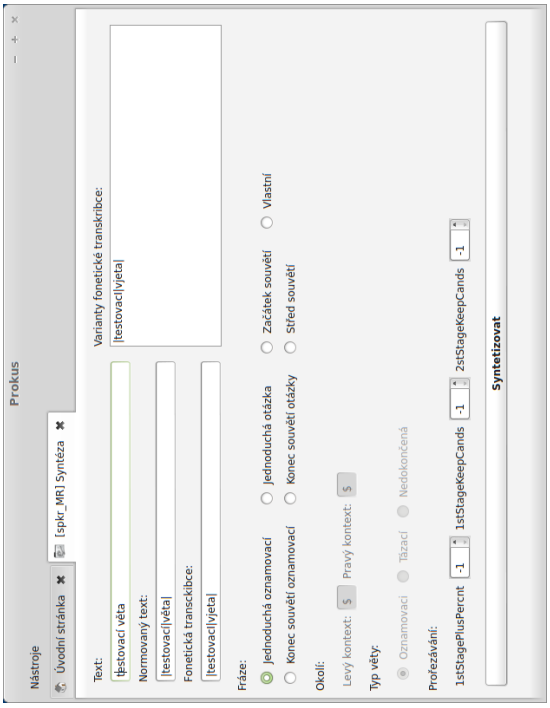
Obrázek 3.4: Průběh $F0$, energie a formantu



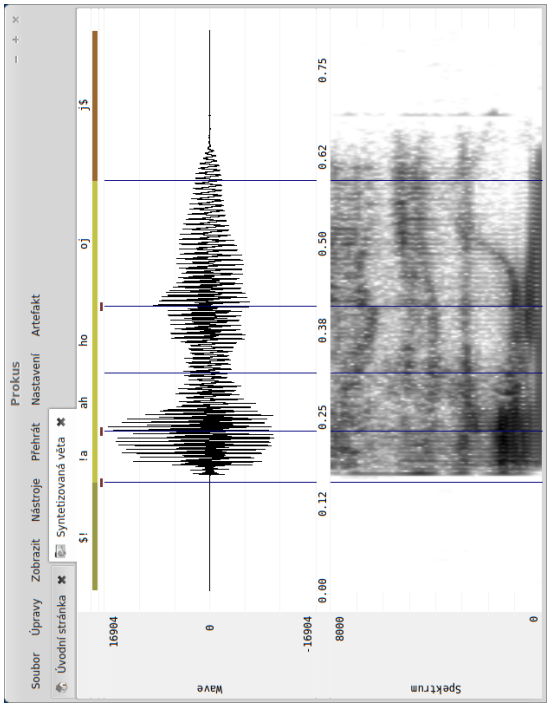
Obrázek 3.5: Nastavení parametrů FFT



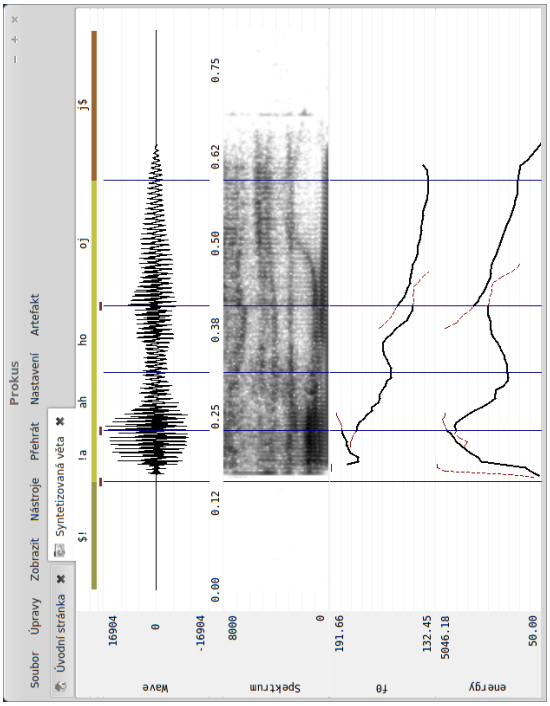
Obrázek 3.6: Zobrazení formantů ve spektru



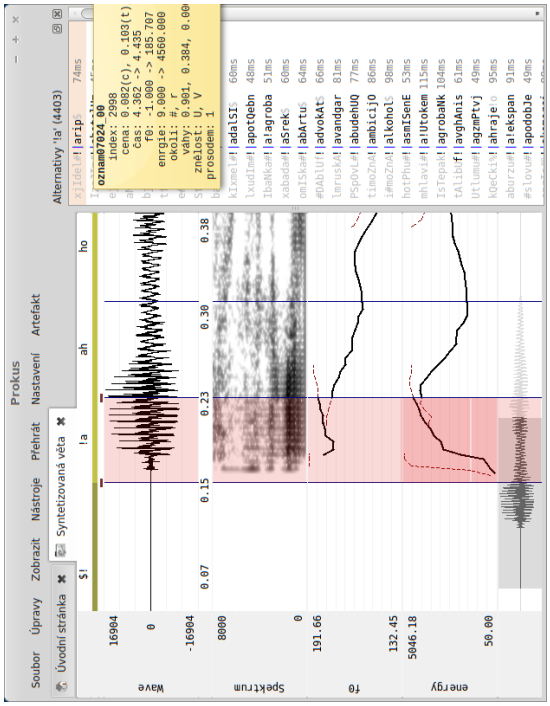
Obrázek 3.7: Syntéza nové věty



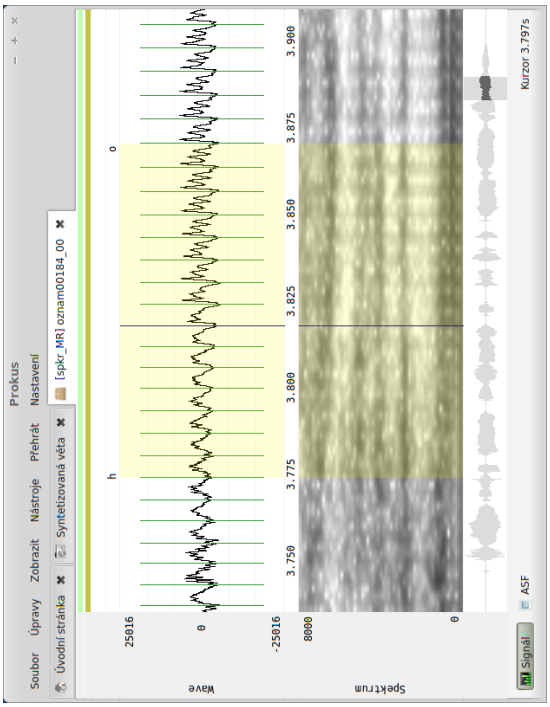
Obrázek 3.8: Výstup syntézy



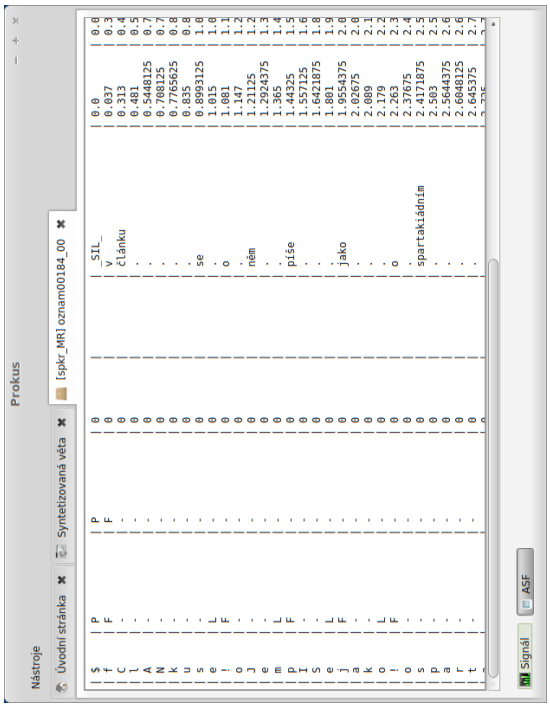
Obrázek 3.9: Překryv průběhu $F0$ a energie



Obrázek 3.10: Výběr alternativní jednotky



Obrázek 3.11: Otevření zdrojové jednotky



Obrázek 3.12: Interní formát segmentace

Kapitola 4

Analýza artefaktů

4.1 Definice

Řečový artefakt je rušivé místo v syntetické řeči, které má lokální charakter, a které způsobuje slyšitelný propad kvality řeči. Většinou tento jev vzniká v místech konkatenace jednotek, jež pochází z různých vět. Jednotky jsou sice vybírány podle hodnotící funkce tak, aby na sebe mimo jiné nejlépe navazovaly, ale v některých případech ani to nestačí.

4.2 Příčiny vzniku

Vznik řečových artefaktů má mnoho příčin. Následující výčet popisuje nejčastější z nich. Můžou ale existovat i další, protože proces vnímání lidské řeči mozkem je velmi komplikovaný a to, jak člověk vnímá syntetickou řeč, může ovlivňovat mnoho faktorů.

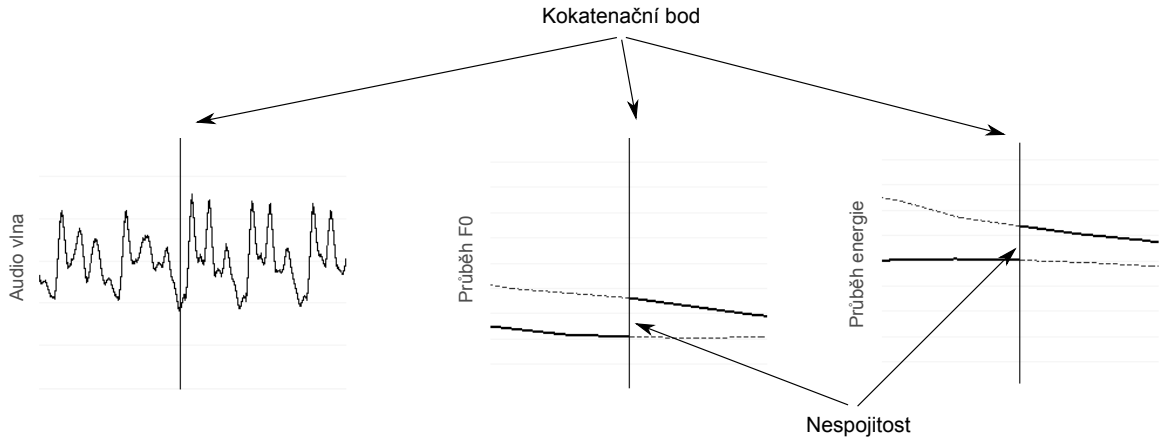
Špatná segmentace jednotek

Při konkatenační syntéze jsou vybrané jednotky z řečového inventáře spojovány. Aby tento proces fungoval, musí být zdrojové promluvy v řečovém korpusu nasegmentované. To znamená, že je věta rozdělena na úseky, kde jeden úsek reprezentuje jednu hlásku. Tato segmentace je v moderních systémech syntézy řeči prováděna téměř výhradně automaticky.

Pokud je však segmentace provedena chybně, může vzniknout problém, kdy konkrétní řečový signál neodpovídá dané jednotce. Tím vznikne v syntetické řeči velmi slyšitelná chyba, neboť na sebe signály vůbec nenavazují v kontextu vyslovované věty, nebo neodpovídají textu, který se má syntetizovat [7].

Chyba segmentace také může vzniknout tak, že řečník špatně přečte daný text. Například spolkne písmeno nebo použije nespisovný výraz. V takovém případě je pak ve zdrojové větě označena jednotka, která tam vůbec není.

Oprava této chyby vyžaduje ruční zásah znalého uživatele, který dokáže objevit místo špatné segmentace a ručně ho opravit. Program představený v kapitole 3 tento proces usnadňuje, neboť dokáže zobrazit umístnění jednotek ve zdrojových promluvách. Při otevření zdrojové promluvy lze pak snadno odhalit příčinu problému, rovnou ji odstranit a uložit změny.



Obrázek 4.1: Zobrazení průběhu řečového signálu, $F0$ a energie v místě konkatenace artefaktu. Lze sledovat nespojitosti v průběhů parametrů v místě spojení.

Nespojitost v průběhu hlasivkového tónu

Problém nespojitosti vzniká při napojení jednotek, které nemají v místě spojení shodnou frekvenci hlasivkového tónu $F0$. Vyskytuje se tak pouze u znělých hlásek (nejvíce u samohlásek, jak je zmíněno v [5] a potvrzeno v předposlední kapitole v tabulce).

Frekvence $F0$ hraje asi největší vliv na vznik řečového artefaktu. Proto je na ní kladen velký důraz. Diference její hodnoty v místě konkatenace je jednou ze složek hodnotící funkce. V optimálním případě by tak měla přímo navazovat.

Existují názory, že i když je v místě konkatenace hodnota frekvence stejná, může i přesto spojení znít nepřirozeně. To nastane, pokud nesedí trend jejího průběhu v čase. Například pokud v minulé jednotce byl rostoucí trend (výška hlasu stoupala) a v aktuální je trend klesající.

Pokud je diference malá, nelze předem odhadnout, zda bude v místě řečový artefakt. Pokud je diference v místě konkatenace velká, je velká pravděpodobnost, že se v daném místě artefakt nachází.

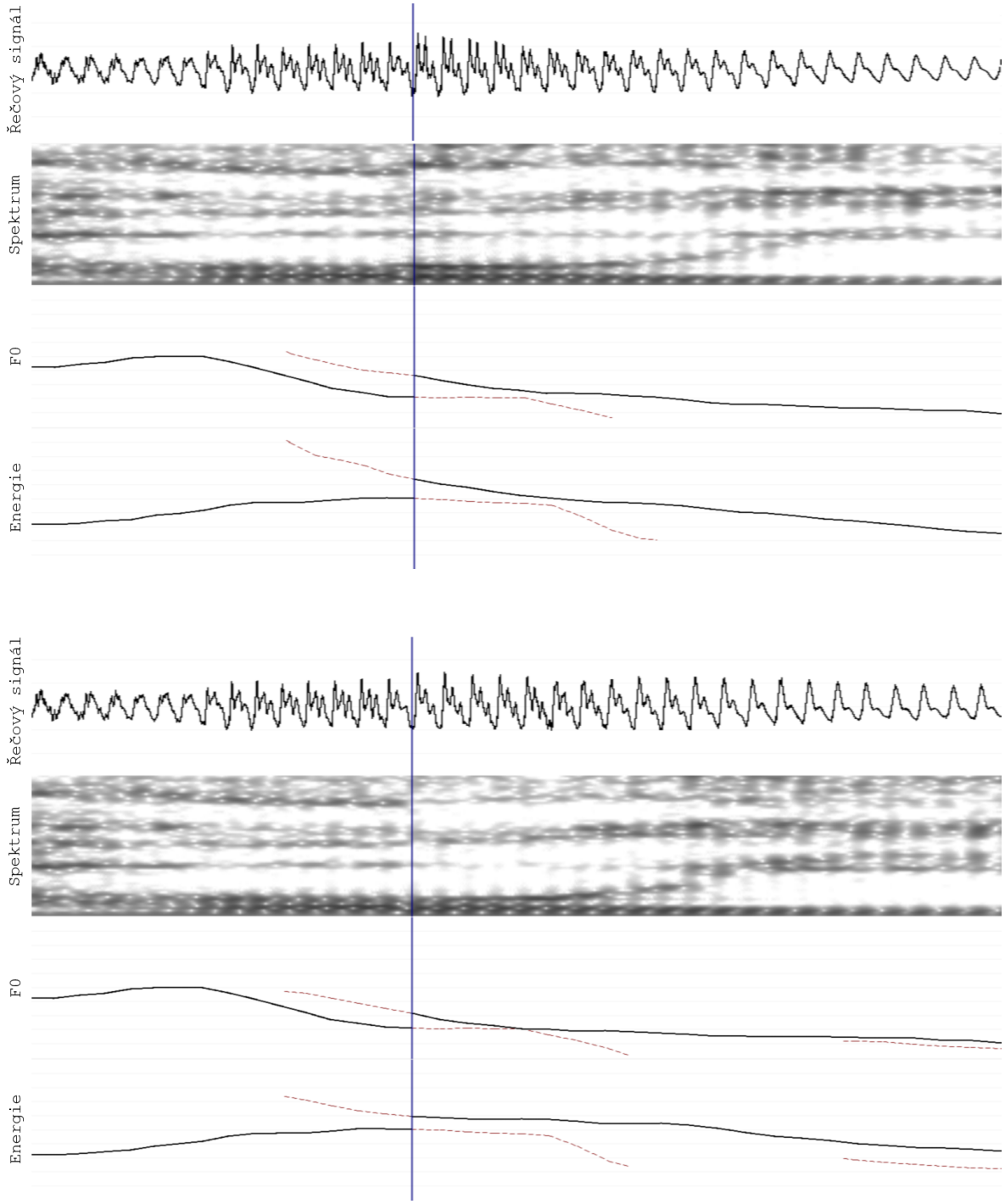
Opravit chyby lze pomocí signálových modifikací. Každá taková modifikace však snižuje kvalitu a přirozenost syntetické řeči. Je tedy otázka, zda by oprava ještě více nezhoršila problém. Lepším způsobem je (pokud je to možné) vybrat jinou sekvenci jednotek, které nebudou mít tak rozdílné frekvence $F0$.

Neadekvátní trvání či změna tempa

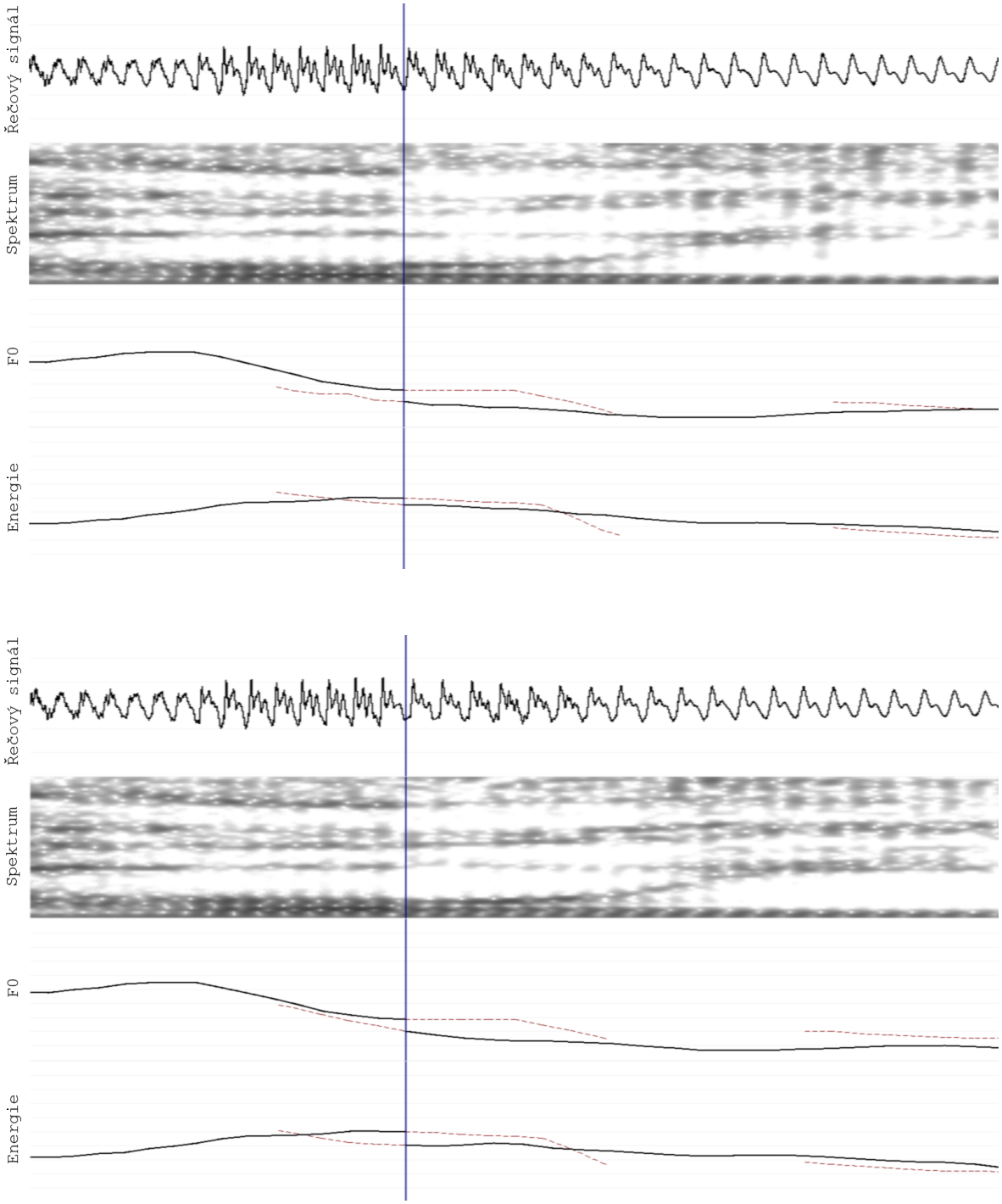
Jednotky pocházejí z různých vět a každá může být vyslovena s různým tempem. Pokud jsou takovéto jednotky na sebe napojeny, vzniká v syntetické řeči místo, kde se prudce změní rychlost výslovnosti věty. Tento jev je velmi slyšitelný a rušivý pro posluchače.

Tento problém se nevyskytuje tak často. Jeho opravu lze opět provést signálovou modifikací, ale opět platí nevýhody jako u nespojitosti v $F0$.

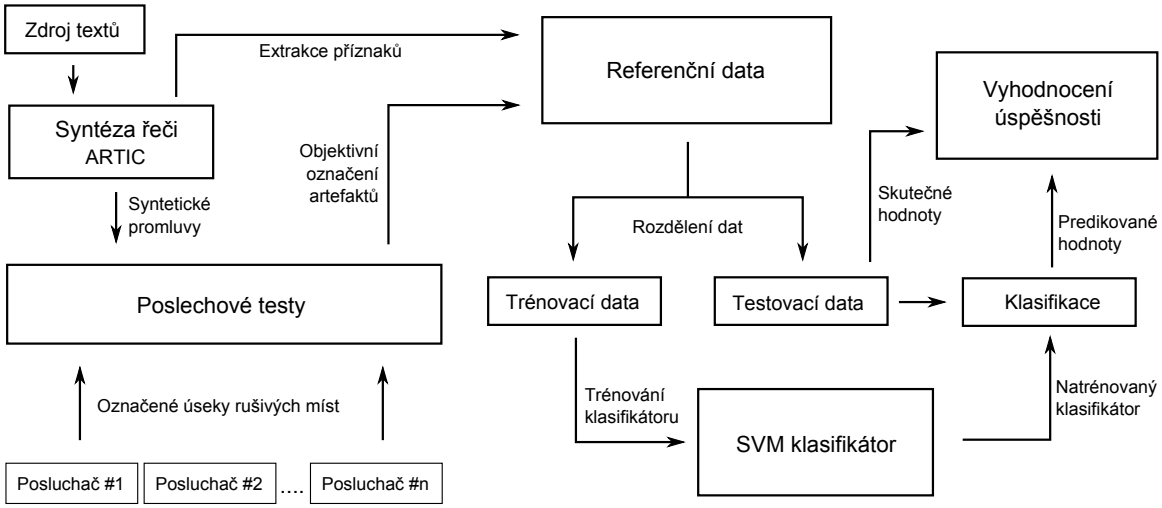
Parametry o délce trvání nejsou v systému ARTIC součástí hodnotících kritérií, podle kterých se vybírají jednotky. Tím, že se vybírají jednotky se shodným okolním kontextem a zejména se správnými pozičními parametry, se výskytu tohoto jevu částečně předchází.



Obrázek 4.2: Na obrázku lze pozorovat alternativy napojení dvou jednotek. Svislá čára znamená konkatenační bod. První jednotka je vždy stejná. Druhá se liší. Každá jednotka má různé průběhy audio signálu a $F0$. Spektrum se také mění. Lze sledovat velkou rozmanitost průběhů parametrů.



Obrázek 4.3: Další alternativy napojení dvou jednotek. Svislá čára znamená konkatenační bod. První jednotka je vždy stejná. Druhá se liší.



Obrázek 4.4: Schéma systému automatické detekce řečových artefaktů.

Nespojitost ve spektru

Opět tady platí pravidlo, že čím větší rozdíl je v místě konkatenace, tím je větší pravděpodobnost, že se artefakt vyskytne. Zde je ale situace komplikovanější, neboť u spektra se porovnává v případě použití MFCC koeficientů 12 hodnot. I když z principu jejich výpočtu budou nejdůležitější první hodnoty, neboť ty reprezentují základní tvar spektrální obálky.

O spektru částečně vypovídají i hodnoty formantů, kde největší roli hrají první čtyři. Problémem však je, že je obtížné jednoznačně určit jejich hodnoty.

Suprasegmentální problém

Pokud je artefakt způsoben něčím, co přesahuje jednu jednotku, hovoří se o suprasegmentální úrovni. Takový artefakt je těžké detekovat a tedy i opravovat, neboť vyžaduje pohled na sekvenci jednotek „s větším odstupem“.

Jev se například vyskytne, pokud dvě na sebe napojené zdrojové promluvy byly nahrány v různém nahrávacím sezení a celkově i s jiným „rozpoložením“ řečníka (jiná nálada či hlasová dispozice)[1]. Všechny parametry mohou na sebe navazovat, ale hlas má prostě jinou barvu. Ve výsledku jsou pak místa spojení rozpoznatelná tak, že se v nich najednou změní styl hlasu.

Tento problém přesahuje rámec práce. Vzhledem k tomu, že se vyskytuje docela často, měl by být i pro něj navrhnut nějaký způsob detekce a oprav. V práci je věnován důraz pouze na detekce lokálních nespojitostí.

4.3 Návrh systému automatické detekce artefaktů

Systém musí umět automaticky rozhodnout, zda konkrétní místo spojení obsahuje artefakt. Tento problém se nazývá klasifikace do dvou tříd. Systém automatické detekce artefaktů bude tedy realizován klasifikátorem. Ten pro každé „fyzické“ napojení jednotek v syntetické řeči dokáže rozhodnout, zda dané místo obsahuje řečový artefakt. Jeho vstupem bude vektor

[1] Z tohoto důvodu se často jako řečník volí profesionál (např. moderátor, herec).

příznaků, který bude spočítán z akustických a kontextových parametrů daného konkatenačního místa. Schéma návrhu systému automatické detekce řečových artefaktů je zobrazeno na obrázku 4.2.

Klasifikátor je trénován referenčními daty, ze kterých se naučí závislosti, podle kterých se bude později rozhodovat předloží-li se mu nová data. Tomuto způsobu se říká *učení s učitelem*. Klasifikátor tedy potřebuje mít k dispozici referenční data, která budou reprezentovat zástupce jak špatných konkatenačních míst, tak i těch dobrých.

Jak bylo ukázáno v předchozí části, příčin vzniku řečových artefaktů je mnoho. K tomu je nutné přidat fakt, že vnímání artefaktu je velice subjektivní záležitost. Tam, kde jeden posluchač slyší rušivý zvuk, druhý může slyšet normální spojení bez chyb. Každý má trochu jiné preference na to, co mu vadí.

K sestavení objektivních označení je nutné provést poslechový test s více posluchačů. Jedině tak lze vybrat objektivní označení artefaktů, které bude nezávislé na posluchači. Návrhu a realizaci poslechových testů se věnuje další kapitola. Samotnému vytvoření klasifikátoru a jeho natrénování se věnuje šestá kapitola, ve které byl takto navržený systém realizován a vyhodnocen.

Kapitola 5

Poslechové testy

5.1 Motivace

Jak již bylo zmíněno, rozhodnout, zda dané místo je artefakt, je velice obtížné. Každý člověk má jiné vnímání řeči a také jiné preference. Některému posluchači může přijít místo spojení naprosto normální, zatímco jiný ho shledá jako velmi rušivé.

Při vnímání řeči hraje také roli řada jiných faktorů. Například prostředí, či aktuální nálada posluchače. I samotný text poslouchaného projevu může ovlivňovat jeho postoj.

Sluchové vnímání řeči člověkem je složitý a komplikovaný proces. Mozek při něm provádí spoustu činností. Zajisté je použita i paměť. Slyšitelný artefakt si může člověk uvědomit například jen poté, co mu předcházel jiný rušivý element. Naproti tomu se mohou dva artefakty po sobě vyrušit nebo naopak zesílit. Nastavit pravidla pro predikci takovýchto složitých jevů by bylo velmi obtížné. Není ani jasné, jak velkou roli hrají jednotlivé signálové charakteristiky.

Má se za to, že velkou roli hraje průběh základní frekvence $F0$. Její absolutní rozdíl i její trend v okolí místa konkatenace, může hrát velký vliv pro vnímání řeči lidským mozkem. Spektrum signálu má dozajista také nemalý vliv. Kontextové okolí, trvání jednotky, průběhy formantů, či charakteristiky znělosti a další příznaky mohou hrát nějakou roli pro výsledný vjem posluchače. Tyto hypotézy bylo nutné ověřit.

Pro provádění experimentů však bylo nejdříve nutno získat reálná data, která by sloužila jako podklad a reference pro různé hypotézy. Místa v syntetické řeči, kde se nachází artefakt musela být jasně označená. Tato data musela být objektivní a nezávislá na konkrétním posluchači.

Proto byl navržen poslechový test. Ten si kladl za cíl získat právě taková referenční data. Tato kapitola popisuje realizaci těchto testů. Data z těchto testů byla poté použita pro experimenty popsané v další kapitole.

5.2 Realizace

Poslechový test byl realizován formou jednoduché webové stránky s přehledným a jednoduchým grafickým návrhem (ukázka na obrázku 5.3). Posluchači mohli pracovat z libovolného místa, většinou z domova. Pro zvýšení motivace byla pro aktivní posluchače slíbena finanční odměna. Pro co největší odrušení vnějších vlivů bylo vyžadováno používání sluchátek.

5.2.1 Implementace

Poslechové testy jsou naprogramovány v jazyce *PHP* a aplikace využívá *Nette framework*. Uložiště odpovědí běží nad databázovým serverem *MySQL*. Návrh grafického rozhraní staví na frameworku *Twitter Bootstrap*.

5.2.2 Zdroj dat

Při požadavku na novou větu byla vždy vybrána náhodně nová věta, tak aby splňovala několik kritérií. Jako bohatý zdroj pro náhodné věty sloužily novinové články z aktuálních témat vydávané na internetových stránkách českých deníků. Při nedostatku vět byl stažen aktuální archív článků a z nich byly vybrány věty, které splňovaly parametry pro testy.

Věta musela obsahovat pět až osm slov. Malý počet slov byl volen proto, aby bylo možné celou větu poslechnout a udržet v pozornosti celé její vyznění. Kvůli minimalizaci výskytu chyby fonetické transkripce a špatnému zpracování textu, byly vybrány pouze věty, jež obsahovaly jen česká slova. Číslovky a jiné větné útvary byly rovněž zakázány.

5.2.3 Příprava vět

Pokud věta prošla výběrem, byla převedena fonetickou transkripcí na sekvenci hlásek. K této větě byl rovněž vytvořen zvukový soubor, který obsahoval syntetickou řeč z textu dané věty[1]. Pro pozdější analýzu byly k větě rovněž uloženy veškeré parametry, jež byly použity v syntéze během algoritmu vybírání nejlepší sekvence kandidátů (unit selection). Z těchto parametrů bylo možné zpětně vyčíst hodnoty ceny cíle a spojení pro každou vybranou jednotku, navíc i hodnoty, ze kterých byly ceny vypočítány. Pro každou jednotku ze sekvence byly vyextrahovány parametry, jež později sloužily pro výpočet příznaků klasifikátoru. Pro výpočet příznaků klasifikátoru byly použity i další příznaky, které byly spočteny například z originálních vět řečového korpusu. Bližší pohled na výběr příznaků bude popsán v kapitole 6.

5.2.4 Návrh testu

V každé z testovaných vět mohl posluchač označit libovolný počet hlásek, jež se mu zdály rušivé či nezapadaly do celkového vyznění věty. Možnost označit větší úsek než je jedna hláska byla volena s ohledem na fakt, že označit přesné místo vzniku artefaktu může být v některých případech velmi obtížné. Vnímání chyb v syntéze řeči je silně ovlivněno okolím věty. Maximální velikost úseku byla volena pět fónů.

5.2.5 Výběr artefaktů

Všechna řetězení, která nebyla posluchačem označena, byla pro další výpočty uvažována jako bezchybná. Při syntéze řeči konkatenační syntézou jsou při výběru vybírány jednotky, které (mimo jiné) na sebe dobře navazují a dají se dobře spojovat. Jeden z kandidátů na takovou jednotku je její bezprostřední následovník. Pokud je posloupnost fónů stejná jako v originální větě, velmi často je vybrán celý úsek jednotek. Ten potom samozřejmě uvnitř neobsahuje žádná místa spojení, neboť signál na sebe přímo navazuje. V praktickém nasazení to pak

[1] Pro syntézu řeči byl použit hlas s interním označením „spkr_MR“ v revizi 81.

znamená, že místo fyzického spojení různých signálů se objevuje až po několika jednotkách. Některé věty mohou obsahovat velmi dlouhé úseky na sebe navazujících jednotek, oproti tomu některé mohou mít každou jednotku vybránu z jiné věty.

Výše uvedený fakt byl posluchačům zatajen. Zejména kvůli tomu, aby nenutil posluchače přemýšlet nad dalším omezením místa označení a také kvůli ověření, zda jsou jako místa chyb označována právě místa fyzických spojů. Posluchači mohli označovat libovolnou jednotku. Až teprve při vyhodnocování výsledků byla místa mimo fyzické spoje odstraněna.

5.2.6 Distribuce poslechových dat

Každému posluchači byly přidělovány náhodné věty v náhodném pořadí ze zdrojové databáze vět. Pro porovnání všech posluchačů mezi sebou bylo 200 vět (asi 10 %) označeno předem jako ověřovací. Ty byly v průběhu testů přiděleny všem posluchačům. Ostatní věty byly náhodně rozloženy mezi posluchače. Velikost databáze vět byla volena tak, aby průměrně každá věta byla přidělena mezi 50 % všech posluchačů. Tento postup byl zvolen pro to, aby bylo zpracováno co nejvíce vět. Zároveň aby byla každá věta vyhodnocena velkým počtem posluchačů a věrohodnost odpovědí byla vyšší. [2]

Do poslechového testu byli, pro zajištění variability, pozvaní jak lidé se zkušenostmi se syntézou řeči, tak zároveň lidé bez jakékoliv zkušenosti s uměle vytvářenou řečí. Někteří posluchači se testu zúčastnili pouze okrajově (tj. naposlouchali málo vět). Jejich odebráním z testovacích odpovědí by byla ztracena cenná data. Proto byli uživatelé zahrnuti do celkových výsledků, nicméně jejich odpovědím byla přikládána nižší váha.

5.3 Příprava dat pro klasifikátor

Data z poslechových testů vytvořily rozsáhlou databázi podezřelých míst v syntetické řeči. Před samotným trénováním klasifikátoru bylo nutné tato data patřičně roztřídit a profiltrovat. Jelikož každá věta mohla být poslouchána různou skupinou posluchačů, bylo nutné navrhnout vážící mechanismus, který zohlednil věrohodnost jednotlivých označených míst podle toho kolik a kým byla tato místa označena.

5.3.1 Struktura

Pro trénování klasifikátoru bylo nutné sestavit dvě množiny vzorků. Místa výskytu chyb (pozitivní vzorek) a naopak místa, kde se žádná slyšitelná chyba nevyskytuje (negativní vzorek)[3]. Tyto dvě množiny pak posloužili jako trénovací data pro klasifikátor. Jako vzorek byl uvažován vždy fyzický konkatenační bod tj. místo, kde dochází k řetězení signálů, které pochází z různých vět.

[2] V [5] byl rovněž použit poslechový test pro získání referenčních dat. Zde ale každý posluchač měl jiné věty. Autoři tak získali více odpovědí, ale neměli nástroj pro určení věrohodnosti jednotlivých označení.

[3] Označení je trochu matoucí. Instinktivně se nabízí označovat artefakty jako negativní vzorek, neboť reprezentují chybu v řečovém signálu, tj. „špatná místa“. Označení artefaktu jako pozitivní vzorek je zde kvůli zachování konvence, která se v úloze klasifikace používá. Pozitivní vzorek je označení těch vzorků, které mají být klasifikátorem označovány.

5.3.2 Algoritmus

Algoritmus výběru vzorků artefaktů lze popsat následujícím postupem (schéma je znázorněno na obrázku 5.1):

- Pro každou testovanou větu i byly vybrány všechna konkatenační místa j ve kterých se nacházel fyzický spoj různých audio signálů (tj. je zde napojení většinou dvou různých vět). Těmto místům bylo na začátku nastaveno skóre $s_{I,j} = 0$. Tato místa reprezentovala kandidáty na vzorek pro artefakt $a_{i,j}$.

- Skóre věrohodnosti cs_l každého posluchače l, jenž hodnotil větu i, bylo přičteno do skóre $s_{I,j} =$ a to tehdy, pokud daný posluchač označil místo j jako chybné.

- Hodnoty skóre byly potom normalizovány na dané větě v intervalu 0 až 1. Skóre $s_{I,j} = 1$ znamená, že všichni posluchači dané místo označili jako špatné.

- Pokud na dané větě pracovalo více posluchačů, kteří se shodli v určitém místě, má tato odpověď zajisté větší váhu než pokud by větu poslouchal pouze jeden uživatel. Pro zohlednění tohoto faktu bylo skóre věrohodnosti místa j vynásobeno následující hodnotou:

$$n_i = \sum_{l \in L_i} cs_l$$
$$ucs_i = 1 - x^{-n_i}$$

  Kde L_i je množina posluchačů, jenž vyhodnocovali větu i a n_i je součet věrohodností jednotlivých řečníků. Exponenciální funkce byla zvolena proto, aby pro vyšší počet už nestoupala tak rychlým tempem a zastavila se na hodnotě 1. Cíl tohoto návrhu byl zvýhodnit věty, kde odpovídalo více řečníků. S vyšším počtem však již tolik nestoupá důvěryhodnost odpovědí. Každý posluchač označuje trošku jiné chyby a tím rychle roste počet označených drobných chyb, které tím zakryjí jednu opravdu velkou. Parametr x ve vzorečku udává strmost růstu exponenciální funkce. V experimentu byla použita hodnota $x = 1.3$. Při tomto nastavení byla hodnota $ucs_i = 9$ právě tehdy, když větu poslouchal průměrný počet posluchařů (v tomto případě 9).

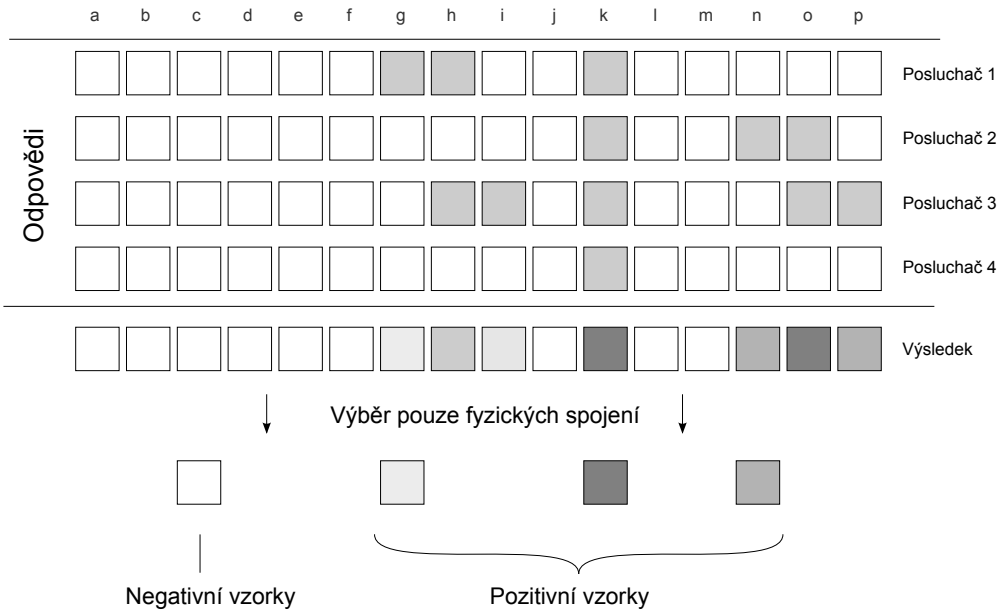
- Kandidáti $a_{i,j}$, který měly $s_{i,j} > 0$ byly uloženy do databáze pozitivních vzorků.

- Kandidáti, pro něž platilo $s_{i,j} = 0$, byly přidány do databáze negativních vzorků.

5.3.3 Nastavení vah

Databáze artefaktů byla následně použita pro trénování systému automatické detekce artefaktů. Pozitivní vzorky artefaktů sloužili jako zástupci míst, kde posluchači slyšeli rušivý element v syntetické řeči. Tato místa sloužila jako reference konkatenačních bodů, které měl systém detekce artefaktů eliminovat.

Naopak pozitivní vzorky sloužili jako referenční vzor pro bezchybná data. Tato místa by naopak klasifikátor označovat neměl, jelikož reprezentují čistý konkatenační bod a nevyvolávají v posluchačích rušivé jevy.



Obrázek 5.1: Schéma výběru vzorků artefaktů.

Každý artefakt v databázi měl uloženou svoji váhu, která byla vypočtena při algoritmu výběru artefaktů z poslechových dat. Jednotlivé vzorky byly sestupně seřazeny tak, aby při experimentech byly využívány ty artefakty s nejvyšší vahou.

Pokud se například v experimentu pracovalo jen se 100 vzorky, byly to právě ty vzorky s nejvyšší vahou. V tomto případě to znamená, že artefakty pocházely z vět, které poslouchalo hodně lidí a téměř všichni dané místo označili jako špatné. Naopak negativní vzorky byly takové, které v těchto větách ani jeden posluchač neoznačil.

5.4 Měření konzistence posluchačů

Poslechový test vyžaduje velkou pozornost posluchačů a jejich pečlivost. Občas se může vyskytnout případ, kdy posluchač své práci nevěnuje dostatečnou pozornost a v krajních případech například odpovídá náhodně, či uměle natahuje nebo manipuluje svou odpověď s vidinou snazší práce a tím i vyššího zisku.

V poslechových testech byla proto zavedena dodatečná opatření, která se snažila minimalizovat možnosti podvádění. Společně s tím zároveň odhalovala kvalitu jednotlivých posluchačů ve smyslu konzistence odpovědí.

5.4.1 Návrh měření

Přibližně každá 20. věta, která byla posluchači předložena, byla jím již jednou odpovězena. Byla mu tedy předložena podruhé. Posluchač nevěděl o tom, že na danou větu již odpovídal, neboť ta byla vybrána z minulosti tak, aby si jí pokud možno nepamatoval. Posluchači nebyli obeznámeni o těchto kontrolách. Společně s odeslanou odpovědí byla odeslána také informace o době strávené na dané stránce (kde stránka musela běžet v popředí operačního systému). Dále byl odeslán údaj o počtu přehrání věty nebo úseků. Tyto informace sloužili pro výpočet statistik.

Konzistence uživatelů byla získána porovnáním jejich odpovědí na stejné věty. Pro každého posluchače byla spočtena Cohenova kappa (viz. kapitola 2.5.1) pro věty, na které odpovídal vícekrát. Dle této míry bylo posluchači přiděleno skóre důvěry, které bylo později použito pro vážení odpovědí při trénování klasifikátoru.

5.4.2 Výsledky

Skóre důvěry se lišilo značně pro jednotlivé posluchače. Někteří jedinci dosahovali hodnoty důvěry, které se blížilo hodnotě $0,8$. Průměrná hodnota byla $0,58$. Každá věta byla v průměru přehrána 7krát a vyžadovala 26 sekund stráveného času.

Žádný posluchač nebyl vyloučen z poslechového testu z důvodu podvádění. Přesto však různí posluchači dosahovali různé míry konzistence. Měření skóre důvěry posluchače tedy mělo přínos v lepším popsání vstupních dat díky tomu, že odpovědi měly správně nastaveny váhy.

5.5 Míra shody posluchačů

Kromě měření konzistence uživatelů byl do systému poslechových testů zaveden způsob měření míry shody posluchačů. Ten umožňuje jedním číslem v rozmezí 0 až 1 vyjádřit shodu libovolných dvou posluchačů. Čím vyšší číslo, tím vyšší míra shody. Pokud by oba posluchači odpovídali identicky, byla by hodnota shody rovna jedné.

5.5.1 Návrh měření

Pro výpočet shody dvou posluchačů byla použita Cohenova kappa. Nejdříve byly vybrány věty, na něž odpovídali oba posluchači. Pro každý fyzický spoj konkatenace, který se nacházel v dané větě, byly srovnány odpovědi obou posluchačů (pokud označil úsek obsahující toto místo, byla odpověď pozitivní, jinak negativní). Takto se postupovalo pro všechny vybrané věty. Nasčítané hodnoty porovnaných odpovědí (shodná, pokud se oba vyjádřili negativně nebo oba pozitivně, různá, pokud se každý vyjádřil jinak) se dosadily do vzorečku výpočtu Cohenovy Kappy (viz. vzorec 2.7). Průměrná hodnota míry shody posluchačů byla 0,27, přičemž maximum bylo 0,43 a minimum 0,08.

5.5.2 Návrh upraveného měření

Jelikož lokalizace přesné pozice vzniku artefaktu je velmi obtížná (obzvláště když posluchač nezná umístění fyzických spojů konkatenace), byla navržena jiná metoda pro výpočet míry shody. Tato metoda nabídla mírnější a shovívavější způsob porovnávání shodné odpovědi. Odpovědi dvou posluchačů byly vzaty za stejné, pokud alespoň jeden dané místo označil jako chybné a druhý posluchač označil jakýkoliv hlásku v okolí 3 jako chybnou. V původním případě se okolí neuvažovalo. Dané místo mohlo tedy být označeno jen jedním uživatelem, a přesto byla odpověď obou posluchačů brána jako shoda. Pokud oba místo neoznačili, byla odpověď rovněž shodná, jelikož oba se vyjádřili negativně. Při použití této metody výpočtu se průměr Kappy zvedl na 0,53 s maximem rovno 0,69 a minimem 0,26. Je těžké říci, zda

Tabulka 5.1: Nejčastější hlásky, které byly označeny jako artefakt

Hláska	Počet výskytů	Relativní četnost
e	219	0,094
a	178	0,076
i	127	0,054
o	127	0,054
í	123	0,053
á	120	0,051
n	114	0,049
t	99	0,042

daný postup je správnější či jen pouze uměle zvyšuje hodnoty. Každopádně je patrné, že ne vždy posluchači označovali jako původ artefaktu přesně stejné místa.

5.5.3 Celková shoda posluchačů

V rámci experimentu byla spočtena i kappa Fleissova (viz. 2.5.2). Ta dokáže popsat míru shoda všech posluchačů najednou. Její hodnota vyjadřuje celkovou shodu. Výpočet této hodnoty znesnadňuje fakt, že každý posluchač označoval různé věty. Fleissova kappa totiž vyžaduje srovnávat odpovědi na stejných datech. Proto musel být výpočet omezen jen na věty, na které odpovídali všichni posluchači. To v tomto případě bylo kolem 10 % vět, které byly označeny jako ověřovací.

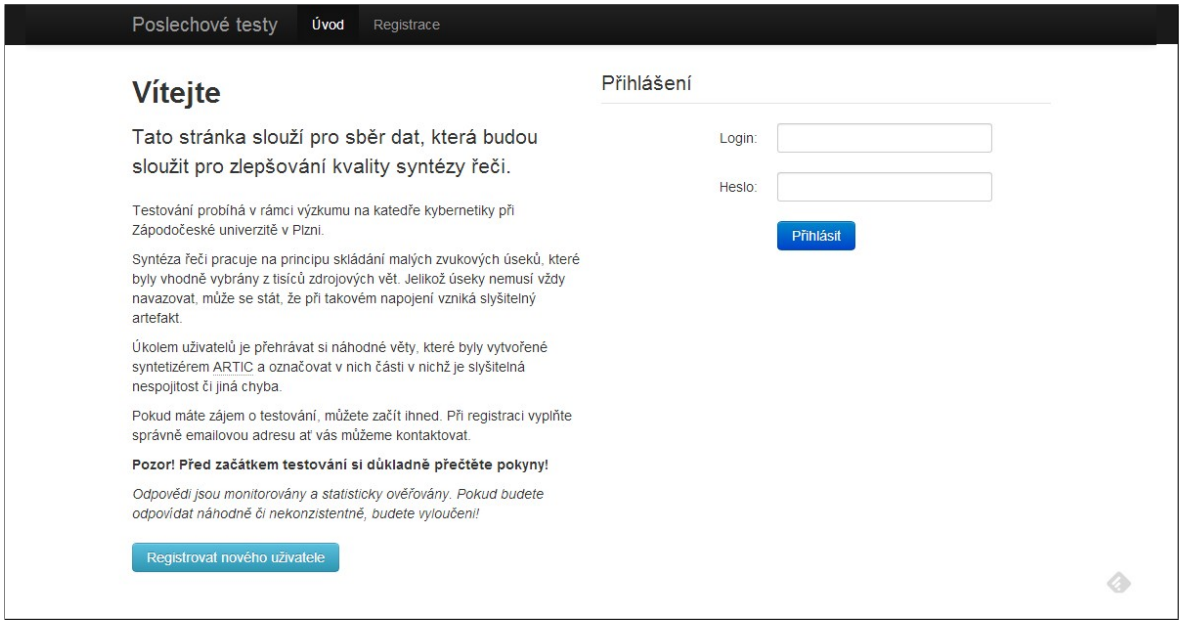
Fleissova kappa byla spočtena pro posluchače, kteří odpověděli na všechny ověřovací věty. V praxi to znamenalo lidi, kteří naposlouchali přibližně hodinu a více dat. Na 137 větách byla vypočtena hodnota Fleissovy Kappy 0,29. Nízká hodnota udává velkou variabilitu v odpovědích mezi uživateli a jen potvrzuje předpoklad o těžké lokalizaci a popsání artefaktů.

5.6 Výsledky poslechových testů

Celkově bylo pro poslechové testy použito 1900 vět. Posluchači odeslali 7200 odpovědí a označili více než 4700 podezřelých úseků. Nejpilnější posluchač odeslal 1250 vět. Nejkonzistentnější odpovědi měl posluchač, jehož Kappa na ověřovacích větách dosahovala 0,8. Rekord v nejdelší sekvenci odpovědí dosáhl posluchač, který naposlouchal 150 vět v kuse. Maximum odeslaných vět za jeden den bylo 290. Nejaktivnější hodina dne byla čas mezi jedenáctou a dvanáctou večerní hodinou.

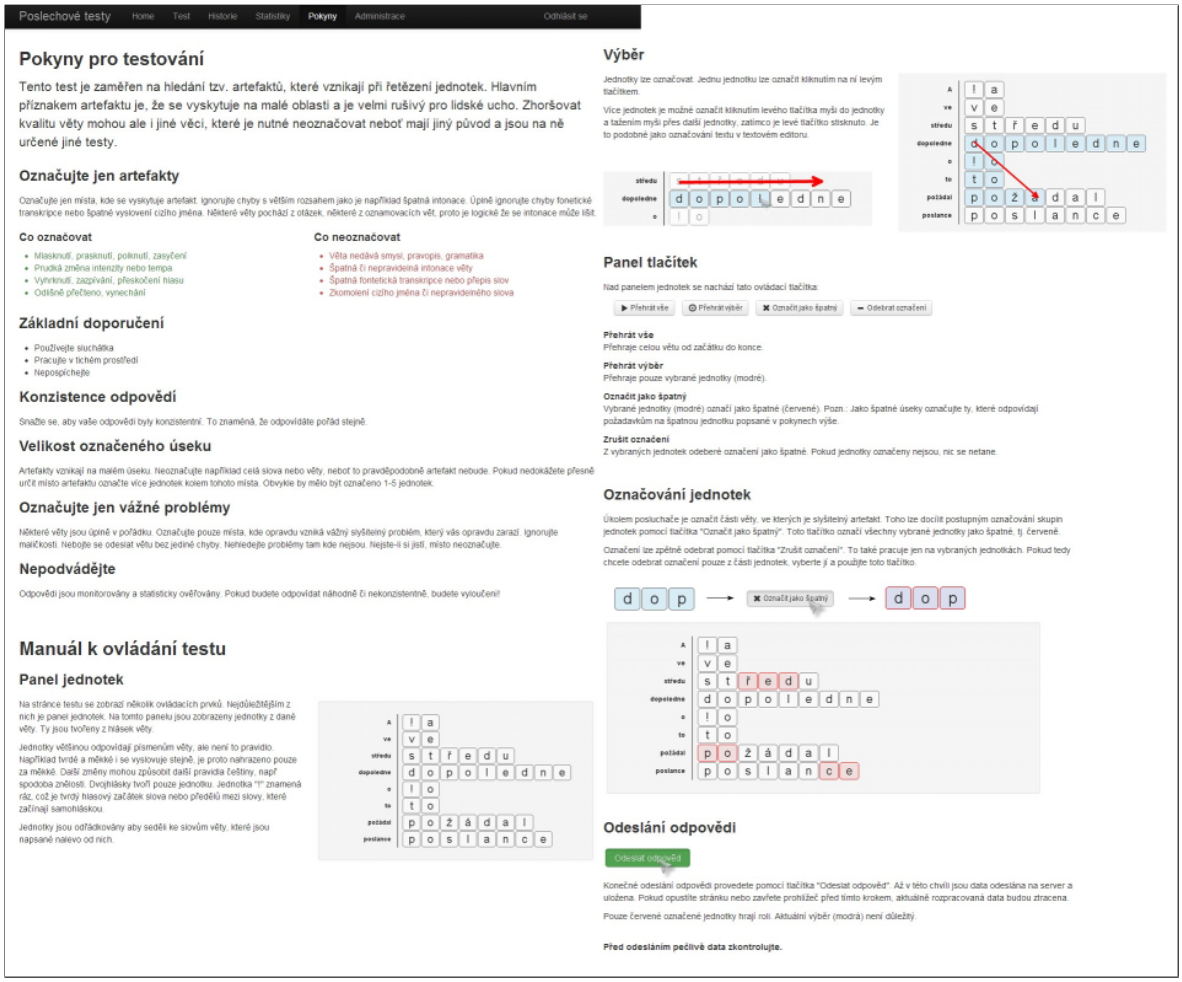
5.7 Grafické rozhraní

Následující obrázky ukazují nejdůležitější součásti webové aplikace pro poslechové testy.

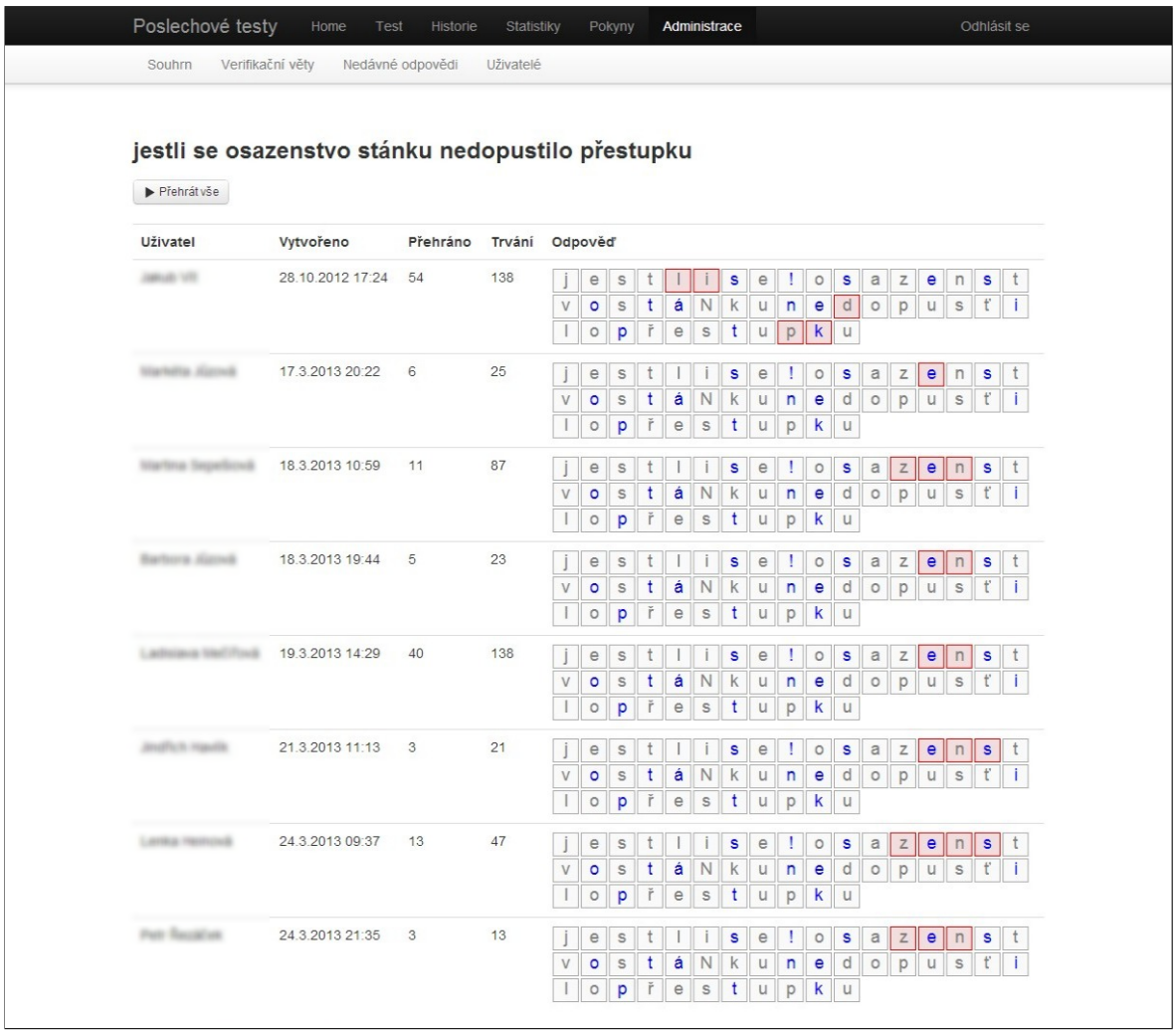


Obrázek 5.2: Úvodní obrazovka

Obrázek 5.3: Průběh testu - označování rušivých míst



Obrázek 5.4: Pokyny pro testování - dokumentace



Obrázek 5.5: Administrační pohled na všechny odpovědi pro danou větu

Kapitola 6

Vývoj systému detekce artefaktů

Posledním krokem pro vytvoření systému automatické detekce artefaktů v syntéze bylo natrénování klasifikátoru, který by chyby dokázal předpovídat. Jako trénovací data posloužily artefakty vybrané z výsledků poslechových testů.

6.1 Úloha

Klasifikační je úloha velmi standardní, jedná se to totiž o klasický problém klasifikace do dvou tříd. Tento typ klasifikace je nejběžnější typ. Jedná se o tzv. učení s učitelem, kde vstupem klasifikátoru jsou exempláře z obou požadovaných tříd a úkolem klasifikátoru je naučit se rozložení těchto tříd v rovině příznaků. Poté by měl být klasifikátor schopen pro nový vzorek určit, do jaké třídy patří.

V této úloze příznaky reprezentují akustické a kontextuální parametry v místě konkatenace. Z poslechových testů jsou již k dispozice trénovací data. Ty obsahují sadu označených dobrých a špatných konkatenačních míst ze syntetických promluv. Ke každému vzorku je k dispozici skóre důvěry. Podrobněji byl tento proces rozebrán v předchozí kapitole.

Úkolem klasifikátoru je dokázat pro nový vektor příznaků rozhodnout, zda daný konkatenační bod v syntetické větě bude znít rušivě či nikoliv. Při této znalosti pak není problém modifikovat systém syntézy řeči tak, aby tyto spojení nevybíral. Druhou možností je dané spojení vybrat ale provést spektrální modifikaci, která dané místo vyhladí. Využití systému je popsáno v další kapitole.

6.2 Trénování klasifikátoru

Připomeňme, že klasifikátor byl trénován pomocí dvou tříd vzorků. Pozitivní artefakty reprezentují špatné konkatenační spojení. Naproti tomu negativní vzorky udávají příklad toho, jak má vypadat bezchybné spojení.

6.2.1 Volba klasifikátoru

Jako klasifikátor byl zvolen SVM (viz. kapitola 2.3). Tento poměrně nový klasifikátor je v dnešní době jeden z nejpoužívanějších. Je používán díky své vysoké úspěšnosti. Jeho odol-

nost vůči přetrénování je v této úloze velmi vítána. Tento klasifikátor se používá v široké škále úloh, kde dosahuje dobrých výsledků.

Může se stát, že jsou používány příznaky, které nemusejí nutně přímo souviset s výskytem artefaktu. Vzhledem k omezené množině trénovacích dat je velké riziko, že klasifikátor se může přetrénovat na závislostech, jež přímo nesouvisí s úlohou. Odolnost vůči přetrénování je proto velmi podstatná.

6.2.2 Nastavení jádrové funkce

Klasifikátor SVM je podrobně popsán v kapitole 2. V experimentech spojených s naší úlohou byla jako jádrová funkce (Kernel) zvolena RBF (Radial Base Function). Tento kernel má pouze jeden parametr (γ), který musí být nastaven před samotným učením. Spolu s parametrem C je tak nutné hledat nejlepší kombinace jen dvou parametrů. Tento kernel je proto vhodný pro prvotní experimenty, což je i doporučeno v [3].

Pro učení klasifikátoru byl použit toolkit LIBSVM [1]. Tento balík nástrojů obsahuje vše potřebné pro následné učení klasifikátoru. Parametry nastavení jádrové transformace byly nastavovány pomocí 10-násobné cross validace.

6.2.3 Vážení dat

K nástroji LIBSVM existuje i jeho rozšířená verze, která navíc umožňuje každému vzorku trénovacích dat přiřadit jeho váhu. Ta určuje, jak moc velkou roli bude vzorek hrát v procesu učení. Klasifikátor se tak více soustředí na ty vzorky, které mají větší váhu. Díky tomu, že každý vzorek z poslechových testů již obsahuje váhu (v tomto případě skóre $s_{i,j}$ definované v kapitole 5.3.2), byla použita i tato rozšířená verze.

6.2.4 Výpočet příznaků

Ke každému vzorku artefaktu byl spočten vektor příznaků. Všechny příznaky byly převedeny na desetinné číslo a škálovány na intervalu <-1,1> . Některé příznaky v sobě nesou pouze logickou hodnotu ano-ne. V těchto případech byla hodnotě nepravda přiřazena -1 a hodnotě Pravda +1. Příznaky byly extrahovány z okolí místa spojení potenciálního artefaktu.

Pro vektor příznaků byly použity následující skupiny:

- **Statické akustické parametry:**
  - *Rozdíl frekvence F0* – Absolutní rozdíl hodnoty frekvence $F0$ v místě fyzického spojení. Frekvence $F0$ udává jakou má hlas v tomto místě výšku.
  - *Rozdíl energií* – Absolutní rozdíl hodnoty energie v místě fyzického spojení. Energie souvisí s vnímáním hlasitosti lidským uchem.
  - *Rozdíl spektra* – Absolutní rozdíl hodnot *melovských frekvenčních keprstrálních koeficientů* (MFCC) v místě fyzického spojení. Rozdíl je počítán jako Euklidovská vzdálenost. MFCC koeficienty v sobě nesou informace o spektru signálu.
  - *Poměr délky trvání* – Poměr délky trvání mezi aktuálním difónem a tím, který by se v tomto místě nacházel, kdyby věta nebyla rozdělena. Oba difóny reprezentují

stejnou jednotku, jen pochází z jiné věty a mají jiný akustický signál. Při stejné délce trvání je poměr $r = 1$.

- **Kontextuální parametry:**
  - *Charakteristika znělosti* – Tři binární hodnoty. Každá udává, zda je dané místo znělé či neznělé a to před místem spojení, v místě spojení a za místem spojení. Znělost signálu je určena dle toho, pokud se v něm vyskytuje základní frekvence $f0$, kterou vyvolává kmitání hlasivek. Neznělý signál je tvořen barevným šumem.
  - *Hranice slova* – Binární hodnota, která indikuje, zda je konkrétní místo spojení na hranici slova.
  - *Samohláska* – Tato binární hodnota udává, zda je daný fón samohláska či souhláska.
- **Dynamické parametry.** Následující parametry porovnávají rozdíly sledovaných atributů v čase celé hlásky. Rozdíly mezi jednotkou jsou počítány před a za, kde po konkatenaci zůstává vždy jen první půlka první jednotky a druhá půlka druhé jednotky. Hodnoty parametrů jsou počítány vždy jako absolutní rozdíl a jsou nasčítány do jednoho čísla:
  - *Suma rozdílů* $F0$ – Součet rozdílů hodnot $F0$.
  - *Suma rozdílů energií* – Součet rozdílů mezi energiemi.
  - *Suma rozdílů spekter* – Součet rozdílů prvních 4 formantů.
  - *Suma rozdílů znělostí* – Součet úseků signálů, kde se liší znělost.

  Příklad výpočtu rozdílu pro frekvenci $F0$ je předveden na obrázku 6.1

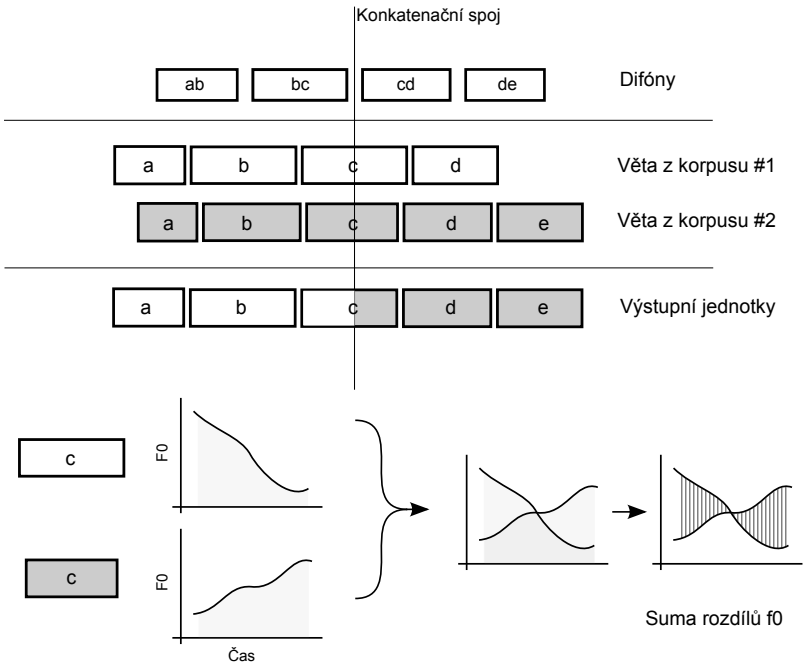
6.3 Experimenty

Při trénování klasifikátoru byly provedeny čtyři experimenty (EXP1, EXP2, EXP3, EXP4). Každý experiment měl jinak vybraná trénovací data. Přehled experimentů je shrnut v tabulce 6.1.

Ve všech experimentech byla použita vyvážená množina trénovacích dat. To znamená, že počet pozitivních vzorků (N_p) byl stejný jako počet negativních vzorků (N_n). Systém syntézy řeči ARTIC používá velmi rozsáhlý řečový korpus. Výsledná syntetická řeč je velmi kvalitní a neobsahuje mnoho artefaktů. Většina konkatenačních míst byla proto v pořádku. Negativních vzorků bylo proto méně než těch pozitivních. Proto bylo vybráno vždy jen N_p nejlepších negativních vzorků, aby trénovací množina byla vyvážená.

6.3.1 Nezávislost na okolí

Od samého začátku byl dán předpoklad, že řečový artefakt je způsoben vždy jedním konkatenačním místem a jeho okolí nemá žádný vliv na výsledek. Proto byly příznaky vytvářeny z parametrů z okolí místa spojení signálů, a byl brán jen malý ohled na kontext jednotky.



Obrázek 6.1: Ilustrace výpočtu sumy rozdílů $F0$.

Pro ověření této hypotézy byl proveden experiment. Klasifikátor byl trénován dvěma způsoby. Poprvé byl trénován se všemi vzorky, které byly k dispozici. V druhém případě však byly vyřazeny ty artefakty, které měly jako souseda rovněž artefakt nebo jen fyzický konkatenační spoj. Předpokládá se přitom, že pokud se v určité části věty nachází více artefaktů po sobě, pravděpodobně bude každý z nich přispívat k chybě konkrétního místa.

V druhém případě tedy byly použity jen ty artefakty, které měli kolem sebe klidné okolí. Trénování prvním způsobem bylo provedeno v experimentech EXP3 a EXP4. Zbylé dva experimenty (EXP1 a EXP2) byly trénovány pomocí druhého způsobu.

6.3.2 Použití vah

Trénování klasifikátoru pomocí vážených vzorků pracovalo s váhami, které přímo odpovídaly hodnotě skóre s_{ij}. Tato hodnota byla použita při výběru referenčních dat z poslechových testů (viz. kapitola 5.3.2). Při vyšším skóre by měli být artefakty více slyšitelné a více rušivé. Tím pádem by vyšší skóre mělo rovněž znamenat lepší rozpoznatelnost artefaktů (v prostoru příznaků by měly být dále od rozdělující nadroviny).

Pro ověření této hypotézy byl klasifikátor jednou trénován pomocí všech trénovacích vzorků. V druhém případě byly vybrány jen ty vzorky, které měli nejvyšší váhu. V těchto případech by se měl klasifikátor snáze učit, neboť vzorky by měly být snáze rozlišitelné.

Trénování s pouze omezeným počtem nejlepších zástupců bylo učiněno v experimentech EXP1 a EXP3. Naopak, v experimentech EXP2 a EXP4 nebyly v tomto směru data nijak vybírána.

Experiment	Výběr okolí	Sada vzorků
EXP1	klidné	jen ty nejlepší
EXP2	klidné	všechny
EXP3	libovolné	jen ty nejlepší
EXP4	libovolné	všechny

Tabulka 6.1: Přehled experimentů

6.3.3 Hodnocení při použití vah

Při hodnocení úspěšnosti klasifikátoru, jenž se trénoval s použitím váh, nastává problém. Ve standardním výpočtu přesnosti a úplnosti se totiž váhy nezohledňují.

Nastává tak situace, kdy se klasifikátor sice měl snažit více učit na vzorcích s větší vahou, při špatné klasifikaci je ale penalizace za každý vzorek počítáno stejně.

Při předpokladu, že klasifikátor se bez vah dokáže naučit nejlépe jak je možné, aby dosáhl nejlepší přesnosti, tak potom musí zvýhodnění některých vzorků (s velkou váhou) vést k minimálně stejnému snížení úspěšnosti na ostatních (s menší váhou). Celková úspěšnost tedy bude klesat. Nejlepší úspěšnosti by tak paradoxně klasifikátor získal, pokud by váhy zcela ignoroval.

Upravený výpočet

Při použití vážených trénovacích dat byl proto klasifikátor lehce upraven. V klasické verzi je při výpočtu četnosti shodných a neschodných odpovědí pro každý vzorek přičítáno číslo 1. Ve vážené verzi je příspěvkem každého vzorku jeho váha $s_{i,j}$. Porovnání výpočtu četnosti odpovědí *true positive* a *false positive* je znázorněno následujícím vztahem:

Nevážená verze	Vážená verze
$tp = \sum_{a_{i,j}} \begin{cases} 1 & y_{i,j}^{(\mathrm{t})} = 1 \wedge y_{i,j}^{(\mathrm{p})} = 1 \\ 0 & \text{jinak} \end{cases}$	$tp = \sum_{a_{i,j}} \begin{cases} s_{i,j} & y_{i,j}^{(\mathrm{t})} = 1 \wedge y_{i,j}^{(\mathrm{p})} = 1 \\ 0 & \text{jinak} \end{cases}$
$fp = \sum_{a_{i,j}} \begin{cases} 1 & y_{i,j}^{(\mathrm{t})} = 0 \wedge y_{i,j}^{(\mathrm{p})} = 1 \\ 0 & \text{jinak} \end{cases}$	$fp = \sum_{a_{i,j}} \begin{cases} s_{i,j} & y_{i,j}^{(\mathrm{t})} = 0 \wedge y_{i,j}^{(\mathrm{p})} = 1 \\ 0 & \text{jinak} \end{cases}$
$tn = \sum_{a_{i,j}} \begin{cases} 1 & y_{i,j}^{(\mathrm{t})} = 0 \wedge y_{i,j}^{(\mathrm{p})} = 0 \\ 0 & \text{jinak} \end{cases}$	$tn = \sum_{a_{i,j}} \begin{cases} s_{i,j} & y_{i,j}^{(\mathrm{t})} = 0 \wedge y_{i,j}^{(\mathrm{p})} = 0 \\ 0 & \text{jinak} \end{cases}$
$fn = \sum_{a_{i,j}} \begin{cases} 1 & y_{i,j}^{(\mathrm{t})} = 1 \wedge y_{i,j}^{(\mathrm{p})} = 0 \\ 0 & \text{jinak} \end{cases}$	$fn = \sum_{a_{i,j}} \begin{cases} s_{i,j} & y_{i,j}^{(\mathrm{t})} = 1 \wedge y_{i,j}^{(\mathrm{p})} = 0 \\ 0 & \text{jinak} \end{cases}$

Kde $y_{i,j}^{(\mathrm{p})}$ a $y_{i,j}^{(\mathrm{t})}$ vyjadřují predikovanou a skutečnou hodnotu artefaktu $a_{i,j}$ (přičemž $y(.) = 1$ znamená pozitivní vzorek artefaktu a $y(.) = 0$ znamená negativní vzorek).

6.4 Výsledky

V tabulce 6.2 je možno nalézt výsledky trénování klasifikátoru. Každý řádek odpovídá jednomu experimentu. Je třeba říci, že absolutní výsledky vážené a nevážené verze nelze přímo srovnávat.

Tabulka 6.2: Výsledky experimentů trénování klasifikátoru

	Počet vzorků				Nevážený SVM				Vážený SVM			
	N_p	N_n	$N_p^{(vše)}$	$N_n^{(vše)}$	R	P	$F1$	A	R	P	$F1$	A
EXP1	500	500	1574	3605	0.72	0.80	0.76	0.74	0.79	0.88	0.83	0.78
EXP2	1574	1574	1574	3605	0.63	0.80	0.71	0.67	0.80	0.95	0.87	0.79
EXP3	1000	1000	2458	4025	0.68	0.81	0.74	0.71	0.79	0.91	0.85	0.78
EXP4	2458	2458	2458	4025	0.61	0.76	0.68	0.64	0.79	0.95	0.86	0.79

První sloupcový blok obsahuje počty vzorků, které byly použity pro klasifikaci a počty vzorků, které byly k dispozici. Vzorky byly seřazeny dle svých vah, to znamená, že pokud bylo použito méně vzorků než bylo k dispozici, se jednalo o n nejlepších vzorků.

V druhém a třetím bloku jsou napsány samotné výsledky úspěšnosti klasifikace. Úplnost (R), přesnost (P), $F1$ hodnota a celková úspěšnost (A). Výsledky jsou rozděleny pro případ s použitím vážení a pro trénování bez použití vah.

Tyto výsledky ukazují, jak dobře je klasifikátor schopen se naučit od sebe odlišovat rušivá místa v syntéze řeči (artefakty) od míst, která jsou v pořádku. Úspěšnost zároveň udává, jak moc dobře je klasifikátor schopen predikovat tyto jevy v budoucnosti na neznámých datech.

Diskuze nad výsledky, vyhodnocení úspěšnosti experimentů a shrnutí celé práce je provedeno v následující závěrečné kapitole.

Kapitola 7

Závěr

V této práci byl představen systém automatické detekce řečových artefaktů. V úvodu práce byl čtenář seznámen se základními pojmy souvisejícími s problematikou této práce. Velký důraz byl kladen na úlohu syntézy řeči a na úlohu klasifikace. Byl zmíněn klasifikátor SVM, který byl v práci používán. Dále byl představen program pro analýzu syntetické řeči. Ten je přímo provázán se systémem syntézy řeči ARTIC. Pomocí něj bylo možné analyzovat libovolná místa v syntetické řeči, tedy i ta, co zní rušivě. Poté byla provedena analýza problému vzniku artefaktů. K tomu byly využity poznatky získané při používání vyvinutého programu. Byly sestaveny kategorie vzniku příčin řečových artefaktů. Každá kategorie byla popsána jak z hlediska příčiny, tak i z hlediska možné opravy. V této kapitole byl představen samotný systém detekce artefaktů a kroky nutné k jeho vytvoření. Před samotným trénováním klasifikátoru byla řešena úloha získání referenčních dat pro trénování systému detekce artefaktů. K tomu byly využity poslechové testy. Posluchači označovali místa v syntetických promluvách, kde slyšeli výskyt artefaktů. Byl představen algoritmus, jak z odpovědí posluchačů vybrat „objektivní“ označení artefaktů, kde každému označení byla přidělena váha důvěry. Ke konci práce byl natrénován SVM klasifikátor. Jako učící data posloužila objektivní označení z poslechových testů. Tento klasifikátor dokáže pro každý konkatenační spoj syntetické řeči rozhodnout, zda je místo v pořádku, či se v něm vyskytuje řečový artefakt. Při trénování byly provedeny čtyři experimenty. Jejich výsledky jsou zde shrnuty.

7.1 Zhodnocení výsledků

Při pohledu na výsledky (tabulka 6.2) je patrné, že bez použití vah je výhodnější použít pouze artefakty, které mají klidné okolí (experiment EXP1 oproti EXP3, EXP2 oproti EXP4). Tím lze cílit na artefakty, jejichž původ je přesně v dané jednotce a okolí nehraje vliv. Tím je potvrzen předpoklad, že některé artefakty mohou mít původ problému ve více jednotkách současně. Tyto artefakty je pak nemožné přesně zaměřit na jeden konkatenační bod jak klasifikátorem, tak i poslechovými testy. Tím jsou trénovací data pro klasifikátor znehodnocena vzorky, které, byť tak označeny, nemusejí nutně být artefakt.

Druhým poznatkem, který lze z výsledků nevážené verze klasifikátoru pozorovat, je znatelné zlepšení úspěšnosti při použití pouze těch vzorků, které mají vysoké skóre (EXP1 oproti EXP2, EXP3 oproti EXP4). To svědčí o tom, že skóre jsou nastavena správně, neboť vyšší

skóre znamená větší důvěru ve výskyt artefaktu. Jednotlivé třídy jsou pak reprezentovány pouze těmi vzorky, které do ní s největší pravděpodobností patří, a prostor příznaků je pak pro klasifikátor snáze rozlišitelný.

Vážená verze klasifikátoru naopak netrpí neduhy nevážené verze. Vysoká úspěšnost je zachována při všech experimentech. To naznačuje, že váhy byly nastaveny správně. Čím více bylo označení posluchače konkrétní (tj. označený úsek v syntetické větě byl menší), tím větší byla váha. Tím dostaly artefakty, kde bylo nejasné místo původu, menší váhu a méně tak ovlivňovali výsledky klasifikátoru. Rovněž výběr vzorků podle vah nemá smysl, neboť tyto váhy klasifikátor respektuje při učení. Méně významné vzorky mají z principu menší váhu a slouží jen jako doplňující informace. Z tohoto důvodu je pro váženou verzi nejlepší použít všechny vzorky. Klasifikátor pak zachovává vysokou úspěšnost a učící množina je velmi velká. Při správném nastavení vah se tak jeví použití vážené verze jako lepší volba, což dokazují i výsledky.

Při pohledu na tabulku 5.1 je zřejmé, že nejčastější hlásky, které jsou označovány jako artefakt, jsou samohlásky. To potvrzuje zjištění, že nejvíce artefaktů vzniká ve znělých úsecích. Velkou roli zde hraje průběh frekvence $F0$.

Výsledky učení klasifikátoru jsou velmi dobré, zvlášť u vážené verze. Klasifikátor byl učen na vyvážených datech (tj. stejný počet zástupců obou tříd). Pokud by klasifikátor artefakty detekoval náhodně, byla by úspěšnost 50 %. V nejlepším případě byla úspěšnost klasifikace 75 % resp. 80 % v případě vážené verze. To je velmi dobrý výsledek.

7.2 Návrhy na budoucí práci

V budoucí práci je možné dále rozšířit sadu příznaků, podle kterých klasifikátor rozhoduje, zda se v daném místě vyskytuje řečový artefakt. Jako další příznaky by bylo možné použít například syntaktický nebo i sémantický kontext v místě konkatenace. V další práci by rovněž bylo možné upravit klasifikátor tak, aby v syntetické řeči nepracoval jen na jediném konkatenačním bodu. Pokud by klasifikátor dokázal pracovat s větším okolím, bylo by možné detekovat artefakty, které mají suprasegmentální charakter (tj. jejich vznik ovlivňuje více po sobě jdoucích jednotek). Místo většího okolí by bylo možné použít vícefázový klasifikátor. V první fázi by klasifikátor každý konkatenační bod ohodnotil pravděpodobností, s jakou se v daném místě vyskytuje artefakt. Tato ohodnocení by pak byla vstupem druhého klasifikátoru, který by teprve rozhodl, ve kterých místech se nachází artefakt.

Důležitým krokem, který je potřeba provést, je napojení systému automatické detekce artefaktů do samotného systému syntézy řeči ARTIC. Jednou z možností je zakomponování systému přímo jako součást hodnotící funkce. Algoritmus výběru jednotek by tak přímo vybíral posloupnosti jednotek, v nichž se artefakty nevyskytují. Druhou možností je nejdříve provést výběr jednotek, a poté přidat druhý krok. Sekvence jednotek by byla v tomto kroku zkontrolována na výskyt artefaktů. Pokud by byl artefakt objeven, jednotka by mohla být příslušným způsobem opravena, například pomocí signálové modifikace nebo pomocí HMM syntézy (v odborné literatuře se tento přístup nazývá *hybridní syntéza* [4]).

Literatura

[1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technolog.*, 2(3):27:1–27:27, 2011.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Leaming*, 20(3):273–279, 1995.

[3] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification, 2000.

[4] Zhen-Hua Ling and Ren-Hua Wang. HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1245–IV–1248, 2007.

[5] Heng Lu, Si Wei, Lirong Dai, and Ren-Hua Wang. Automatic error detection for unit selection speech synthesis using log likelihood ratio based SVM classifier. In *Proc. INTERSPEECH*, pages 162–165, Makuhari, Japan, 2010.

[6] J. Matoušek, D. Tihelka, and J. Romportl. Current state of czech text-to-speech system artic. In *Text, Speech and Dialogue*, volume 4188 of *Lecture Notes in Computer Science*, pages 439–446. Springer, Berlin, Heidelberg, 2006.

[7] J. Matoušek, D. Tihelka, and L. Šmídl. On the impact of annotation errors on unit-selection speech synthesis. In *Text, Speech and Dialogue, Proceedings of the 15th International Conference TSD 2012*, volume 7499 of *Lecture Notes in Artificial Intelligence*, pages 456–463. Springer, Berlin-Heidelberg, Germany, 2012.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] J. Psutka, L. Müller, J. Matoušek, and V. Radová. *Mluvíme s počítačem česky.* Academia, Prague, 2006.

[10] Paul Taylor. *Text-to-Speech Synthesis.* Cambridge University Press, Cambridge, 2009.

[11] Daniel Tihelka, Jiří Kala, and Jindřich Matoušek. Enhancements of Viterbi search for fast unit selection synthesis. In *Proc. INTERSPEECH*, pages 174–177, Makuhari, Japan, 2010.