Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Detekce přízvuků v ruštině

PLZEŇ, 2019 ANASTASIIA CHIZHOVA

P R O H L Á Š E N Í

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

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

V Plzni dne ……

.....

podpis

Anotace

Tato bakalářská práce je věnovaná návrhu postupu automatické detekce přízvuku v ruském jazyce, která může být využita v rámci syntézy řeči. K detekci přízvuku jsou využity metody strojového učení: klasifikátory Logistic Regression a Support Vector Machine a jednoduchá neuronová síť. V bakalářské práci jsou popsány principy obecného syntetizéru, význam přízvuků ve světových jazycích a jejich specifika v ruském jazyce oproti ostatním. Byly prozkoumány obecné klasifikační principy a teoreticky analyzovány vybrané metody strojového učení. Ty byly následně aplikovány na připravená data. Výsledky efektivnosti vybraných metod byly zanalyzovány a porovnány mezi sebou.

Klíčová slova: počítačové zpracování řeči, syntéza řeči z textu, strojové učení, slovní přízvuk, detekce přízvuků, ruština, klasifikace

Abstract

This bachelor thesis dedicates to the design of automatic word stress detection process in Russian language that can be used in speech synthesis. Several machine learning methods are used to detect word stress: Logistic Regression and Support Vector Machine classifiers and a simple natural network. The thesis explains in whole the structure of a speech synthesizer. It also describes the role of word stress in world languages and its specifics in Russian language over the others. General classification principles were examined and selected methods of machine learning were theoretically analyzed. Afterwards, they were applied to prepared data. Then the results of the selected techniques for word stress detection were analyzed and compared with each other.

Keywords: computer speech processing, text–to–speech, machine learning, word stress, word stress detection, Russian language, classification

Obsah

Kapitola 1 Úvod ..... - 7 -
Kapitola 2 Syntéza řeči z textu ..... - 9 -
2.1 Úvod ..... - 9 -
2.2 Zpracování přirozeného jazyka ..... - 9 -
2.3 Analýza textu ..... - 10 -
2.3.1 Předzpracování textu ..... - 11 -
2.3.2 Normalizace textu ..... - 11 -
2.3.3 Morfologická analýza ..... - 12 -
2.3.4 Kontextová analýza ..... - 12 -
2.3.5 Syntakticko-prozodický rozbor ..... - 13 -
2.4 Fonetická transkripce ..... - 13 -
2.4.1 Fonetický slovník ..... - 14 -
2.4.2 Fonetická transkripční pravidla ..... - 14 -
2.4.3 Kombinovaný přístup ..... - 15 -
2.5 Generování prozódie ..... - 15 -
2.5.1 Intonace ..... - 15 -
2.5.2 Intenzita ..... - 16 -
2.5.3 Časování ..... - 16 -
2.5.3.1 Generování trvání (tempo) ..... - 16 -
2.5.3.2 Generování pauz ..... - 17 -
2.5.3.3 Generování přízvuku ..... - 17 -
2.6 Hodnocení kvality syntetické řeči ..... - 17 -
2.6.1 Testy srozumitelnosti ..... - 17 -
2.6.2 Testy přirozenosti ..... - 18 -
2.7 Aplikace syntézy řeči a systém TTS ..... - 19 -
Kapitola 3 Slovní přízvuk ..... - 21 -
3.1 Přízvuk jako vlastnost slova ..... - 21 -
3.2 Slovní přízvuk a jeho fyzikální vlastnosti ..... - 21 -
3.3 Trvání samohlásek ..... - 22 -
3.4 Různorodost přízvuku ..... - 23 -
3.5 Přízvuk pohyblivý a stálý ..... - 25 -
Kapitola 4 Úvod do strojového učení ..... - 27 -
4.1 Základy strojového učení ..... - 27 -
4.2 Základy klasifikačních přístupů ..... - 27 -
4.2 Metody strojového učení ..... - 29 -
4.2.1 Logistic Regression ..... - 29 -

4.2.2 Support Vector Machine ..... - 30 -
4.2.3 Neuronová síť ..... - 31 -
Kapitola 5 Praktická část ..... - 34 -
5.1 Příprava trénovacích dat ..... - 34 -
5.1.1 První fáze ..... - 34 -
5.1.2 Druhá fáze ..... - 36 -
5.1.3 Třetí fáze ..... - 38 -
5.2 Trénování a testování klasifikátorů ..... - 39 -
5.2.1 Metody vyhodnocení ..... - 40 -
5.3 Trénování a testování neuronové sítě ..... - 41 -
Kapitola 6 Výsledky ..... - 43 -
6.1 Výsledky trénování a testování na ručně připravených datech ..... - 43 -
6.2 Výsledky trénování na ručně připravených datech a testování na RNC ..... - 49 -
6.3 Výsledky zobecněného trénování ..... - 54 -
6.4 Shrnutí výsledků ..... - 59 -
Kapitola 7 Závěr ..... - 60 -
Literatura ..... - 61 -
Obsah přiloženého CD ..... - 63 -

Seznam obrázků

Obrázek 1: Schéma systému konverze textu na řeč (systém TTS)..... - 9 -
Obrázek 2 Morfologicko-syntaktický analyzátor systémů TTS..... - 11 -
Obrázek 3: Framework KKY ZČU k hodnocení přirozenosti pomocí MUSHRA testu..... - 18 -
Obrázek 4: Framework KKY ZČU k preferenčnímu testu..... - 19 -
Obrázek 5: Graf průměrné délky trvání samohlásek ruského a českého jazyka..... - 23 -
Obrázek 6: Ukázka umístění slovního přízvuku v češtině, francouzštině a polštině..... - 24 -
Obrázek 7: Blokové schéma fáze nastavování klasifikátoru..... - 28 -
Obrázek 8: Blokové schéma fáze klasifikace..... - 28 -
Obrázek 9: Model perceptronu..... - 32 -
Obrázek 10: Model vícevrstvé dopředné sítě..... - 32 -
Obrázek 11: Model sítě se zpětnou vazbou..... - 33 -
Obrázek 12: Ukázka souboru pro slovo „дома“..... - 36 -
Obrázek 13: Ukázka výstupního souboru pro slovo дома s délkami kontextu L-8, P-4..... - 37 -
Obrázek 14: Ukázka výstupního souboru pro slovo стороны s délkami kontextu L-20, P-5..... - 38 -
Obrázek 15: Ukázka mapování řetězců „дентом_страны” a „ентом_страны-“..... - 39 -
Obrázek 16 Ukázka rozdělení případů na řádky..... - 40 -
Obrázek 17: Ukázka rozdělení dat pomocí Leave-One-Out..... - 41 -
Obrázek 18: Průběh trénování neuronové sítě..... - 42 -
Obrázek 19: Graf průměrných výsledků pro všechna slova, při vzrůstu délky kontextu..... - 54 -
Obrázek 20 Graf průměrných výsledků obecného trénování přes všechny slova..... - 58 -

Seznam tabulek

Tabulka 1: Průměrná délka trvání samohlásek ruského a českého jazyka ..... - 23 -
Tabulka 2: Skloňování slov арбуз, блюдо, досуг, квартал ..... - 26 -
Tabulka 3: Skloňování slov города, учителя ..... - 26 -
Tabulka 4: Počet výskytů jednotlivých přízvuků v RNC ..... - 36 -
Tabulka 5: Vybrané délky kontextu ..... - 37 -
Tabulka 6: Výsledky klasifikace Logistic Regression, 100 případů ..... - 44 -
Tabulka 7: Výsledky klasifikace Logistic Regression, 200 případů ..... - 44 -
Tabulka 8: Průměrné výsledky klasifikace Logistic Regression, 100 a 200 případů ..... - 45 -
Tabulka 9: Výsledky klasifikace SVM kernel = linear, 100 případů ..... - 46 -
Tabulka 10: Výsledky klasifikace SVM kernel = linear, 200 případů ..... - 47 -
Tabulka 11: Průměrné výsledky klasifikace SVM kernel = linear, 100 a 200 případů ..... - 47 -
Tabulka 12: Výsledky klasifikace SVM kernel = rbf, 100 případů ..... - 48 -
Tabulka 13: Výsledky klasifikace SVM kernel = rbf, 200 případů ..... - 49 -
Tabulka 14: Průměrné výsledky klasifikace SVM kernel = rbf, 100 a 200 případů ..... - 49 -
Tabulka 15: Počet trénovacích a testovacích dat pro učení a vyhodnocování ..... - 50 -
Tabulka 16: Výsledky klasifikace Logistic Regression, testování na RNC ..... - 51 -
Tabulka 17 Výsledky klasifikace SVM kernel = linear, testování na RNC ..... - 52 -
Tabulka 18: Výsledky klasifikace SVM kernel = rbf, testování na RNC ..... - 52 -
Tabulka 19: Výsledky trénování neuronové sítě, testování na RNC ..... - 53 -
Tabulka 20: Průměrné výsledky pro všechna slova, testování na RNC ..... - 53 -
Tabulka 21: Výsledky klasifikace Logistic Regression, trénování na 9 slovech ..... - 55 -
Tabulka 22: Výsledky klasifikace SVM kernel = linear, trénování na 9 slovech ..... - 56 -
Tabulka 23: Výsledky klasifikace SVM kernel = rbf, trénování na 9 slovech ..... - 57 -
Tabulka 24: Výsledky trénování neuronové sítě, trénování na 9 slovech ..... - 57 -
Tabulka 25: Průměrné výsledky pro všechny slova ..... - 58 -

Kapitola 1

Úvod

Jacques Pesce napsal: „Všechno nové je dobře zapomenuté staré“. Moderní systémy umožňující syntézu řeči nejsou zásluhou 21. století. Již v 18. století byly učiněny první pokusy o vytvoření syntetizované promluvy. Na konci 18. století dánský vědec Christian Kratzenstein vytvořil model lidského řečového traktu, který dokázal produkovat pět dlouhých samohláskových zvuků (1). Jenže oproti němu jsme v dnešní době téměř schopni plně nahradit lidskou řeč. Syntéza řeči se stala hodně populárním a využívaným nástrojem, který pomáhá řešit a vylepšovat problémy v mnoha oblastech lidského života.

Proto se problém syntézy řeči řeší v mnoha jazycích. Pokud jste se alespoň jednou pokoušeli o učení se cizímu jazyku, nebude pro vás překvapením, že všechny jazyky a jejich vlastnosti se liší. Mnoho z nich je možné roztřídit do skupin s podobnými vlastnostmi, ale i přesto každý bude mít svoje pravidla tvoření slov a jejich vyslovování, která se postupně utvářela odedávna od vzniku daného jazyka. Z toho plyne, že každý jazyk má svoje specifika a výjimky, které je třeba řešit při návrhu systému syntézy řeči. Jak už bylo uvedeno, syntéza řeči je dostatečně využívaný nástroj, např. v oblasti medicíny, ve službách telekomunikací a monitorování, ve výuce cizích jazyků, v reklamě apod. Ve všech uvedených aplikacích je důležité, aby syntetizovaná řeč byla co nejpřirozenější a nejpochopitelnější, tj. vstupní text by měl být přečten správně. Tím se vysvětluje důležitost správné interpretace zadaného textu.

Existující systémy z pohledu pochopitelnosti už dosáhly dost vysoké úrovně. Ale z pohledu přirozenosti stále potřebují vylepšení a vyřešení na jednu stranu drobných, ale přitom důležitých jazykových problémů. Například v ruském jazyce je jednou z nejistot při syntéze řeči umístění přízvuku, který není oproti českému jazyku stálý. Pozice přízvučné slabiky, resp. samohlásky, není v psaném textu označena, výjimku tvoří učebnice ruského jazyka jako jazyka cizího. U většiny slov je přípustná pouze jedna varianta jeho umístění, a proto lze využít rozsáhlé slovníky (podrobně v kapitole 2.4.1). Některá slova ale připouští více variant umístění přízvuku v závislosti na tom, v jakém kontextu je slovo použito. Změna umístění přízvuku u velkého množství takových slov může způsobit částečnou nebo úplnou změnu významu slova (podrobně v kapitolách 3.4 a 3.5). Daný problém není možné vyřešit použitím statistik nebo jednoduchých pravidel.

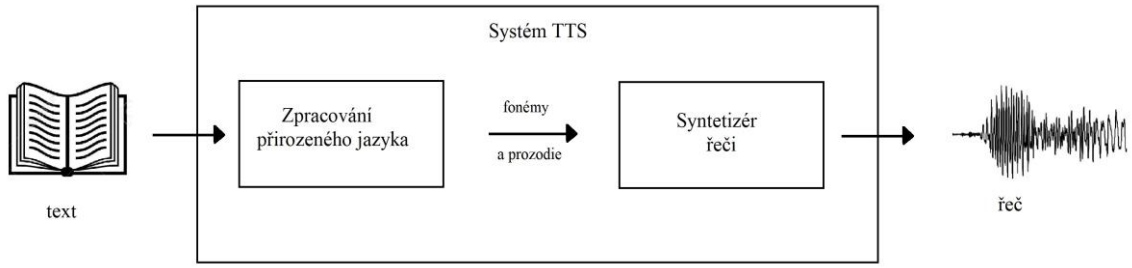
Proto je tato práce věnována automatizaci detekce přízvuku v ruském jazyce. Konkrétně se jedná o detekci přízvuku u homografů, což jsou slova, která mají stejnou grafickou, ale různou zvukovou reprezentaci. K pokusu o řešení existujícího problému bylo rozhodnuto použít metody strojového učení – klasifikátory a jednoduchou neuronovou síť. Při řešení daného problému byl použit pouze text, tj. vybraná okolní slova, protože nejčastějším vstupem do systému převodu textu do řeči je samotný text. Konečným výsledkem by potom měla být aplikace detekce přízvuku v celém systému a následné zlepšení úspěšnosti výsledné syntézy.

Kapitola 2

Syntéza řeči z textu

2.1 Úvod

Syntéza řeči z textu je nejobecnějším a nejkomplikovanějším problémem počítačové syntézy řeči. Úkolem je navrhnout systém, který by byl schopný automaticky převést libovolný psaný text na mluvenou řeč. Výsledná řeč by v nejlepším případě měla mít formu a kvalitu, jako by stejný text četl člověk s dobrou recitací. Cílem je generovat přirozenou řeč, jejíž poslech by neměl přitahovat pozornost. Hlavním úkolem je generovat libovolnou promluvu, což je mnohem složitější než resyntéza, tj. zkonstruování původní promluvy. Syntéza řeči (2) je proces umělého vytváření řeči. Vytváření řeči provádí zařízení - syntetizér řeči. Syntetizér hraje roli systému, který vytváří řeč na základě vstupní informace. V obecném případě syntézy řeči z textu (*angl*. *text-to-speech*, zkr. TTS) za vstupní informaci považujeme normální psaný text. Systém nejprve provádí analýzu textu a odhad fonetických a prozodických vlastností vytvářené řeči, pak na základě prostého textu může vytvořit odpovídající řeč. Tím se zabývá modul zpracování přirozeného jazyka (*angl*. *natural language processing*, zkr. NLP). Obecné schéma systému syntézy řeči z textu je znázorněné na obrázku 1:



Obrázek 1: Schéma systému konverze textu na řeč (systém TTS), převzato z (1)

2.2 Zpracování přirozeného jazyka

V případě, že nějaký text čte člověk, je schopný si díky svým znalostem a zkušenostem doplňovat detaily, které pomáhají ke správnému přečtení a interpretaci daného textu, např. intonace, pauzy mezi slovy, spojení, která tvoří některé předložky nebo sama slova mezi sebou. Předpokládá se,

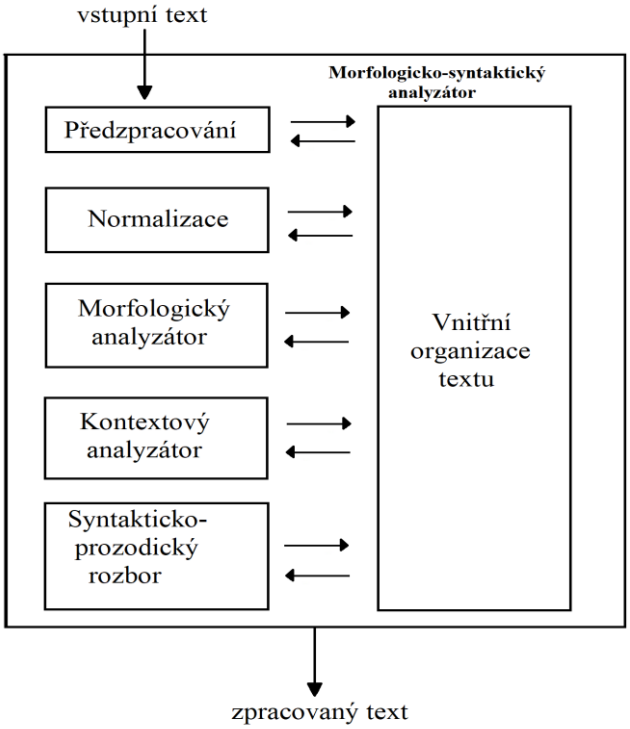
že člověk daný jazyk zná a umí psaný text interpretovat, protože samotný psaný jazyk je oproti mluvené řeči velmi zredukován. Například informační přenos zvuku zapsaného v CD kvalitě je okolo 60 kB/s, zatímco psaný text potřebuje jen 4kB/s (3). I když vezmeme v úvahu, že těch 60 kB/s obsahuje šumy, pozadí a další informace, které nejsou pro TTS podstatné, stále zbývá hodně informací, které psaný text neobsahuje, což dělá zpracování textu v systémech TTS komplikovaným úkolem.

Důležitou roli hraje fonetická transkripce textu, která v podstatě popisuje výslovnost slov, což jsou nejčastěji posloupnosti fonémů a prozodických značek. Modul zpracování přirozeného jazyka se skládá ze dvou bloků, modulu fonetické transkripce a generátoru prozódie. Ale ani jeden z těchto bloků nemusí být jednoznačný jen na základě psaného textu. Jedním z důvodů nejednoznačnosti výsledků činnosti modulu NLP je umístění přízvuků v jazycích, které nemají pevně danou pozici přízvuku, a tím se zabývá právě tato práce. Mezi jazyky, které nemají pevně umístěný přízvuk, patří i ruský jazyk a jedna z jeho vlastností, kterou je přízvuk, bude podrobně probrána v kapitole 3. Pro upřesnění fonetické transkripce je modul NLP také doplněn analyzátorem textu, který je určen k podrobné analýze vstupního textu (morfologické, kontextové nebo syntaktické informace) (4).

2.3 Analýza textu

Jak už bylo uvedeno, psaný text sám o sobě nese z pohledu systému TTS docela skromnou informaci. Proto je kladen velký důraz především na organizaci věty vyjádřenou vztahy mezi slovy, hlavně jejich závislostí na sousedních slovech. Zde lze jednu z hlavních pomůcek pro člověka při čtení, mluvnické kategorie jednotlivých slov (různé pády, rody, časy a čísla), využít i k navázání vztahů mezi slovy pro systém syntézy TTS.

Analyzátor textu je realizován morfologicko-syntaktickou analýzou (MSA) vstupního textu (5). MSA odhaduje vnitřní strukturu věty, tj. reprezentuje větu právě z pohledu syntaktických vazeb a vztahů mezi sousedními slovy a mluvnickými kategoriemi jednotlivých slov. Uvažujeme o MSA také kvůli prozódii, která je u přirozených vět velice závislá na syntaxi. V současných systémech TTS (znázorněno na obr. 2) komponenty analyzátoru běží paralelně. Tím pádem každá z komponent tedy může přidat nově zjištěnou informaci do zpracovávaného textu v průběhu paralelního zpracování ostatních komponent. Doplnění nové informace jednou komponentou pak může způsobit opětovné vyvolání příslušné komponenty, která na základě nových informací může upřesnit odhad některých vlastností zpracovávaného textu. Obecná struktura morfologicko-syntaktického analyzátoru je znázorněna na obrázku 2.



Obrázek 2 Morfologicko-syntaktický analyzátor systémů TTS, převzato z (3)

2.3.1 Předzpracování textu

Tento krok obvykle sjednocuje text, neboli provádí jeho unifikaci: určuje jeho typ (prostý text, HTML, XML, e-mail atd.), odstraňuje některé redundantní formátovací znaky, bílé znaky apod.

2.3.2 Normalizace textu

Normalizace textu také slouží ke sjednocení textu. Tento modul například přepisuje slova vstupního textu do plné slovní formy ve správné mluvnické kategorii. Jedná se o následující přepis:

- čísla:
  - číslovky: *20 zaměstnanců → dvacet zaměstnanců*
  - data: *1812 → osmnáct set dvanáct*
  - čas: *11:20* → např. *jedenáct hodin dvacet minut*
  - peníze: *800 Kč → osm set korun českých*
  - telefonní čísla apod.;
- zkratky:

- popř. → *popřípadě*;
- *SAE* → *Spojené arabské emiráty*

- akronymy:
  - *NASA* → *NASA*;
- symboly:
  - *%* → *procento, procenta, procent, atd.*

Obvykle je normalizace realizovaná pomocí pravidel a slovníků, nebo je možné využít výstupní informace z taggeru k přepsání zkratky do správného pádu atd.

2.3.3 Morfologická analýza

Z názvu je vidět, že morfologická analýza analyzuje slova s pohledu morfémů, kde morfém je nejmenší nedělitelná část slova, která sama o sobě dává smysl. Obecně lze definovat funkční slova[1] a významová[2] slova (4). Je zřejmé, že funkční slova tvoří určité ohraničené množství slov, která se jednoduše vejdou do slovníku. Oproti nim významová slova jsou flektivní kvůli morfémům, mezi které patří kmen slova, prefixy (předpony), sufixy (přípony) a koncovky. Tím pádem významová slova tvoří nekonečné množství slov. Při jejich analýze, většinou pomocí rozsáhlého slovníku, se detekuje základ slova či kmen a způsob utvoření daného slova. Jakékoli tvary slov jsou tvořeny morfémy. Morfologická analýza navrhuje všechny možné mluvnické kategorie každého slova věty izolovaně bez kontextu sousedních slov.

2.3.4 Kontextová analýza

Podle názvu se dá odvodit, že oproti předchozímu kroku se kontextová analýza (tagování, *angl. tagging*) zabývá analýzou jednotlivých slov v kontextu. Díky vazbám mezi slovy umožňuje kontextová analýza zredukovat množinu mluvnických kategorií slov z předchozího kroku na množství nejvíce pravděpodobných slov. Při tomto postupu existují dva typy metod: pravděpodobnostní a deterministické (nepravděpodobnostní) metody. Pravděpodobnostní metody jsou založeny na přechodových pravděpodobnostech mezi sousedními mluvnickými kategoriemi dvou slov. Mohou se počítat explicitně, pomocí n-gramů, nebo implicitně, pomocí neuronových sítí. Druhý typ metod je založen na klasifikačních a regresních stromech (*angl. classification and*

[1] také se jim říká slova neohebná: předložky, spojky atd.
[2] ohebná slova - samostatná slova, která mají význam

regression tree, zkr. CART). Využívají rozhodovací pravidla typu ano/ne, tím se „rozvětví“ - přijmou nebo odmítnou určitou kombinaci syntaktických kategorií.

2.3.5 Syntakticko-prozodický rozbor

Ve větě zpracovávané v předchozích krocích se hledají úseky nebo fráze podle prozodické realizace věty[3].

Nejjednodušší, a proto nejčastěji používanou metodou je ta, při které se úseky nebo fráze odlišují podle pozic interpunkce (čárek, středníků, pomlček, závorek apod.). Další skupinou metod jsou metody používající gramatiky: rozšířené bezkontextové a pravděpodobnostní gramatiky. Dále se také využívají korpusově orientované metody, využívající toho, že hranice mezi frázemi v korpusu jsou již označeny a je tedy umožněno automatické odvození heuristických pravidel. Modelování hranic mezi frázemi je realizováno skrytými Markovovými modely (*angl. Hidden Markov Model*, HMM), klasifikačními a regresními stromy nebo i neuronovými sítěmi.

2.4 Fonetická transkripce

Fonetická transkripce je určena k popisu výslovnosti řeči tak, jak to dělá člověk. Nejčastěji za předpokladu, že mluví spisovně. Fonetickou transkripci lze provádět ručně nebo automaticky. Fonetická transkripce, která se provádí ručně, probíhá na základě textu nebo přímo z řečového signálu promluvy tak, že člověk přepisuje text podle určitých pravidel do fonémů.

Můžeme uvést několik příkladů v ruštině:

- Klasický pozdrav „Здравствуйте“ (*čes. Dobrý den*) se vysloví trochu jinak, než se píše. Píšeme „zdrAvstvujTe“[4], vyslovujeme „zdrAstvujTe“

- Slovo „солнце“ (*čes. slunce*) je také dobrý příklad. Zde se písmeno „л“ nevyslovuje a fonetická transkripce bude zredukována z psaného „sOlnce“na „sOnce“.

Daný proces ale je, jako jakýkoli jiný proces zpracování velkého množství dat, pracný a časově náročný. Co se týče automatického vytváření fonetické transkripce, provádí se bez přímé účasti experta, a to může být realizováno třemi způsoby. Jako vstup do modulu fonetické transkripce slouží text, předem zpracovaný modulem MSA, tzn. je doplněn o morfologické, kontextové a syntakticko-prozodické informace. Pokud některá z těchto informací chybí, zpracovávaný text

[3] pauzy nebo naopak akcenty v určitých částech věty v závislosti na jejím smyslu, kontextu fráze apod.
[4] při přepisu do fonetické transkripce ruštiny se velkými písmeny vyznačují přízvučné samohlásky a změkčené hlásky

je z pohledu automatické fonetické transkripce nejednoznačný, a tudíž nemůžeme očekávat přesnou fonetickou transkripci.

2.4.1 Fonetický slovník

Přepis na základě slovníku tedy vypadá tak, že slovník obsahuje ortografickou (psanou, textovou) a fonetickou (výslovnostní) formu slova. Jeho použití je však vhodné spíše pro analytické jazyky, jako např. angličtinu, protože ta neobsahuje velké množství mluvnických kategorií jednoho slova. Oxford English Dictionary uvádí kolem 300 tisíc hlavních slov a k nim ještě 320 tisíc odvozených (včetně staroanglických slov). Oproti tomu základ češtiny také činí 300 tisíc slov, ale jenom od jednoho podstatného jména lze odvodit až 14 dalších slovních tvarů skloňováním v jednotném a množném čísle, podobné to je i se slovesy a přídavnými jmény. Kvůli tomu by měl kompletní slovník miliony položek. Moderní slovníky ruského jazyka obsahují v průměru 150 tisíc slov (6), která tvoří tvary stejným způsobem jako čeština, a některé zdroje uvádějí, že rozsah kompletního slovníku by činil téměř 3 miliony slov.

Přitom v knize O češtině v číslech (7) se uvádí, že průměrný Čech používá slovní zásobu o několika tisících slovech. Filologové – lidé, kteří pracují hodně se slovy – používají skoro 10 tisíc slov. To je tzv. aktivní slovní zásoba, tedy počet slov běžně používaných člověkem. Pasivní slovní zásoba, jež obsahuje slova, která známe, ale nepoužíváme je často, je tři- až šestkrát větší než aktivní. Z toho lze usoudit, že tří set tisícový slovník, který umí odvozovat slova i pomocí morfémů, je postačující k fonetickému přepisu téměř jakéhokoliv slova.

2.4.2 Fonetická transkripční pravidla

Pokud chceme provést fonetickou transkripci pro flektivní jazyky, takové jako jsou čeština, ruština a další slovanské jazyky, ve kterých existuje mnoho tvarů odvozených od stejného slova (např. podstatného jména), nemůže být využití fonetického slovníku efektivní. V takovém případě je efektivnější najít obecná pravidla toho, jak spolupracují fonémy mezi sebou, a podle toho lze provádět fonetický přepis automaticky. Pravidla obvyke navrhuje jazykový expert. Například spojení **di**, **ti**, **ni** se přepisují na [**d'i**], [**t'i**], [**ňi**] (8):

d→d' pokud následuje i nebo í – dítě [**d'ít'e**]

t→t' pokud následuje i nebo í – ticho [**t'icho**]

n→ň pokud následuje i nebo í – nikdy [**ňigdi**].

Pro přepis potom lze využít expertní systémy nebo metody strojového učení, které jsou na základě dostatečného množství trénovacích dat schopny automatického hledání pravidel. Mezi tyto systémy patří klasifikátory, neuronové a hluboké neuronové sítě.

2.4.3 Kombinovaný přístup

Nejčastěji při vytváření fonetické transkripce bude nejprve slovo hledáno ve slovníku, potom, pokud slovo není nalezeno, se použijí transkripční pravidla. Například v češtině by v případě slov cizího původu, jako matematika či kybernetika, bylo pomocí transkripčního pravidla uvedeného v bodě 2.4.2 spojení **ti** přepsáno na [**t'i**], což správně ale není – [**matematika**], [**kibernetika**][5]. Proto je vhodné v první řadě hledat příslušnou transkripci ve slovníku.

2.5 Generování prozódie

Ve slovníku literárních termínů S. Bobrova (9) je termín prozódie vysvětlen takto:

„Prozódie zahrnuje učení o zvuku, slabice, přízvuků, délce slabiky, intonace a pauzy, tedy o rytmickém materiálu verše mimo speciálních rytmických pododdílů... Do oboru prozódie patří: fonémy, kvalita zvuku, akustika fonémů, jejich genetika, akustický efekt, akustický rozdíl prózy, poezie a zpěvu, druhy přízvuků (lexikologické, syntaktické, frazeologické), rytmické segmenty, formy rytmu ve verších a próze, vztahy mezi délkou a přízvukem, prodloužením, výška fonémů, přízvuk a intonace, vyjádření přízvuků, délky a výšky."

Srozumitelnost a přirozenost řeči je přímo závislá na prozodických charakteristikách řeči a zejména na intonaci, rytmu a intenzitě. Ke generování prozódie patří i správnost přízvuku – lingvistická úroveň reprezentace prozódie (4), čemuž je věnovaná právě tato práce.

2.5.1 Intonace

Intonace promluvy souvisí s průběhem frekvence základního hlasivkového tónu nebo výškou hlasu. Generování intonace probíhá tak, že se nejdříve vytvoří symbolický popis intonace jako doplněk k ortografické a fonetické transkripci promluvy. Symbolický popis prozódie je charakteristický prostředek, jak lze nejlépe a nejpřesněji popsat průběh prozódie (a zejména intonace) dané promluvy. Nejčastěji se přitom omezuje na popis „význačných" událostí, hlavně na přízvuky a koncové tóny, např. pokles intonace na konci oznamovací věty. Pak zvolený intonační model generuje na základě tohoto popisu výslednou intonační konturu (kadenci) (5).

[5] Čte se tvrdě, tj. ‚i' nezměkčuje ‚t' před sebou

„Klasickým pojmem intonačního popisu je tzv. *intonační kadence* (*tone-pattern*, *tune-pattern*, *pitch configuration*, *intonation pattern*). Je to nejmenší abstraktní melodické schéma, které v souvislém průběhu intonace při popisu vyčleňujeme. Pro stanovení kadencí je podstatnou vlastností jejich zvuková jednoduchost a způsob, jak v intonačním průběhu určujeme některou z jejich hranic.“ (10)

Použitý symbolický popis prozódie pak určuje intonační modely vhodné pro generování intonace. Intonační modely poté generují výsledný průběh základního hlasivkového tónu. Nejpoužívanějšími intonačními modely jsou akustické modely. Tyto modely vycházejí z akustické reprezentace intonace pomocí průběhu základního hlasivkového tónu v čase. Používají se například Fujisakiho model, neuronové sítě a fuzzy systémy. V současných systémech TTS se používají také percepční a lingvistické modely intonace. Z hlediska samotného generování kontury se intonační modely rozlišují na generované podle pravidel, parametrické modely intonace a korpusově orientované generování intonace.

2.5.2 Intenzita

Tato charakteristika zpravidla nejméně ovlivňuje význam promluvy z uvedených třech charakteristik a spíše hraje roli emocionálního zabarvení promluvy řečníka. Při generování intenzity se pracuje především s hláskami (popř. fonémy), to jsou základní a nejmenší jednotky. K určení jejich intenzity je třeba mít rozsáhlé řečové korpusy, ze kterých se obvykle počítají statistiky. Často se pro generování intenzity používají i další specifické informace, jako pozice hlásky ve slově, větném úseku nebo ve větě, přízvučnost, popř. nepřízvučnost, apod.

2.5.3 Časování

Obecně v systémech TTS časování implikuje veškeré informace o pauzách, rytmu a tempu, které jsou důležité pro to, aby řeč zněla přirozeně.

2.5.3.1 Generování trvání (tempo)

Stejně jako při generování intenzity se i tady pracuje především se segmenty na úrovni fonémů. A opět, jako i v případě generování intenzity, modely trvání využívají znalostí o artikulačních a fonologických aspektech těchto segmentů pomocí pravidel nebo statistik vypočtených na základě rozsáhlých řečových korpusů.

2.5.3.2 Generování pauz

Při generování pauz se umísťují pauzy na hranice mezi některými frázemi, pauzy přitom mohou mít různé trvání.

2.5.3.3 Generování přízvuku

V systémech TTS se zatím bere v úvahu pouze slovní přízvuk. K určení větného přízvuku by bylo nutné znát význam věty. Přízvuk je způsoben změnou třech základních charakteristik – intonace nebo frekvence základního hlasivkového tónu, trvání i intenzity. Cílem je určit pro každé slovo přízvučnou slabiku. U jazyků s pevným přízvukem, mezi které patří i čeština (přízvuk se nachází vždy na první slabice taktu), jde o relativně jednoduchou úlohu, jiné jazyky jako například angličtina a ruština už představují komplikovanější úkol. V ruštině přízvuk přímo souvisí s přítomností samohlásky. Slova, která ji neobsahují (většinou se jedná o jednopísmenné předložky), jsou z hlediska přízvuku připojená k sousedním slovům, která ji obsahují. Z tohoto hlediska tedy můžeme zaměnit pojem "přízvučná slabika" za " přízvučná samohláska".

2.6 Hodnocení kvality syntetické řeči

Ohodnotit kvalitu rekonstruované řeči je teoreticky možné pomocí poslechových testů (5). V současné době ale objektivní testy neexistují, protože nejsme schopni objektivně vnímat kvalitu rekonstruované řeči. Není úplně jasné, jak kvalitu hodnotit, neboť u krátkých slovních spojení bude kvalita zpravidla lepší. Čím delší syntetizovaná věta je, tím více se slova navzájem ovlivňují a věta zní neutrálně. To se stává i v případě, kdy by měla věta vyjadřovat nějaké emoce. Při poslechových testech však můžeme subjektivnost vykompenzovat velkým počtem testovaných posluchačů. Při hodnocení syntetické řeči je hlavní důraz kladen na srozumitelnost a přirozenost. Testy také slouží k tomu, aby bylo možné detekovat a následně odstraňovat chyby, kterých se může dopustit syntetizér.

2.6.1 Testy srozumitelnosti

Testy srozumitelnosti slouží k určení toho, jak dobře posluchači rozumějí syntetické řeči. Rozlišují se testy na úrovní slov nebo celých vět.

K nejpoužívanějším testům srozumitelnosti na úrovni slov patří test diagnostikou rýmu (*angl. Diagnostic Rhyme Test*, zkr. DRT) a test modifikací rýmu (*angl. Modifie Rhyme Test*, zkr. MRT). MRT testy probíhají tak, že se posluchačům přehrává slovo po slově a jejich úkolem je stanovit

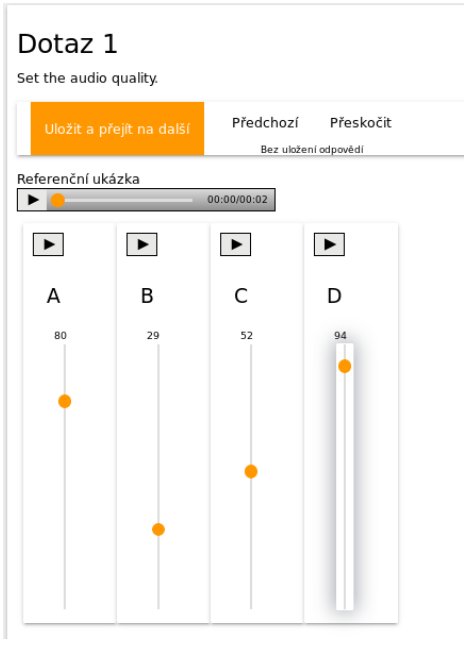
každé slovo ve skupině slov jemu podobných (jsou předem dané v dotazníku). Výhodami těchto testů jsou spolehlivost a malý počet „nekvalifikovaných“ posluchačů.

Testy na úrovni celých vět zastupuje například test sémanticky nepredikovatelných vět (*angl. Semantically Unpredictable Sentences*, zkr. SUS). Sémanticky nepredikovatelné věty jsou věty, které jsou gramaticky správné, ale nedávají smysl. Tím pádem kontextová a sémantická informace je zredukována, čímž je proces rozpoznání zkomplikován. Tudíž pro posluchače není možné si neznámé slovo „domyslet“. Úkolem posluchačů je napsat přesně to, co slyší.

2.6.2 Testy přirozenosti

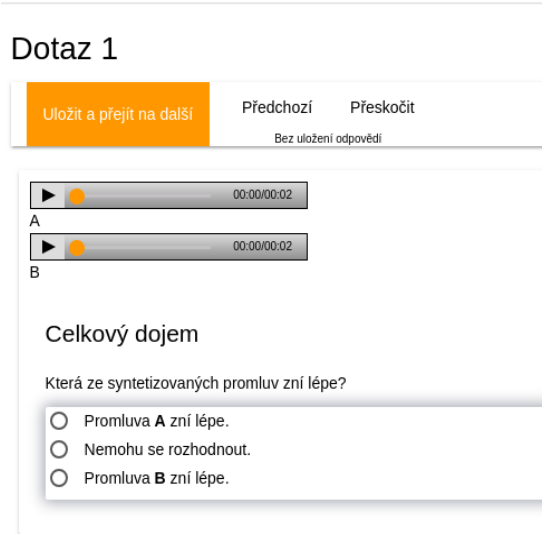
Testy přirozenosti kromě srozumitelnosti mají také posoudit celkovou kvalitu vnímání a míru přirozenosti generované řeči. Určit přirozenost řeči je také dost komplikovaná úloha, a proto se používají subjektivní metody.

Nejpoužívanějším testem je test definovaný Mezinárodní telekomunikační unií (*angl. International Telecommunication Union*, ITU) - **MOS** (*angl. Mean Opinion Score*). Testy MOS spočívají v tom, že posluchači hodnotí kvalitu přehrávaných promluv ve stupnici od 1 do 5. Potom lze vyhodnocením všech výsledků od všech posluchačů získat průměrnou známku přirozenosti. Podobná metoda je využita v tzv. MUSHRA testech, které jsou v současnosti často využity ve vědeckých publikacích. Momentálně jsou na katedře kybernetiky ZČU MUSHRA testy realizovány tak, že posluchači hodnotí kvalitu přehrávaných promluv od 0 do 100 vůči referenční ukázce, jak je to znázorněno na obrázku 3.



Obrázek 3: Framework KKY ZČU k hodnocení přirozenosti pomocí MUSHRA testu

Další metoda hodnocení přirozenosti řeči je porovnávání párů nebo se jí také říká „preferenční“ test. Posluchačům jsou k hodnocení nabízeny dvě stejné věty generované různými verzemi TTS systému a úkolem je vybrat tu, která se jim líbí více z hlediska přirozenosti. Existují různé variace stupnice k porovnávání dvou vzorků, katedra kybernetiky ZČU používá 3 stupně, které jsou znázorněny na obrázku 4. Obecně se používají ale i stupnice rozšířené do 5 či 7 stupňů.



Obrázek 4: Framework KKY ZČU k preferenčnímu testu

2.7 Aplikace syntézy řeči a systém TTS

Obecně je problém srozumitelnosti syntetické řeči z většiny vyřešen. Vědci jen neustále vylepšují systémy z pohledu jejich přirozenosti. A proto je v dnešní době hromadné automatizace procesu syntéza řeči velice používaná a užitečná hned v několika oblastech.

První z nich je medicína, a to zejména pro lidi s poruchami hlasu a nevidomé lidi. Němí lidé zapíší pomocí klávesnice svoje myšlenky a systém TTS vygeneruje jejich zvukovou reprezentaci. Speciální telefony dokonce umožňují i volání v reálném čase. V rámci již dokončeného projektu HCENAT na ZČU a probíhajícího projektu Laryngo Voice pro lidi, kteří mají riziko ztráty hlasu, existuje možnost připravit systém TTS tak, aby mluvil jejich hlasem. Toto by mělo pomoct a psychicky podpořit takové lidi a jejich příbuzné. Pro nevidomé lidi systém poskytuje možnost automatického čtení SMS, e-mailů, novin a knih.

Další použití systému TTS najdeme ve službách telekomunikací a monitorování. Jde o různé dialogové systémy od informačních linek po chytré domácnosti, nebo například komunikace automobilu či systému navigace s řidičem, pro kterého je vhodné vnímat informace ve zvukové podobě vzhledem k tomu, že vizuálně musí vnímat především situaci na silnici.

Mezi ostatní využití syntézy řeči můžeme zařadit automatický dabing filmu pro lidi s citlivým sluchem nebo jeho poruchami, výuku cizích jazyků, reklamy, hlášení v hromadné dopravě a zábavu, jako např. různé hry a hračky.

Kapitola 3

Slovní přízvuk

3.1 Přízvuk jako vlastnost slova

Vzhledem k tomu, že každé nezávislé slovo má vlastní přízvuk, přičemž obvykle pouze jeden, můžeme tvrdit, že přízvuk je jedna z hlavních vlastností nezávislého slova (11). Například při výslovnosti výrazu z románu „Dubrovský“ od Puškina

«На друг**о**й д**е**нь в**е**сть о пож**а**ре разнесл**а**сь по всем**у** окол**о**тку»[6]

(čes. *Na druhý den se zprávy o požáru rozšířily po celém okolí.*)

bude zřetelně slyšet sedm samostatných slov, protože ve výrazu se vyskytuje sedm přízvuků. Ale nelze říci, zda se v něm vyskytují pomocná slova a částice, protože ty nemají vlastní přízvuk a přiléhají k sousedním samostatným slovům. Nicméně, bez znalosti ruského jazyka, tj. bez znalosti slov, významu výrazů a také bez znalosti poměru mezi kvalitou samohlásek a přízvukem (nepřízvučné samohlásky jsou kvalitativně zredukovány), není možné stanovit hranice slov, protože v ruském jazyce přízvuk není stálý. V jazycích s pevným přízvukem umožňuje přízvuk také určit hranice mezi slovy, protože pokaždé ukazuje na určitou slabiku slova (např. v češtině na první, v arménštině na poslední, v polštině na předposlední atd.).

3.2 Slovní přízvuk a jeho fyzikální vlastnosti

Vyznačení přízvučné slabiky se obvykle provádí zesíleným výdechem, tj. exspirací. Z toho je odvozen pojem exspirační přízvuk. Přízvuk je produkován také změnou tónu, tj. jeho výškou. Obvykle oba tyto jevy fungují spolu, ale v jedněch jazycích přízvučná slabika vyniká především změnou síly výdechu, v jiných změnou výšky tónu.

V německém, francouzském, ruském, ukrajinském, běloruském, polském, českém, bulharském, a dokonce i latinském jazyce je přízvuk expirační. Naopak v srbsko-chorvatském, slovinském, klasickém řeckém jazyce a sanskrtu přízvuk vzniká změnou výšky tónu.

Jak uvádí autor (11), v ruštině je důrazu na přízvučné slabice dosaženo zesílením při výdechu, s tím je spojena i větší nataženost hlasivek. Akusticky to vytváří dojem větší hlasitosti přízvučné

[6] V této práci budou přízvučné samohlásky v ruském jazyce označeny větším, tučným a podtrženým písmenem.

slabiky oproti nepřízvučné. Přízvučná samohláska je vyslovována nejen hlasitě, ale také déle než odpovídající nepřízvučná.

3.3 Trvání samohlásek

Jak už bylo uvedeno výše, přízvučná samohláska je vyslovována déle než nepřízvučná. V různých jazycích jsou relace mezi přízvukem a větším či menším trváním (délkou) zvuku ve slabice odlišné. V řadě jazyků délka trvání samohlásek na přízvuku nezávisí, například ve francouzštině a češtině. V těchto jazycích nejsou rozdíly v délce trvání samohlásek foneticky podmíněny, a proto mohou být použity k určení významu slov. V těchto jazycích existuje kategorie dlouhých a krátkých samohláskových fonémů. „Například v českém jazyce:

- *Dráha* je vyslovováno s dlouhým **á** v přízvučné slabice, a krátkým **a** v nepřízvučné slabice.
- *Drahá* je vyslovováno s krátkým **a** v přízvučné slabice, a dlouhým **á** v nepřízvučné slabice.“ (11)

Nebo:

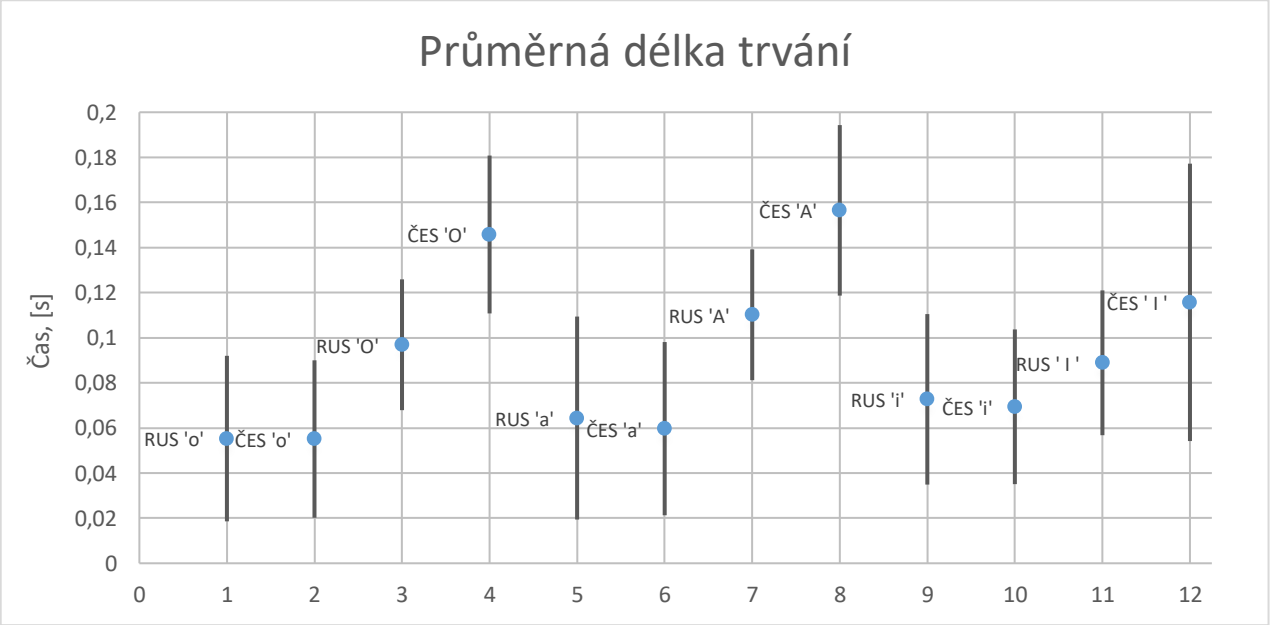
- Slovo *být* je vyslovováno s dlouhým **ý** v přízvučné slabice.
- Slovo *byt* je vyslovováno s krátkým **y** v přízvučné slabice.

V prvním případě se jedná o sloveso s ekvivalentním smyslem ke slovu *žít*, v druhém případě se ale jedná o nemovitost. Takže je vidět, že ve výše uvedených příkladech je rozdíl ve významu těchto slov opravdu určen délkou samohlásek.

V ruštině a některých dalších jazycích trvání samohlásek závisí na přízvuku. V ruštině oproti češtině neexistují slova, kde by samohlásky byly odlišně dlouhé nebo krátké, je to jen dáno přízvukem - přízvučné samohlásky jsou nutně vyslovovány dlouze. Naopak nepřízvučné samohlásky jsou nutně vyslovovány krátce. Na základě dat z vybraného ruského a českého řečového korpusu katedry kybernetiky ZČU v Plzni byly spočteny průměrné délky trvání nejčastěji použitých samohlásek ‚o‘, ‚a‘ a ‚i‘ (viz Tab. 1, Obr. 5). Při pohledu na tabulku 1 a graf na obrázku 5 je vidět, že délky trvání krátkých českých a nepřízvučných ruských samohlásek jsou velmi podobné, stejně jako délky trvání dlouhých českých a přízvučných ruských samohlásek. Z vypočtených statistik a jejich podobnosti můžeme usoudit, že ruské nepřízvučné samohlásky mají téměř stejnou délku jako české krátké. A ruské přízvučné samohlásky, přestože jsou kratší, svými vlastnostmi odpovídají dlouhým českým. V tom smyslu můžeme říct, že ruský a český jazyk jsou si podobné. Ale je důležité upozornit na to, že jak délka závisí na přízvučnosti, tak přízvuk v mnoha případech určuje význam slova.

	České, [ms]	Počet dat	Ruské, [ms]	Počet dat
O	145,83 ± 34,96	719x	96,92 ± 29,00	65 681x
o	55,11 ± 34,92	51 135x	55,29 ± 36,73	21 640x
A	156,53 ± 37,81	47 266x	110,22 ± 29,10	54 248x
a	59,67 ± 38,40	15 635x	64,43 ± 45,08	26 140x
I	115,67 ± 61,45	39 446x	88,98 ± 32,10	37 301x
i	69,42 ± 34,30	24 994x	72,72 ± 37,78	13 161x

Tabulka 1: Průměrná délka trvání samohlásek ruského a českého jazyka a jejich směrodatná odchylka, malá písmena ‚o', ‚a', ‚i' označují krátké nebo nepřízvučné samohlásky, velké ‚O', ‚A', ‚I' naopak



Obrázek 5: Graf průměrné délky trvání samohlásek ruského a českého jazyka a jejich směrodatná odchylka

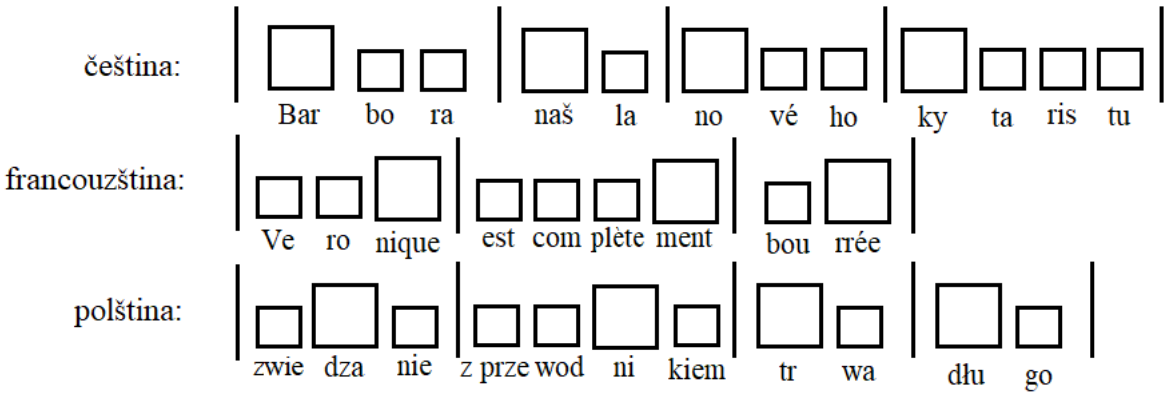
3.4 Různorodost přízvuku

V ruském jazyce je přízvuk různorodý (ve smyslu pohyblivý a volný), tj. jeho umístění není fixováno ani na jakoukoliv určitou slabiku slova, ani na konkrétní část slova a může být umístěn na jakoukoliv slabiku slova nebo jakoukoliv jeho část. Ukážeme si to na příkladu, kde přízvuk je umístěn na různých slabikách a částech slov:

- na kmeni - фр**у**кты, хом**я**к, г**а**ечка, **ё**ршик (*čes. ovoce, křeček, matice šroubu, ježdík*)
- na předponě - в**ы**яснить, н**а**крест, п**о**мощь (*čes. vyjasnit, křižem, pomoc*)

- na příponě - хомяч**о**к, маз**ю**кать, саранч**о**вый (*čes. zdrobnělina slova křeček, čmárat, sarančí*)
- na koncovce - рук**а**ми, молок**о**, профессор**а** (*čes. rukama, mléko, profesora*)

Naopak v mnoha jiných jazycích není přízvuk různorodý a je fixován k určité slabice. Tak například v českém, lotyšském, estonském a finském jazyce je přízvuk vždy umístěn na první slabice, v polském a gruzínském - na předposlední slabice, v arménském a francouzském - na poslední slabice atd. Autoři knihy „Zvuková báze řečové komunikace“ (12) uvádějí následující ukázku umístění slovního přízvuku v češtině, francouzštině a polštině, kde větší čtverce označují přízvučné slabiky, menší nepřízvučné a svislé čáry zdůrazňují hranice mluvních taktů:



Obrázek 6: Ukázka umístění slovního přízvuku v češtině, francouzštině a polštině

Různorodost přízvuku v ruském jazyce je použita k rozlišení slov podle smyslu, který je podstatný pro práci modulu kontextové analýzy při syntéze řeči, což také bylo důvodem k vytvoření této práce. Stačí změnit umístění přízvuku, aby se slova, která mají stejnou grafickou podobu, lišila ve smyslu. Uvádím příklady převzaté z (13):

- з**а**мок и зам**о**к (*čes. hrad/zámek a zámek na klič*),
- пр**о**пасть и проп**а**сть (*čes. propast a zmizet*),
- хл**о**пок и хлоп**о**к (*čes. bavlna a tlesknutí*),
- в**и**на и вин**а** (*čes. vína a provinění*),
- **о**рган и орг**а**н (*čes. orgán a varhany*),
- **а**тлас и атл**а**с (*čes. atlas a satén*),

- п**о**зднее и поздн**е**е (*čes. pozdní a později*),
- с**у**шу и суш**у** (*čes. souš a suším*),
- п**о**том и пот**о**м (*čes. potem a potom*),
- М**а**шина и маш**и**на (*čes. Máši a auto*),
- с**а**мого и самог**о** (*čes. samého a samotného*).

V angličtině přízvuk také není stálý a také se setkáváme s tím, že přízvuk určuje význam slov, například:

- confl**i**ct (sloveso, které znamená „být v rozporu“)
- c**o**nflict (podstatné jméno, které znamená „spor“, „konflikt“)

nebo

- r**e**cord (podstatné jméno, které znamená „zápis“, „záznam“)
- rec**o**rd (sloveso, které znamená „zapisovat“, „poznamenávat“)

Je zřejmé, že v jazycích se stálým přízvukem nemůže přízvuk sloužit k rozlišování slov. Jak už bylo řečeno, v češtině se například pro rozlišování slov používají dlouhé a krátké samohlásky.

3.5 Přízvuk pohyblivý a stálý

V ruském jazyce je přízvuk u některých kategorií slov pevný, to znamená, že při tvorbě gramatických forem slova zůstává vždy na stejném místě. U jiných kategorií slov je přízvuk pohyblivý a při tvorbě různých gramatických forem se přemisťuje mezi slabikami a částmi slova. V případě, že je přízvuk pohyblivý, se pohyblivost přízvuků používá k tvorbě a rozlišení gramatických forem. Primárním způsobem je ale tvorba pomocí afixů: předpony, přípony a koncovky. Přízvuk je tedy doplňkovým, pomocným gramatickým nástrojem k tvorbě slov. Tudíž různé formy stejného slova se můžou lišit od sebe buď přízvukem, nebo se obvykle liší různými afixy (většinou koncovkami).

Například zkusíme skloňovat slova арб**у**з, бл**Ю**до, дос**у**г, кварт**а**л[7] (viz Tab. 2). Zde při tvorbě gramatických forem byly použity koncovky a přízvuk všude zůstal na stejném místě. Naopak u řady slov s pohyblivým přízvukem byl přechod z jednotného čísla do množného realizován nejen přidáním koncovky, ale i přízvuk umístěný na kmenu se při změně čísla přemisťuje na koncovku (viz. Tab. 3). Potom uvažujeme také o přízvuku volném, změna umístění, která už není podmíněna změnou mluvnické kategorie, ale změnou slova jako takového, tj. mění jeho význam, příklady viz kapitola 3.4.

Jednotné číslo				
1. pád	арб**у**з	бл**Ю**до	дос**у**г	кварт**а**л
2. pád	арб**у**за	бл**Ю**да	дос**у**га	кварт**а**ла
3. pád	арб**у**зу	бл**Ю**ду	дос**у**гу	кварт**а**лу
4. pád	арб**у**з	бл**Ю**до	дос**у**г	кварт**а**л
5. pád	арб**у**зе	бл**Ю**де	дос**у**ге	кварт**а**ле
6. pád	арб**у**зом	бл**Ю**дом	дос**у**гом	кварт**а**лом
Množné číslo				
1. pád	арб**у**зы	бл**Ю**да	дос**у**ги	кварт**а**лы

Tabulka 2: Skloňování slov арбуз, блюдо, досуг, квартал (čes. meloun, jídlo, volný čas, čtvrť)

Jednotné číslo		
1. pád	г**о**род	уч**и**тель
Množné číslo		
1. pád	город**а**	учител**я**
2. pád	город**о**в	учител**е**й
3. pád	город**а**м	учител**я**м
4. pád	город**а**	учител**е**й
5. pád	город**а**ми	учител**я**х
6. pád	город**а**х	учител**я**ми

Tabulka 3: Skloňování slov города, учителя (čes. města, učitelé)

[7] Pozn.: V ruském jazyce je jenom 6 pádů, český 5. pád je vynechán.

Kapitola 4

Úvod do strojového učení

4.1 Základy strojového učení

Pojem strojové učení byl zaveden v roce 1959 Arturem Samuelem. Strojové učení je oblast počítačových technologií, které využívají statistické metody, aby umožnily počítačům „nastudování" dat bez zjevného programování (14). Strojové učení zkoumá osvojení a vybudování algoritmů, které se mohou učit a predikovat na základě dat (15). Takové algoritmy překonávají statistické instrukce predikcí a rozhodováním, které je založeno na modelech pracujících se vzorky dat (16).

Úlohy strojového učení se dělí na dvě rozsáhlé kategorie v závislosti na tom, je-li k dispozici „signál" k učení nebo zpětná vazba:

- Učení s učitelem (*angl. Supervised Learning*)

  Počítači jsou představeny příklady vstupních i výstupních dat (požadovaný výstup) poskytnutých „učitelem". Cílem je vytvoření obecného pravidla, které definuje vztah mezi vstupem a výstupem. Výjimečně vstup může být zredukován nebo omezen na zvláštní zpětnou vazbu.

- Učení bez učitele (*angl. Unsupervised Learning*)

  Počítač má k dispozici pouze vstupní data a musí samostatně určit strukturu jen na základě vstupních dat. Učení bez učitele může být použito za účelem nalezení skrytých šablon v datech nebo přímého prostředku k učení. Při učení bez učitele se provádí automatické shlukování buď do předem známého, nebo neznámého počtu tříd.

V rámci této práce byl vybrán přístup Učení s učitelem.

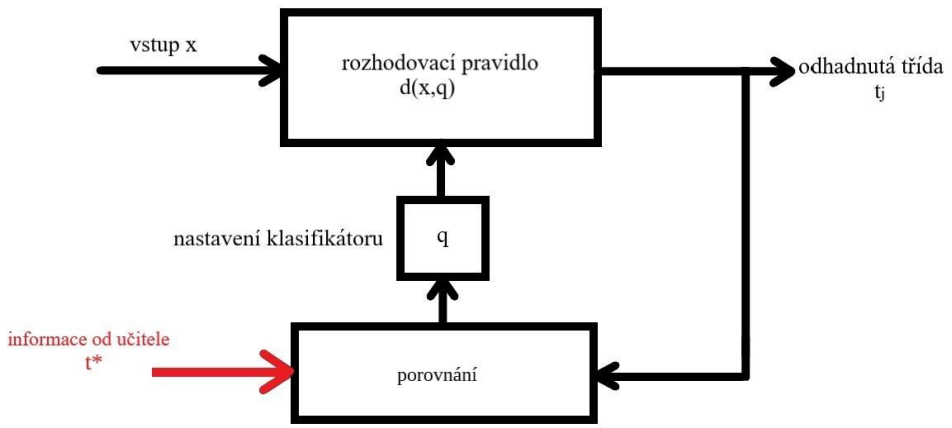
4.2 Základy klasifikačních přístupů

Historicky vznikla úloha klasifikace od úlohy počítačového vidění, kvůli tomu se klasifikaci také říká rozpoznávání obrazu (17). V klasické úloze klasifikace jsou data představena jako sada objektů $X = \{x_i\}_{i=1}$, které jsou popsány vektorem skutečných vlastností $x_i = (x_{i,1}, \dots, x_{i,d})$ (říkáme jim příznaky). Danému vektoru, jehož souřadnice tvoří příznaky, se říká obraz. V praxi

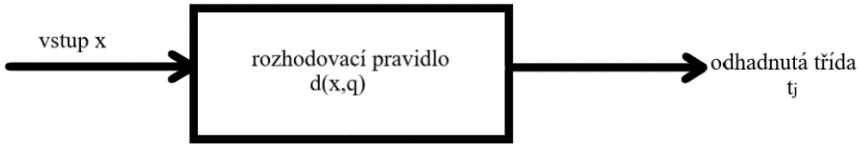
se nerozpoznává objekt, ale jeho obraz. Rozpoznávání předmětů je totiž třídění do jednotlivých tříd. Počet tříd je představen konečnou množinou $T = \{t_1, \dots, t_n\}$. Úlohou je tedy sestavit algoritmus (klasifikátor), který podle vektoru příznaků x_i, vrátí příslušnost objektu k některé třídě t_j z množiny T, nebo vektor aposteriorních pravděpodobností příslušnosti objektu ke každé třídě z množiny T.

Klasifikační přístupy jsou využívány v různých oblastech, mezi ně patří i oblasti řečových technologií (18). Při návrhu systému automatického rozpoznávání se obvykle řeší problém volby příznaků a návrh klasifikátoru. Na rozdíl od návrhu klasifikátoru, pro který už jsou v současné době dobře propracované metody, způsob volby příznaků zůstává na návrhářích a podporuje se jenom expertními znalostmi. Při volbě příznaků je důležité, aby příznaky popisovaly co nejjednoznačněji rozdíl mezi jednotlivými třídami. Ale ne vždy se podaří vybrat příznaky, které by byly schopny přesně (lineárně) odlišit například 2 třídy mezi sebou, proto je potom třeba minimalizovat chybu špatné klasifikace, tj. stanovit rozhodovací kritérium tak, abychom minimalizovali ztrátu.

Jak bylo popsáno v bodě 4.1, máme-li informace o počtu tříd a množině obrazů (vstupní data, pro která je známa správná klasifikace) a výstupní data, pak se jedná o řešení metodami učení s učitelem. Činnost klasifikátoru se dělí na fázi nastavování neboli trénování (viz Obr. 7) a fázi klasifikace či testovaní (viz Obr. 8).



Obrázek 7: Blokové schéma fáze nastavování klasifikátoru



Obrázek 8: Blokové schéma fáze klasifikace

4.2 Metody strojového učení

V této kapitole budou rozebrány metody strojového učení použité při zpracování praktické části: Logistic Regression, Support Vector Machine (SVM) a jednoduchá neuronová síť.

4.2.1 Logistic Regression

Logistická regrese je model, který byl v roce 1958 navrhnut statistikem Davidem Coxem, používá se ve statistice pro predikci pravděpodobnosti toho, že by nastal nějaký jev podle množství příznaků (19). Statisticky se definuje tak, že existuje sada jevů $x_1, x_2, \dots, x_n$ (neboli nezávislých proměnných) a y (neboli závislá proměnná), která říká, jestli nastal daný jev – $(y = 1)$ nebo nenastal – $(y = 0)$. Úkolem logistické regrese je určit pravděpodobnost toho, že daný jev nastane, či ne:

$P\{y = 1|x\}$, y má hodnotu 0 nebo 1, tím pádem $P\{y = 0|x\} = 1 - P\{y = 1|x\}$

Předpokládá se, že $P\{y = 1|x\} = f(x)$, kde $f(x) = \frac{1}{1+e^{-x}}$, k tomu je ale třeba přidat parametry $\{\omega_0, \dots, \omega_n\}$, které se opírají o nezávislé proměnné. K tomuto účelu nejlépe poslouží lineární kombinace nezávislých proměnných $\omega^T = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n$ a pravděpodobnost se potom definuje jako:

$$P\{y = 1|x\} = f(x) = \frac{1}{1+e^{\omega_0 + \omega^T x}}, \qquad P\{y = 0|x\} = 1 - f(x)$$

Pak obecně platí:

$$P\{y|x\} = f(x)\big(1 - f(x)\big), y \in \{0,1\}.$$

Pro výběr parametrů $\{\omega_0, \dots, \omega_n\}$ se pak používá metoda maximální věrohodnosti.

Z pohledu klasifikace je to lineární klasifikátor, takže nezávislé proměnné $x_1, x_2, \dots, x_n$ jsou vektory příznaků zkoumaného objektu (v daném případě je zkoumaným objektem samohláska a příznaky jsou okolní znaky), závislá proměnná y je tedy klasifikace do 2 tříd. Tudíž na základě daného modelu a získaných pravděpodobností je možné zařadit objekt do třídy, která udává větší pravděpodobnost, neboli lze říct, zda je samohláska přízvučná, či není.

4.2.2 Support Vector Machine

Úkolem této metody je nalezení takové hyperplochy, aby suma vzdáleností k nejbližšímu bodu z každé množiny byla maximální (20). Pokud taková hyperplocha existuje, příslušný její lineární klasifikátor se nazývá optimální klasifikátor.

Podobně jako v případě logistické regrese máme dvojice proměnných, kde $x_i \in R^n$ a $y_i \in \{-1,1\}$, $i = 1, \dots, n$ a považujeme je za separabilní hyperplochu $(\overline{\omega} \cdot \overline{x}) - b = 0$, pokud existuje jednotkový vektor $|\overline{\omega}| = 1$ a b takový, že platí:

$$(\overline{\omega} \cdot \overline{x_i}) - b > 0, y_i = 1 \quad (1)$$

$$(\overline{\omega} \cdot \overline{x_i}) - b < 0, y_i = -1 \quad (2)$$

Podmínky (1) a (2) můžou byt zapsány jako

$$y_i\big((\overline{\omega} \cdot \overline{x_i}) + b\big) \geq 1, \quad 1 \leq i \leq n \quad (3)$$

Tedy k nalezení optimální hyperplochy je třeba minimalizovat normu vektoru $\|\omega\|$ za podmínky (3), což je úlohou kvadratické optimalizace:

$$(\overline{\omega} \cdot \overline{\omega}) = \sum_{i=1}^{n} \overline{\omega_i}^2 \rightarrow min$$

Podle Kuhnovy-Tuckerovy věty je to ekvivalentní úloze nalezení sedlového bodu Lagrangianu:

$$L(\overline{\omega}, b, \overline{\alpha}) = \frac{1}{2}(\overline{\omega} \cdot \overline{\omega}) - \sum_{i=1}^{n} \alpha_i\big(y_i\big((\overline{\omega} \cdot \overline{x_i}) + b\big) - 1\big) \quad (4),$$

kde $\alpha_i \geq 0$ jsou Lagrangiovy činitelé (multiplikátory).

Po vyřešení dané úlohy ω a b jsou definovány vzorce:

$$\overline{\omega} = \sum_{i=1}^{n} \alpha_i y_i \overline{x_i}$$

$$b = \overline{\omega} \cdot \overline{x_i} - y_i, \ \alpha_i > 0$$

Potom algoritmus klasifikace vypadá takto:

$$a(x) = sign(\textstyle\sum_{i=1}^{n} \alpha_i y_i \overline{x_i} \cdot \overline{x} + b), \ \alpha_i > 0 \quad (5)$$

Uvedený algoritmus klasifikace popisuje obecný princip metody Support Vector Machine, který odpovídá použití lineárního jádra, když jsou data dobře lineárně separovatelná. Existuje ale však metoda, která umožňuje provádět klasifikace i nelineárně separovatelných dat, říká se jí jádrový trik (*angl. kernel trick*). Jádrový trik spočívá v tom, že ve vzorci (5) je skalární součin nahrazen příslušnou jádrovou funkcí.

V této práci jsou použita 2 jádra:

- Lineární jádro, jádrová funkce v daném případě odpovídá skalárnímu součinu
- RBF jádro (*angl. Radial Basic Function*), matematicky je definovaná jako

$exp\left(\frac{-\|\overline{x_i}-\overline{x}\|^2}{2\sigma^2}\right)$, kde σ je volný parametr.

4.2.3 Neuronová síť

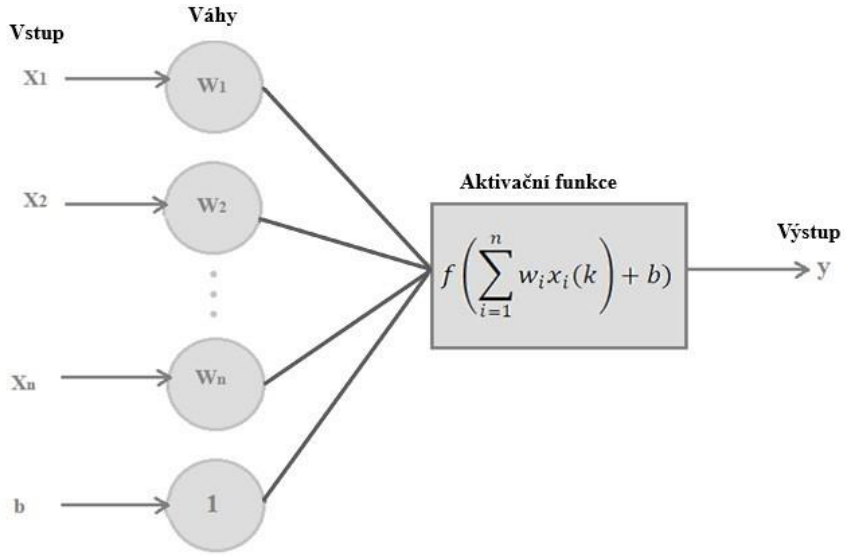
Umělá neuronová síť je jedním z modelů používaných v oboru umělé inteligence (21). Umělá neuronová síť je postavena na skutečném chování biologické neuronové sítě živých organismů. Biologická neuronová síť se skládá z množství neuronů, komunikujících mezi sebou prostřednictvím sítě vazeb. Umělá neuronová síť je stejná, skládá se z umělých neuronů, které jsou vzájemně propojeny a navzájem si předávají signály.

Jedním ze základních a zároveň nejjednodušších modelů je model perceptronu, vynalezený v roce 1958 F. Rosenblatem (viz Obr. 9). Daný model je popsán vektorem vstupu $x = [x_1, x_2, \dots, x_n]^T$, který udává vstupní informace neboli počáteční stav. Váhový vektor $w = [w_1, w_2, \dots, w_n]^T$ slouží jako budicí nebo tlumicí signál a ve složitějších modelech slouží k učení sítě. Práh b určuje aktivaci neuronu. U modelu perceptronu se používají také základní aktivační funkce f a výstup y, který se počítá pro každý průchod sítě podle vztahu

$$y(k+1) = f(\sum_{i=1}^{n} w_i x_i(k) + b) = f(w^T x(k) + b),$$

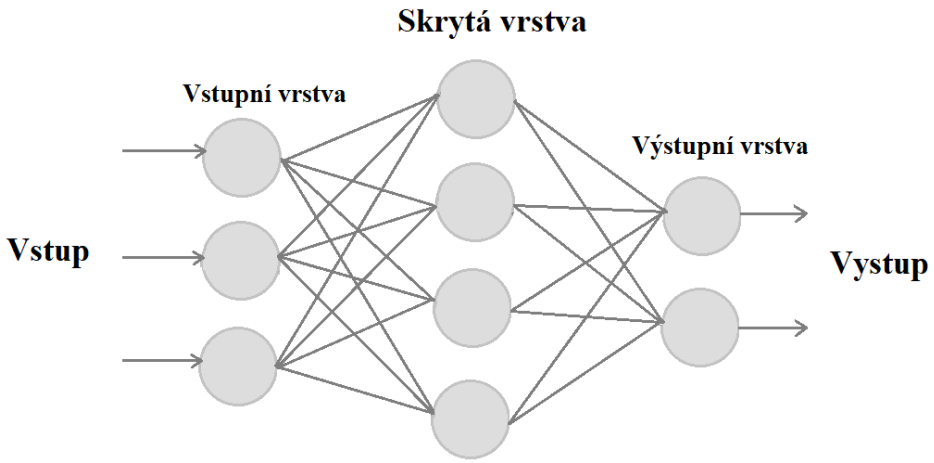
kde $w^T x(k) + b = \xi$ je aktivační hodnota.

Při použití bipolární binární aktivační funkce $f(\xi) = sgn(\xi) = \begin{cases} +1, \xi \geq 0 \\ -1, \xi < 0 \end{cases}$ perceptron rozdělí vstupní rovinu na 2 poloroviny, kde jedné přiřadí hodnotu +1 a druhé -1. Funkce sítě je určena způsobem propojení neuronů, váhami těchto spojení a způsobem práce jednotlivých neuronů, tj. aktivační funkcí. Základními typy jsou vícevrstvé dopředné neuronové sítě a neuronové sítě se zpětnou vazbou.



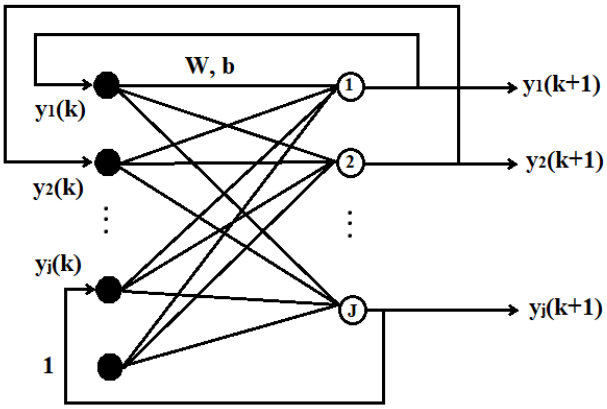
Obrázek 9: Model perceptronu

Vícevrstvé dopředné sítě, nebo také sítě s dopředným šířením (*angl. feedforward networks*), jsou takové sítě, u kterých je výstup jedné vrstvy napojen na vstup následující vrstvy a signál se šíří pouze ze vstupu sítě k jejímu výstupu. Model vícevrstvé dopředné sítě je znázorněn na Obrázku 10.



Obrázek 10: Model vícevrstvé dopředné sítě

Sítě se zpětnou vazbou, neboli rekurentní sítě (*angl. feedback networks*), jsou oproti dopředným sítím takové, že se signál šíří také z výstupu sítě zpět k jejímu vstupu. Neboli její výstup slouží jako vstup pro další průchod sítě. Model sítě se zpětnou vazbou je znázorněn na Obrázku 11.



Obrázek 11: Model sítě se zpětnou vazbou

V této práci byl použit příklad dopředné sítě, který byl uveden v knize Deep Learning With Python (22) (podrobně viz bod 5.3).

Síti je předložena trénovací dvojice: vstup x_p a požadovaný výstup u_p. Trénování probíhá v tzv. trénovacích cyklech, kdy během jednoho trénovacího cyklu jsou síti předloženy všechny dvojice z trénovací množiny právě jednou. Trénování sítě spočívá v tom, že se najde optimální nastavení vah a prahu a minimalizuje se ztrátová funkce, která je definovaná vztahem

$$E = \sum_{p=1}^{P} \varepsilon_p = \frac{1}{2}\sum_{p=1}^{P} \left\| u_p - y_p \right\|^2,$$

kde $y_p = sgn(W \cdot x_p + b)$ je skutečný výstup sítě pro vstup x_p, požadovaný výstup sítě pro vstup $x_p - u_p$, P je počet trénovacích dvojic, ε_p je chyba od p-té trénovací dvojice. Ztrátová funkce v podstatě říká, jak moc se liší predikce sítě od skutečného výsledku přes celou trénovací množinu. Na základě toho se provádí přepočet vah a prahu (optimalizace), který může probíhat buď po průchodu sítí každé trénovací dvojice, anebo až na konci trénovacího cyklu.

Kapitola 5

Praktická část

V této kapitole bude podrobně probrána příprava a zpracování trénovacích a testovacích dat.

5.1 Příprava trénovacích dat

Když se vrátím k tématu práce – Detekce přízvuků v ruštině, je zřejmé, že se při přípravě trénovacích dat bude jednat o práci s textovou sadou. Na druhou stranu počítače pracují s čísly, tudíž úloha přípravy trénovacích dat prošla několika fázemi.

5.1.1 První fáze

První fáze spočívala v nalezení textů, které by obsahovaly obecnou rozsáhlou ruskou řeč. Za tímto účelem bylo v první části práce staženo velké množství novinových zpráv z online časopisů na internetu. Potom bylo třeba provést analýzu, s jakou frekvencí se vyskytují v psané řeči různá slova. Na základě analýzy bylo vybráno 10 slov s nejednoznačným přízvukem. Vzhledem k tomu, že v běžné řeči výskyt různých pádů a čísel jednoho a toho samého slova (tj. slova, které má pohyblivý přízvuk; viz kapitoly 3.4 a 3.5) je častější než výskyt slova, jehož význam ovlivňuje přízvuk (tj. slova, která mají volný přízvuk), dalo se očekávat, že všechna slova z 10 vybraných byla s pohyblivým přízvukem:

1. году
    - к 2010-му г**о**ду (*čes. do roku 2010*),
    - в прошлом год**у** (*čes. loni*)
2. города
    - день г**о**рода (*čes. den města*),
    - большие город**а** (*čes. velká města*)
3. дома
    - у него нет д**о**ма (*čes. on nemá dům*),
    - панельные дом**а** *(čes. panelové domy*)
4. места
    - мало м**е**ста *(čes. málo místa*),

- лучшие мест**а** *(čes. nejlepší místa*),

5. права
    - вы не имеете пр**а**ва *(čes. vy nemáte právo*),
    - все прав**а** охраняются законом *(čes. veškerá práva jsou chráněna zákonem*)
6. самом
    - на с**а**мом деле *(čes. ve skutečnosti*),
    - в сам**о**м городе *(čes. v samotném městě*)
7. слова
    - хозяин своего сл**о**ва *(čes. velitel svého slova*),
    - это все лишь слов**а** *(čes. jsou to jen slova*)
8. стоит
    - сколько это ст**о**ит? *(čes. kolik to stojí?*),
    - давно уже тут сто**и**т? *(čes. jak dávno to tady stojí?*)
9. стороны
    - с правой сторон**ы** *(čes. po pravou stranu*),
    - обе ст**о**роны *(čes. obě strany*)
10. страны
    - стр**а**ны ЕС *(čes. státy EU*),
    - в интересах стран**ы** *(čes. v zájmu státu*)

Vzhledem k tomu, že celá fáze byla provedena ručně (kromě stahování novinových článků a následný vypočet frekvence výskytů), bylo rozhodnuto vybrat pro každé slovo 100 (později dalších 100, celkově 200) případů výskytu a pro každý z nich označit přízvuk v rozpoznávaném slově. Textové okolí slova zde hraje velmi důležitou roli, protože na něm ten tvar a smysl slova přímo závisí. Ruční práce umožnila vybrat data takovým způsobem, že počet případů s každou možnou variantou přízvuků je téměř stejný, občas se ale setkáváme se sedmdesáti- až osmdesátiprocentní předností jedné z variant. V první části všechna připravená data byla použita jako trénovací a zároveň testovací (viz bod 5.2).

V druhé části jsme se rozhodli získat rozsáhlejší řečový korpus. Tímto korpusem se stal Ruský národní korpus (*angl*. *Russian National Corpus*, RNC), který byl poskytnut Ruskou akademií věd. Na svých webových stránkách autoři popisují korpus následovně:

„Národní korpus ruského jazyka zahrnuje především prozaické originální texty zastupující ruský literární jazyk (od začátku 18. století), ale také (v menším rozsahu) přeložená díla (spolu s původními), poetické texty i texty reprezentující neliterární formy moderního ruského jazyka: hovorovou (záznam ústní řeči, veřejné a neveřejné) a dialekty.“ (23)

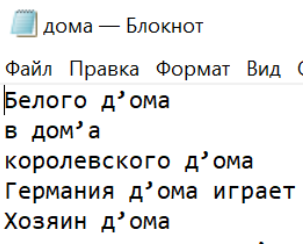
V této části ručně vybrané příklady slouží jako trénovací data a část RNC slouží jako testovací data. Zde pro optimalizaci zpracování dat z celého korpusu byly pro každé z 10 původně vybraných slov vybrány jenom řádky obsahující dané slovo. Takže celý korpus byl rozdělen na 10 testovacích souborů. V tabulce 4 jsou uvedeny výskyty jednotlivých variant přízvuku v RNC pro každé z 10 vybraných slov. Z tabulky je vidět, že jedna z variant bývá mnohem častější.

	1. slabika	2. slabika	3. slabika
Году	13	403	-
Города	112	-	14
Дома	195	53	-
Места	89	60	-
Права	49	34	-
Самом	322	15	-
Слова	90	58	-
Стоит	141	130	-
Стороны	33	-	320
Страны	60	197	-

Tabulka 4: Počet výskytů jednotlivých přízvuků v RNC

5.1.2 Druhá fáze

Za účelem dalšího zpracování textu byl přízvuk v rozpoznávaném slově označen apostrofem před přízvučnou samohláskou (viz Obr. 12).



Obrázek 12: Ukázka souboru pro slovo „дома“

V kapitole 4 je podrobně probrána klasifikace a práce s klasifikátory, podle toho byla zvolena jednoduchá klasifikace do 2 tříd:

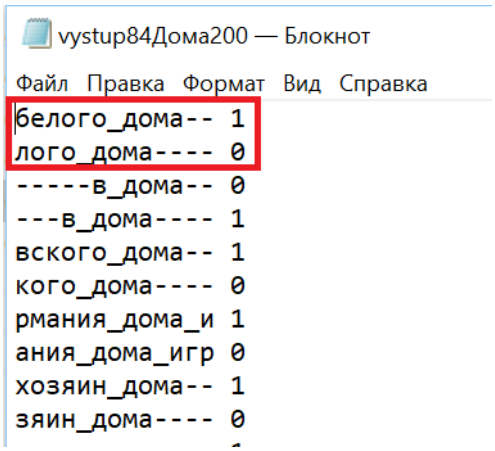
- Třída s přízvukem (samohláska je přízvučná) – označení 1
- Třída bez přízvuku (samohláska není přízvučná) – označení 0

Dále byla zvolena délka pravého a levého kontextu, která se počítala jako počet znaků zprava a zleva od samohlásky, pro kterou se určovala přízvučnost. Původně to bylo 5 znaků zleva a 5 znaků zprava od samohlásky, dále byly postupně otestovány kratší a delší kontexty v závislosti na tom, jaké výsledky udávala každá změna, celkově 9 variací (viz Tab. 5). Předpokladem také je to, že význam více ovlivňuje počet znaků zleva než zprava.

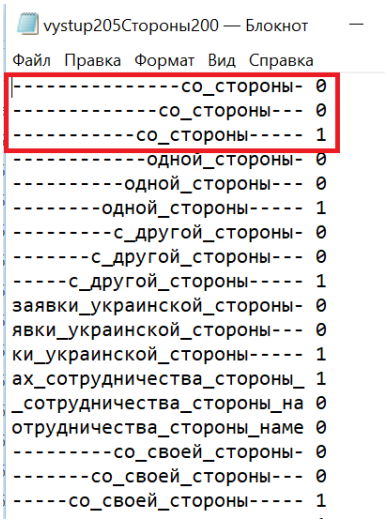
Levá délka	5	8	5	6	20	5	10	6	0
Pravá délka	5	4	3	5	5	8	10	3	0

Tabulka 5: Vybrané délky kontextu

Ručně připravená data (viz Obrázek 12) bylo tedy třeba upravit tak, aby obsahovala pouze danou velikost textového kontextu. Vytvořený skript procházel každý řádek a v rozpoznávaném slově hledal jakoukoli samohlásku, vybíral od ní zadaný počet znaků zleva a zprava, pokud znaky chyběly, byly doplněny určitým množstvím znaků ‚-'. Při výskytu apostrofu před samohláskou byla celému řádku přiřazená hodnota 1, v opačném případě 0. Na každou samohlásku byl tedy připraven jeden řádek, tj. pro slovo se dvěma slabikami dva řádky, s třemi slabikami tři. Pro každou variantu délky kontextu byl vytvořen výstupní soubor, který obsahoval všechny případy výskytu daného slova ve tvaru popsaném výše (viz Obr. 13 a Obr. 14). Daným způsobem byla zpracována jak ručně vybraná data, tak i RNC.



Obrázek 13: Ukázka výstupního souboru pro slovo дома s délkami kontextu L-8, P-4



Obrázek 14: Ukázka výstupního souboru pro slovo стороны s délkami kontextu L-20, P-5

5.1.3 Třetí fáze

Poté co byly vytvořeny výstupní soubory, probíhalo jejich zpracování již automaticky. Pro umožnění činnosti klasifikátorů pak bylo nutné převést každý řetězec znaků do číselné podoby. Jak už bylo řečeno v bodě 5.1.2, řádek se skládá z textu a klasifikace nulou nebo jedničkou. Text je zpracováván zvlášť a klasifikace je tvořena čísly, proto může rovnou tvořit vektor. Při zpracování textové části byla každému řádku (bez znaku klasifikace) vytvořena jeho „abeceda“. Abeceda se tvoří tak, že se vezmou všechny znaky vyskytující se ve zpracovaném souboru na určitých pozicích v řetězci. Například pro soubor tvořený řádky

дентом_страны

ентом_страны-

mapa vypadá následujícím způsobem:

{'д:0': 1, 'е:1': 1, 'н:2': 1, 'т:3': 1, 'о:4': 1, 'м:5': 1, '_:6': 1, 'с:7': 1, 'т:8': 1, 'р:9': 1, 'а:10': 1, 'н:11': 1, 'ы:12': 1}

{'е:0': 1, 'н:1': 1, 'т:2': 1, 'о:3': 1, 'м:4': 1, '_:5': 1, 'с:6': 1, 'т:7': 1, 'р:8': 1, 'а:9': 1, 'н:10': 1, 'ы:11': 1, '-:12': 1}.

A abeceda je potom tvořena 26 „znaky” v unikátní kombinaci znak:pozice a je seřazená podle ASCII tabulky:

'-:12’, ‘_:5’, ‘_:6’, ‘а:10’, 'а:9’, ‘д:0’, ‘е:0’, ‘е:1’, ‘м:4’, 'м:5’, 'н:1’, 'н:10’, ‘н:11’, ‘н:2’, ‘о:3’, ‘о:4’, ‘р:8’, ‘р:9’, ‘с:6’, ‘с:7’, 'т:2’, 'т:3’, 'т:7’, 'т:8’, ‘ы:11’, ‘ы:12’.

Potom se na základě vytvořené abecedy funkcí **DictVectorizer()** z knihovny scikit-learn[8] (24) znaky namapují na konkrétní pozice v řetězci (viz Obr. 15). Vektor pro každý řádek je poměrně dlouhý vektor nul a jedniček. Vektor tím reprezentuje, které znaky se na kterých pozicích v daném řádku vyskytují. Tímto způsobem je zpracován každý řádek v daném souboru. A každý soubor je

[8] knihovna, která je určena pro strojové učení

pak reprezentován dvěma vektory: vektorem textové části v číselné podobě a vektorem správné klasifikace.

```
[ 0.   0.   1.   1.   0.   1.   0.   1.   0.   1.   0.    0.    1. 1. 0.   1.   0.   1.   0.   1.   0.   1.   0.   1.    0.   1.]
 -:12 _:5  _:6 а:10 а:9  д:0  е:0  е:1  м:4  м:5  н:1  н:10  н:11 н:2 о:3  о:4  р:8  р:9  с:6  с:7  т:2  т:3  т:7  т:8  ы:11 ы:12
[ 1.   1.   0.   0.   1.   0.   1.   0.   1.   0.   1.    1.    0.  0. 1.   0.   1.   0.   1.   0.   1.   0.   1.   0.    1.   0.]
 -:12 _:5  _:6 а:10 а:9  д:0  е:0  е:1  м:4  м:5  н:1  н:10  н:11 н:2 о:3  о:4  р:8  р:9  с:6  с:7  т:2  т:3  т:7  т:8  ы:11 ы:12
```

Obrázek 15: Ukázka mapování řetězců „дентом_страны” a „ентом_страны-“

5.2 Trénování a testování klasifikátorů

Samotný proces trénování, testování a vyhodnocování výsledků klasifikace je poměrně jednoduchý. To je z důvodu použití knihovny scikit-learn už uvedené výše (24), která poskytuje bohatou funkcionalitu pro strojové učení. Trénování a testování bylo provedeno pomocí standardních funkcí knihovny. Přitom používají se 3 vektory z kapitoly 5.1.3. První je vektor trénovacích dat, kde každá položka je řetězec reprezentovaný nulami a jedničkami. Druhý je vektor klasifikace, kde každá položka odpovídá zařazení řetězce do třídy 0 nebo 1. Třetí je vektor testovacích dat, obdobně jako vektor trénovacích dat. Knihovna scikit-learn poskytuje klasifikátory: Logistic Regression a Support Vector Machine. Klasifikátor Support Vector Machine v sobě zahrnuje nastavení jádra, které slouží jako jedna z metod řešení lineární neseparability. V bodě 4.2 je SVM popsán obecně a odpovídá lineárnímu jádru. Změna jádra vlastně znamená to, že se každý skalární součin v uvedených vzorcích počítá podle určitého jádra. Změna jádra určuje samotný algoritmus klasifikace (25). V této práci bylo použito lineární jádro a radiální jádro.

Klasifikátor byl natrénován pomocí standardní funkce **fit()** a otestován na trénovacích datech funkcí **predict_proba()**. Jak bylo popsáno v sekci 5.1.2, pro každé predikované slovo bylo vytvořeno tolik řádků, kolik slovo obsahuje samohlásek. Funkce **predict_proba()** vrací pro každý řádek pravděpodobností příslušnosti k třídě 0 a k třídě 1. To umožňuje spočítat úspěšnost nikoliv pro každý řádek zvlášť, ale pro každé celé slovo. To znamená, že v případě dvojslabičného slova víme, že od začátku souboru každé dva řádky patří k sobě, podobně tak v trojslabičných slovech, ale řádky tvoří skupiny po trojicích. Přepočet pravděpodobnosti pro celé slovo se dělá tak, že pro dva nebo tři za sebou jdoucí řádky jsou vraceny pravděpodobnosti příslušnosti do třídy 1. Dále se pravděpodobnosti příslušnosti do třídy 1 porovnávají mezi sebou. Ten řádek, jehož pravděpodobnost je nejvyšší, dostává hodnotu 1, zbytek se ohodnotí jako 0. Můžeme o tom mluvit tak jednoznačně, protože z Kapitoly 3 víme, že přízvuk v ruském jazyce se v každém ze slov vyskytuje poze jednou.

```
1  -----со_стороны------ 0
2  ---со_стороны-------- 0
3  -со_стороны---------- 1
4  --одной_стороны------ 0
5  одной_стороны-------- 0
6  ной_стороны---------- 1
7  _другой_стороны------ 0
8  ругой_стороны-------- 0
9  гой_стороны---------- 1
10 аинской_стороны------ 0
11 нской_стороны-------- 0
12 кой_стороны---------- 1
13 ичества_стороны_намер 1
14 ества_стороны_намерен 0
15 тва_стороны_намерены- 0
```

Obrázek 16 Ukázka rozdělení případů na řádky

Uvedeme příklad predikovaných pravděpodobností pro třídy 0 a 1 pro první ze slov na Obr. 16:

	P(y = 0)	P(y = 1)
-----со_стороны------	[0,2	0,8]
---со_стороны--------	[0,3	0,7]
-со_стороны----------	[0,45	0,55]

V případě odhadu po řádcích by každý řádek dostal klasifikační hodnotu P(y = 1) (ve všech třech řádcích P(y = 1) > P(y = 0)), což ale ve skutečnosti není možné, protože by pak slovo obsahovalo tři přízvučné samohlásky. Proto když porovnáme pravděpodobnosti příslušnosti do třídy 1 mezi třemi řádky, největší pravděpodobnost bude mít první řádek a přízvučná je samohláska ‚o', druhá samohláska ‚o' a třetí ‚ы' spravně se označí jako nepřízvučné a jsou zařazeny do třídy 0.

5.2.1 Metody vyhodnocení

Vzhledem k malému počtu dat bylo zvoleno použití křížové validace pomocí Leave-One-Out. Leave-One-Out funguje tak, že z celého množství dat vždy vynecháme jednu položku na testování a ze zbytku se natrénuje klasifikátor. Přičemž se použijí všechny možné kombinace trénovacích a testovacích dat. Tímto způsobem je možné získat větší objektivitu výsledků (viz Obr. 17).

Výsledky jsou vyhodnoceny pomocí funkce **accuracy_score.** Accuracy je jedna z metod vyhodnocení výsledků, považuje se za nejjednodušší. Počítá se podle následujícího předpisu:

$$accuracy(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} 1(\hat{y}_i = y_i)$$

Zde je n počet predikovaných hodnot, $\hat{y}_i$ je predikovaná hodnota a y_i je příslušná pravdivá hodnota. Zjednodušeně řečeno accuracy udává procento správné predikce. Protože v předložené práci počítáme uspěšnost pro slova, jedná se o procento slov se správně určeným přízvukem.

Obrázek 17: Ukázka rozdělení dat pomocí Leave-One-Out

5.3 Trénování a testování neuronové sítě

Experimenty s neuronovou sítí byly založeny na příkladu z knihy Deep Learning With Python (22). Práce s neuronovými sítěmi a hluboké učení umožňuje knihovna Keras [9]. Dopředná neuronová síť pracuje s daty ve stejném formátu jako klasifikátor, to znamená, že byla použita data z kapitoly 5.1. V kontextu úkolu práce neuronová síť dostávala na vstup při každém průchodu jednu položku z předem definovaného vektoru trénovacích dat. Počet výstupů je jedna, protože řádce reprezentované vektorem velikosti abecedy pro testovaný soubor odpovídá jenom jedna hodnota – třída 0 nebo 1. Síť použitá v této práci má 4 vrstvy: vstupní, dvě skryté a výstupní (podobně jako na Obr. 10). To znamená, že počet neuronů první vrstvy je určen velikostí vektoru trénovacích dat, ale počet neuronů druhé vrstvy může být zvolen libovolně, proto při testování v druhé vrstvě bylo zvoleno 12 neuronů. Ve třetí vrstvě byl zvolen počet neuronů odpovídající počtu neuronů v první vrstvě. Poslední vrstva, výstupní, je představená jedním neuronem, protože na výstupu je třeba znát jen odhadnutou třídu.

Způsob inicializace vah pro všechny vrstvy byl zvolen „uniform“ (26), což je rovnoměrné rozdělení mezi 0 a 0,05 (defaultní pro Keras). Aktivační funkce pro skryté vrstvy je „relu“ (*angl.*

[9] Keras je otevřená knihovna neuronových sítí napsaná v jazyku Python. Je nadstavbou rámce Deeplearning4j, TensorFlow a Theano.

Rectified Linear Unit), která je předepsaná jako $f(x) = max(0, x)$. Vzhledem k tomu, že je třeba dostat výstup typu 0 nebo 1, aktivační funkce výstupní vrstvy je zvolena „sigmoid“, sigmadiagonální funkce.

Když už je síť popsaná, je třeba provést kompilaci pomocí funkce **compile()**. Daná funkce umožňuje zvolit ztrátovou funkci a způsob optimalizace vah a prahu. Za ztrátovou funkci byla zvolena logaritmická funkce binary_crossentropy. Optimalizace probíhá dle funkce „adam“ – algoritmus pro optimalizaci stochastických objektových funkcí založený na gradientu prvního řádu (defaultní pro Keras). Výsledky přesností klasifikace jsou sbírány do metriky „**metrics = [‚accuracy‘]**“.

Dále následuje již trénování modelu funkcí **fit()**, kterému poskytujeme vektory trénovacích dat a požadovaný výstup. Navíc poskytujeme také testovací data spolu s požadovaným výstupem pro ověření funkčnosti modelu po každém cyklu. Počet trénovacích cyklů je 150 a také nastavujeme počet trénovacích dvojic, po kterých se budou optimalizovat váhy. Výsledek výkonnosti sítí můžeme ocenit pomocí funkce **predict_proba()**, která dostává na vstup testovací data a požadovaný výstup a vrací pravděpodobnost příslušnosti ke třídě 1.

Výsledný proces trénování je znázorněn na obrázku 18.

```
400/400 [==============================] - 0s 219us/step - loss: 0.3172 - acc: 0.8150
Epoch 143/150
400/400 [==============================] - 0s 304us/step - loss: 0.3217 - acc: 0.8150
Epoch 144/150
400/400 [==============================] - 0s 264us/step - loss: 0.3322 - acc: 0.8025
Epoch 145/150
400/400 [==============================] - 0s 212us/step - loss: 0.3155 - acc: 0.8150
Epoch 146/150
400/400 [==============================] - 0s 292us/step - loss: 0.3235 - acc: 0.8050
Epoch 147/150
400/400 [==============================] - 0s 254us/step - loss: 0.3175 - acc: 0.8175
Epoch 148/150
400/400 [==============================] - 0s 249us/step - loss: 0.3364 - acc: 0.8000
Epoch 149/150
400/400 [==============================] - 0s 244us/step - loss: 0.3211 - acc: 0.8075
Epoch 150/150
400/400 [==============================] - 0s 234us/step - loss: 0.3236 - acc: 0.8175
298/298 [==============================] - 0s 974us/step
acc: 68.46%
```

Obrázek 18: Průběh trénování neuronové sítě

Kapitola 6

Výsledky

V této kapitole budou znázorněny výsledky 3 experimentu s klasifikátory popsanými výše a neuronovou sítí. Také bude představen počet dat trénovací a testovací sady. Z důvodu přehlednosti práce budou uvedeny jenom některé varianty délky kontextu, kompletní tabulky budou představené na přiloženém CD. Některé experimenty již byly prezentovány na studentské vědecké konferenci FAV v rocích 2017 (27) a 2018 (28).

6.1 Výsledky trénování a testování na ručně připravených datech

V daném experimentu byla využita metoda Leave-One-Out, princip který je popsán v sekci 5.2. Zde také byl proveden přepočet úspěšností pro celé slovo, ne pro každý řádek zvlášť (viz sekce 5.2). Úspěšnost je počítána metodou accuracy.

V následujících tabulkách jsou uvedeny výsledky klasifikace pro vybrané délky kontextu zleva 5, 20, 10 a 0; zprava 3, 5, 10, 0.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	86,00	88,00	88,00	71,00
Города	90,67	90,67	91,33	87,33
Дома	77,00	78,00	74,00	77,00
Места	75,00	73,00	69,00	53,00
Права	76,00	78,00	82,00	39,00
Самом	68,00	69,00	75,00	65,00
Слова	85,00	89,00	88,00	85,00
Стоит	79,00	70,00	74,00	65,00
Стороны	89,67	91,00	92,33	71,67
Страны	82,00	79,00	81,00	73,00

Průměr	80,83	80,57	81,47	68,70

Tabulka 6: Výsledky klasifikace Logistic Regression, 100 případů, %

V tabulce 6 jsou uvedeny výsledky klasifikace pro klasifikátor Logistic Regression pro testování na 100 ručně vybraných výskytu slova v textu. Poslední řádek v tabulce také znázorňuje průměrnou úspěšnost každého kontextu pro všechna slova. Z výsledků můžeme vyvodit, že největší úspěšnosti bylo dosaženo při délce kontextu L – 10 P – 10. Zajímavé zde bylo slovo „году", které prokazuje stejnou úspěšnost pro kontexty L – 20 P – 5 a L – 10 P – 10. Nejvyšší dosažené výsledky činí více než 75 %, maximálně 92,33 % při kontextu L – 10 P – 10 u slova „стороны". Nejmenší výsledek udává nulový kontext L – 0 P – 0 a to je 39,00 %. Nulový kontext v podstatě odpovídá situaci, při které je přednost určená statisticky. To znamená, že má přednost ta pozice, která se v trénovacích datech výskytuje častěji, bez ohledu na skutečný kontext, protože ho při trénování zanedbáváme. Logickým je předpoklad, že u nulového kontextu bude vždy vybraná ta samohláska, která je častěji přízvučná.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	92,50	92,00	92,50	85,00
Города	89,00	93,33	93,67	89,00
Дома	84,50	80,50	79,00	74,00
Места	82,50	80,00	79,00	53,50
Права	74,00	79,00	81,00	47,50
Самом	67,50	64,50	74,00	64,50
Слова	82,00	89,50	87,00	82,00
Стоит	80,75	78,75	81,25	63,25
Стороны	90,83	94,17	95,17	72,50
Страны	86,00	89,50	89,00	75,50
Průměr	82,96	84,13	85,16	70,68

Tabulka 7: Výsledky klasifikace Logistic Regression, 200 případů, %

Podivná se může zdát nejnižší úspěšností pro nulový kontext (39,00 %), tedy menší než 50 %, to ale souvisí se slovem „права", kde obě samohlásky jsou stejné – ‚a'. V tuto chvíli klasifikátor neví, jakou z nich má vybrat, a vybírá náhodně.

V tabulce 7 jsou uvedeny výsledky analogické tabulce 6, ale pro testování 200 ručně vybraných případů. Z tabulky lze usoudit, že nejlepší výsledky také byly dosaženy při délce kontextu L – 10 P – 10. Pozorujeme podobnou situaci jako u 100 případů se slovem „году", které prokazuje stejnou úspěšnost pro 2 různé kontexty, ale pro L – 20 P – 5 a L – 10 P – 10. Nejvyšší dosažené výsledky činí kolem 80 %, maximálně 95,17 %, analogické ke 100 případům, při kontextu L – 10 P – 10 u slova „стороны". Nejmenší úspěšnost můžeme znovu pozorovat u nulového kontextu, což je 47,50 %.

Při porovnání tabulek je i na první pohled zřejmé, že zvětšení počtu trénovacích dat se zlepšuje odhad. V případě všech slov, s výjimkou slova „самом", dominuje zlepšení. Když se obrátíme k průměrné úspěšnosti všech slov u každého kontextu (Tab. 8), je vidět, že výsledky se zlepšily a činí 80 % a více, nejvíce však 85,16 % (pro nenulový kontext).

Příklady	**L – 5 P – 5**	**L – 8 P – 4**	**L – 5 P – 3**	**L – 6 P – 5**	**L – 20 P – 5**	**L – 5 P – 8**	**L – 10 P – 10**	**L – 6 P – 3**	**L – 0 P – 0**
100x	80,73	80,93	80,83	79,77	80,57	80,83	81,47	79,93	68,70
200x	82,21	84,31	82,96	82,93	84,13	82,46	85,16	83,19	70,68

Tabulka 8: Průměrné výsledky klasifikace Logistic Regression, 100 a 200 případů, %

V tabulce 9 jsou uvedeny výsledky klasifikace pro klasifikátor Support Vector Machine s lineárním jádrem pro testování na 100 ručně vybraných případech. Podobně jako Logistic Regression i SVM udává nejlepší výsledky při délce kontextu L – 10 P – 10. Zde je také zajímavé, že slovo „стороны", které prokazuje stejnou úspěšnost při kontextech L – 20 P – 5 a L – 10 P – 10, zároveň udává maximální úspěšnost pro vybranou skupinu výsledků, tj. 91,67 %.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	80,00	89,00	88,00	71,00
Города	90,67	86,00	91,33	87,33
Дома	78,00	73,00	73,00	77,00
Места	57,00	73,00	71,00	40,00

Права	77,00	77,00	84,00	39,00
Самом	67,00	69,00	64,00	65,00
Слова	89,00	84,00	86,00	85,00
Стоит	78,00	70,00	77,00	65,00
Стороны	81,67	91,67	91,67	71,67
Страны	80,00	82,00	83,00	73,00
Průměr	77,83	79,47	80,90	67,40

Tabulka 9: Výsledky klasifikace SVM kernel = linear, 100 případů, %

Nejvyšší dosažené výsledky jsou kolem 80,00 % a výše. Nejmenší úspěšnost již očekávaně pozorujeme u nulového kontextu a je shodná s výsledky Logistic Regression, protože jak už bylo zmíněno výše, nulový kontext odpovídá situaci, při které je přesnost určená statisticky.

V tabulce 10 jsou uvedeny výsledky analogické tabulce 9, ale pro testování na 200 ručně vybraných případech. Zde stejně jako Logistic Regression udává nejlepší výsledky délka kontextu L – 10 P – 10. Slova „города“ a „страны“ prokazují stejnou úspěšnost zároveň ve 2 délkách kontextu - L – 20 P – 5 a L – 10 P – 10.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	92,50	95,50	93,00	85,00
Города	91,67	92,67	92,67	89,00
Дома	82,50	78,00	74,50	74,00
Места	72,50	79,50	75,50	47,50
Права	64,00	78,50	82,50	47,50
Самом	67,50	64,00	72,00	64,50
Слова	87,50	84,00	84,50	82,00
Стоит	78,25	78,75	79,75	63,25
Стороны	89,50	94,50	94,17	72,50
Страны	86,00	91,00	91,00	75,50

Průměr	81,19	83,64	83,96	70,08

Tabulka 10: Výsledky klasifikace SVM kernel = linear, 200 případů, %

Nejvyšší dosažené výsledky jsou již kolem 85 %, maximálně 95,50 % při kontextu L – 20 P – 5 u slova „году“ a nejmenší při nulovém kontextu - 47,50 %. V případě klasifikátoru SVM s lineárním jádrem, stejně jako u Logistic Regression, můžeme pozorovat tendenci zvýšení úspěšnosti s navýšením počtu dat. Výjimku tvoří slovo „слова“.

Pokud se podíváme na průměrnou úspěšnost všech slov u každého kontextu (Tab. 11), tak je zřejmé, že úspěšnost jednoznačně roste průměrně o 3 % s růstem počtu dat a činí 79,00 % a více, nejvíce však 84,69 %. Jak pro 100, tak i pro 200 případů je nejvyšší průměrná úspěšnost představena délkou kontextu L – 8 P – 4, což prokazuje podobnost klasifikace Logistic Regression, ale Logistic Regression celkově udává trochu lepší výsledky.

Příklady	**L – 5 P – 5**	**L – 8 P – 4**	**L – 5 P – 3**	**L – 6 P – 5**	**L – 20 P – 5**	**L – 5 P – 8**	**L – 10 P – 10**	**L – 6 P – 3**	**L – 0 P – 0**
100x	77,23	81,43	77,83	79,50	79,47	79,27	80,90	79,67	67,40
200x	83,04	84,69	81,19	82,81	83,64	81,94	83,96	82,34	70,08

Tabulka 11: Průměrné výsledky klasifikace SVM kernel = linear, 100 a 200 případů, %

V tabulce 12 jsou uvedeny výsledky klasifikace pro klasifikátor SVM s jádrem RBF pro testování na 100 ručně vybraných případech kontextu. Na rozdíl od předchozích dvou klasifikátorů, SVM s RBF jádrem prokazuje ne úplně typické chování, což může být spojeno s tím, že jsou trénovací data dobře lineárně separabilní. Daný typ klasifikátoru ale není určen pro lineárně separabilní data, i přesto bylo užitečné ho vyzkoušet. Také můžeme si všimnout, že daný typ klasifikátoru je výrazně pomalejší než ostatní. Je těžké posoudit, které kontexty udávají nejvyšší úspěšnost, ale výsledky se přiklánějí spíše k délce kontextu L – 5 P – 3.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	71,00	71,00	71,00	71,00
Города	87,33	87,33	87,33	46,00
Дома	77,00	77,00	77,00	77,00
Места	39,00	37,00	36,00	33,00

Права	61,00	45,00	52,00	39,00
Самом	65,00	65,00	65,00	65,00
Слова	85,00	85,00	85,00	85,00
Стоит	65,00	65,00	65,00	65,00
Стороны	83,00	84,33	84,33	71,67
Страны	82,00	73,00	73,00	73,00
Průměr	71,53	68,97	69,57	62,57

Tabulka 12: Výsledky klasifikace SVM kernel = rbf, 100 případů, %

Celková tendence je, že úspěšnost je ze všech 3 klasifikátorů nejnižší. Nejvyšší dosažená úspěšnost je 87,33 % pro všechny kontexty, kromě nulového, u slova „города“.

V tabulce 13 jsou uvedeny výsledky analogické tabulce 12, ale pro testování na 200 ručně vybraných případech kontextu. Analogicky k tabulce 12 nejsou výsledky moc charakteristické, nicméně je vidět malou dominanci kontextu L – 5 P – 3. Nejvyšší dosažená úspěšnost je 89,00 % pro všechny kontexty (zanedbáváme nulový) u slova „году“. Při porovnání tabulek si na první pohled můžeme všimnout, že v daném případě, oproti předchozím, při zvětšení počtu trénovacích dat se odhad naopak může i zhoršovat.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**	**L – 0 P – 0**
Году	85,00	85,00	85,00	85,00
Города	89,00	89,00	89,00	74,00
Дома	74,00	74,00	74,00	74,00
Места	69,00	53,00	53,50	50,00
Права	46,00	46,00	47,50	47,50
Самом	64,50	64,50	64,50	64,50
Слова	82,00	82,00	82,00	82,00
Стоит	63,25	63,25	63,25	63,25
Стороны	86,50	85,83	86,50	72,50

Страны	86,50	75,50	75,50	75,50
Průměr	74,58	71,81	72,08	68,83

Tabulka 13: Výsledky klasifikace SVM kernel = rbf, 200 případů, %

Příklady	**L – 5 P – 5**	**L – 8 P – 4**	**L – 5 P – 3**	**L – 6 P – 5**	**L – 20 P – 5**	**L – 5 P – 8**	**L – 10 P – 10**	**L – 6 P – 3**	**L – 0 P – 0**
100x	70,53	72,47	71,53	72,63	68,97	70,57	69,57	73,83	62,57
200x	71,76	75,13	74,58	74,44	71,81	72,04	72,08	75,38	68,83

Tabulka 14: Průměrné výsledky klasifikace SVM kernel = rbf, 100 a 200 případů, %

Když se podíváme na průměrnou úspěšnost všech slov u každého kontextu (Tab. 14), je vidět, že úspěšnost s rostoucíme počtem dat roste o 1 až 3 %, průměrně se pohybuje od 69 % a výše, nejvyšší je ale jen 75,38 % při kontextu L – 6 P – 3.

Z tabulek 8, 11 a 14 můžeme odvodit, že pokud zanedbáme nulové kontexty, průměrná úspěšnost trénování klasifikátorů se nachází mezi 70 a 85 %. Také můžeme pozorovat lepší odhad při zvýšení počtu trénovacích dat. Pokud se důkladně podíváme na průměrné výsledky, můžeme si všimnout tendence k růstu úspěšnosti odhadu při určité délce kontextu, zejména zleva. Přitom kontext zleva nemá být moc dlouhý - 6-8 znaků od samohlásky. Taková závislost se vztahuje k tomu, že často predikované slovo je závislé na koncovce slova, které se nachází přímo před ním. Koncovka totiž v mnoha případech rozhoduje o tom, v jakém pádě se slovo nachází, tudíž i slovo na něm závislé. Proto 6-8 znaků činí ten důležitý kontext a více znaků od samohlásky už je navíc a nepřináší většinou žádný užitek pro trénovací informaci.

6.2 Výsledky trénování na ručně připravených datech a testování na RNC

V dané části metoda Leave-One-Out už nebyla třeba, protože data jsou jednoznačně separována na trénovací a testovací. Zde stejně jak v předchozím experimentu byl proveden přepočet úspěšností pro celé slovo (viz sekce 5.2). Úspěšnost je počítaná metodou accuracy.

Podíváme se opět pouze na výsledky klasifikace pro vybrané délky kontextu zleva 5, 20 a 10; zprava 3, 5 a 10, kompletní výsledky pro všechny testované kontexty jsou k dispozici na přiloženém CD. Zde už nebudeme zkoumat nulový kontext, protože jeho výsledky jsou uměrné frekvenci výskytu příslušných variant přízvuku uvedených v kapitole 5.1.1, tabulka 4.

Jak už bylo řečeno v bodě 5.1, zde bylo pro trénování jak klasifikátorů, tak neuronové sítě (struktura je popsaná v bodě 5.3) použito 200 ručně vybraných příkladů a testování se provádělo na části RNC. Statistika výskytu jednotlivých slov a přízvuků je uvedena v Tab. 4. Tabulka 15 představuje relace trénovacích a testovacích dat.

V tabulce 16 jsou uvedeny výsledky klasifikace pro klasifikátor Logistic Regression pro trénování na 200 ručně vybraných případů a testování vybraných dat z RNC. Také poslední řádek v tabulce znázorňuje průměrnou úspěšnost každého kontextu pro všechna slova. Oproti prvnímu experimentu lepší úspěšnost prokazuje délka kontextu L – 5 P – 3. Ale obdobně k prvnímu experimentu si můžeme všimnout stejných výsledků pro slova „города“ a „места“ zároveň u 2 kontextů. Nejvyšší dosažené výsledky jsou již kolem 83 %, maximální 98,08 % u slova „году“ při kontextu L – 5 P – 3 a minimum u slova „страны“ 28,04 % u kontextu L – 5 P – 3. Tak nízká úspěšnost může být spojena s tím, že i první samohláska ve slově „страны“ je o 3 znaky vzdálena od začátku slova, zatímco u ostatních slov je to 1 až 2 znaky, takže při kontextu zleva o délce 5 není zachycen skoro žádný kontext.

Slova	Trénovací	Testovací
Году	400	832
Города	600	378
Дома	400	496
Места	400	298
Права	400	166
Самом	400	674
Слова	400	496
Стоит	400	542
Стороны	600	1059
Страны	400	514

Tabulka 15: Počet trénovacích a testovacích dat pro učení a vyhodnocování

Slovo	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	98,08	97,12	97,12
Города	94,71	94,71	94,18
Дома	81,85	76,61	76,21
Места	78,52	82,55	82,55
Права	68,67	66,27	65,06
Самом	91,39	93,18	90,50
Слова	63,71	74,19	77,02
Стоит	67,16	65,31	59,78
Стороны	97,36	95,85	96,98
Страны	28,40	64,59	62,65
Průměr	76,99	81,04	80,20

Tabulka 16: Výsledky klasifikace Logistic Regression, testování na RNC, %

V tabulce 17 jsou znázorněny výsledky klasifikace pro klasifikátor Support Vector Machine s lineárním jádrem pro trénování na 200 ručně vybraných případech a testování na vybraných datech z RNC. Zde obdobně jako v experimentu 6.1 lepší úspěšnost udává délka přízvuku L – 10 P – 10. Nejvyšší dosažené výsledky jsou již kolem 82 %. Maximální výsledek je stejný jako u Logistic Regression, a to je 98,08 % u slova „году“ při kontextu L – 5 P – 3, a minimální u slova „страны“ 23,35 % u kontextu L – 5 P – 3.

Slova	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	98,08	97,60	97,60
Города	96,30	91,01	92,06
Дома	81,85	72,18	74,19
Места	78,52	81,88	79,87
Права	65,06	62,65	69,88

Самом	91,10	92,28	78,34
Слова	77,02	74,19	77,82
Стоит	67,53	63,10	59,41
Стороны	86,40	94,15	96,41
Страны	23,35	68,48	74,32
Průměr	76,52	79,75	79,99

Tabulka 17 Výsledky klasifikace SVM kernel = linear, testování na RNC, %

V tabulce 18 jsou znázorněny výsledky klasifikace pro klasifikátor Support Vector Machine s RBF jádrem pro trénování na 200 ručně vybraných případech a testování na vybraných datech z RNC. Zde podobně jako v experimentu 6.1 nemůžeme mluvit o jednoznačných výsledcích. Ale tendence lepší úspěšnosti je u kontextu L – 5 P – 3 a L – 10 P – 10. Nejvyšší dosažené výsledky jsou již kolem 78 %. Maximální výsledek je 96,88 % u slova „году“ pro všechny kontexty a minimální u slova „страны“ 23,74 % u kontextu L – 5 P – 3.

Slova	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**
Году	96,88	96,88	96,88
Города	92,59	92,59	92,59
Дома	78,63	78,63	78,63
Места	66,44	59,73	59,73
Права	40,96	39,76	39,76
Самом	95,55	95,55	95,55
Слова	63,71	63,71	63,71
Стоит	52,03	52,03	52,03
Стороны	92,82	94,90	96,41
Страны	23,74	76,65	76,65
Průměr	70,34	75,04	75,19

Tabulka 18: Výsledky klasifikace SVM kernel = rbf, testování na RNC, %

Tabulka 19 znázorňuje výsledky, které při stejných podmínkách, jaké měly klasifikátory, udává neuronová síť. Zde jde o velice podobný, ale jiný postup ve strojovém učení. Větší úspěšnost udává délka přízvuku L – 20 P – 5. Nejvyšší dosažené výsledky jsou již kolem 84 %. Maximální výsledek je podobný jako u Logistic Regression, a to je 97,36 % u slova „году“ při kontextu L – 5 P – 3, a minimální u slova „страны“ 45,40 % u kontextu L – 20 P – 5.

Slova	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	97,36	96,15	97,12
Города	89,95	93,65	91,01
Дома	75,40	77,82	73,39
Места	79,87	79,19	80,54
Права	56,63	71,08	65,06
Самом	83,38	45,40	91,39
Слова	78,63	74,19	84,27
Стоит	66,42	63,47	59,41
Стороны	92,07	95,09	92,45
Страны	51,75	84,82	82,49
Průměr	77,14	78,09	81,71

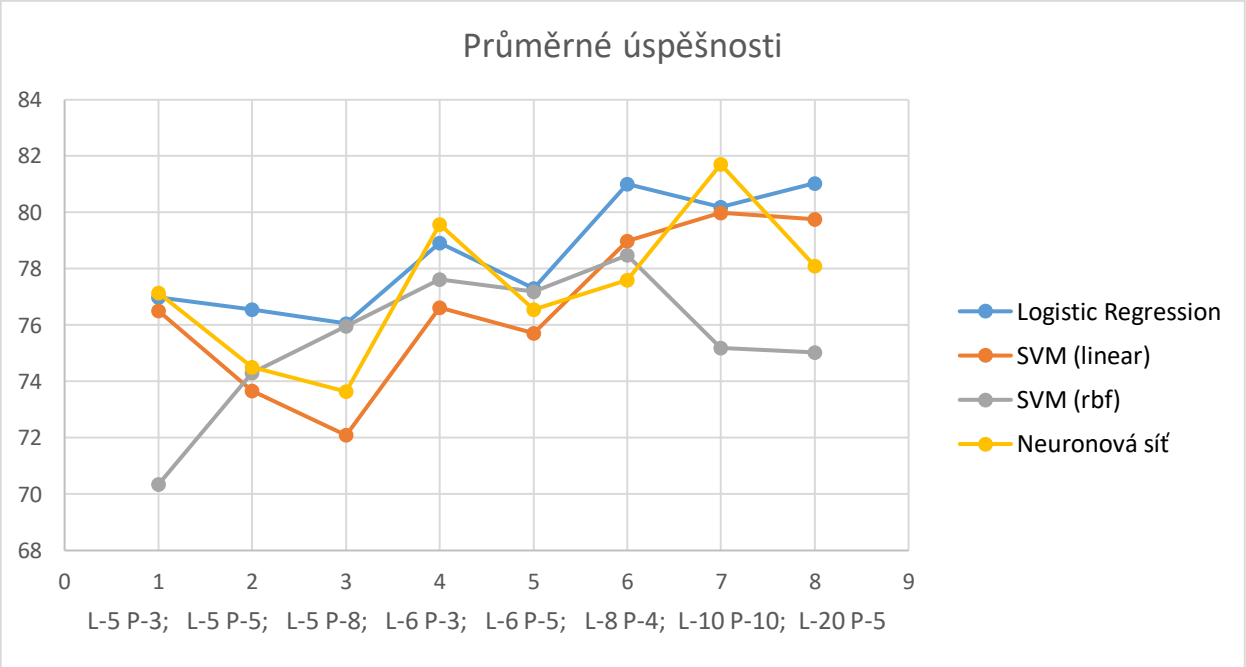
Tabulka 19: Výsledky trénování neuronové sítě, testování na RNC, %

Případy kontextů	L – 5 P – 5	L – 8 P – 4	L – 5 P – 3	L – 6 P – 5	L – 20 P – 5	L – 5 P – 8	L – 10 P – 10	L – 6 P – 3
Logistic Regression	76,56	81,00	76,99	77,31	81,04	76,06	80,20	78,92
SVM, linear	73,66	78,98	76,52	75,71	79,75	72,09	79,99	76,63
SVM, rbf	74,30	78,48	70,34	77,20	75,04	75,96	75,19	77,62
Neuronová síť	74,52	77,59	77,14	76,56	78,09	73,64	81,71	79,57

Tabulka 20: Průměrné výsledky pro všechna slova, testování na RNC, %

Pokud se podíváme na tabulku 20, můžeme získat přehled o průměrných výsledcích pro jednotlivé kontexty pro všechna slova. Ve většině případů o lepší výsledky konkurují klasifikátor Logistic Regression, který udává nejlepší výsledky pro 5 kontextů: L – 5 P – 5; L – 8 P – 4; L – 6 P – 5; L – 20 P – 5; L – 5 P – 8; a neuronová síť, která udává nejlepší výsledky pro 3 kontexty L – 5 P – 3; L – 10 P – 10; L – 6 P – 3.

Můžeme také vyvodit, že se průměrná úspěšnost trénování nachází mezi 70 a 80 %. Zde podobně jako v kapitole 6.1 lze sledovat růst úspěšnosti odhadu při určité délce přízvuku zejména zleva. Změnu úspěšnosti s růstem počtu znaků zleva (následně zprava) znázorňuje Obrázek 19. Na něm je dobře vidět, že každý ze způsobů predikce při některé délce kontextu dosahuje svého maxima, po kterém následuje pokles. Výjimkou by mohl být klasifikátor Logistic Regression, který při zvětšení kontextu do L – 8 P – 4 dosahuje maxima – 81,00 %, potom klesá, ale při zvětšení kontextu do největší délky L – 20 P – 5 dosahuje o 0,04 % větší hodnoty - 81,04 %.



Obrázek 19: Graf průměrných výsledků pro všechna slova, při vzrůstu délky kontextu, pro 4 způsoby predikce - Logistic Regression; SVM, linear; SVM, rbf; neuronová síť

Obecně můžeme říct, že nejlepší výsledky lze získat pomocí klasifikátoru Logistic Regression a neuronové sítě, o něco horší je klasifikátor Support Vector Machine s lineárním jádrem, rbf jádro oproti němu udává ještě nižší výsledky, a k tomu je navíc časově náročnější.

6.3 Výsledky zobecněného trénování

Cílem třetího experimentu bylo zjistit úspěšnost jednotlivých metod strojového učení při trénování na 9 slovech z 10 vybraných a následném testování na zbývajícím (desátem) slově. V tomto

experimentu jako i v experimentu v kapitole 6.2 byla pro trénování použita pouze ručně vybraná data a pro testování byla využita data z RNC. Následující tabulky opět uvádí úspěšnost přepočtenou pro celé slovo (viz sekce 5.2).

Podíváme se na výsledky klasifikace pro vybrané délky kontextu zleva 5, 20 a 10; zprava 3, 5 a 10. V tabulce 21 jsou uvedeny výsledky pro klasifikátor Logistic Regression. Z výsledků je vidět, že zobecněné trénování není tak úspěšné jako trénování zaměřené na konkrétní slovo, ale i přesto udává zajímavé výsledky. Lepší výsledky klasifikátor udává pro kontext L – 5 P – 3. Průměrná úspěšnost klasifikátorů činí 50 %. Nejvyšší dosažená úspěšnost činí 86,65 % pro slovo „самом" a kontext L – 5 P – 3. Také můžeme pozorovat i velmi malou úspěšnost pro slovo „году". To může být vysvětleno tím, že dvojice samohlásek v daném slově je „о" a „у". Samohláska „о" je obsažena i v jiných slovech, ale samohláska „у" se v žádném jiném slově nevyskytuje. Podobně je tomu u slova „места", které má také samohlásku „е", která se nevyskytuje nikde jinde, ale přesto nevykazuje tak nízké hodnoty úspěšnosti. To je možná také ovlivněno tím, že počet trénovacích dat v případě slova „году" je dvakrát menší než testovacích, zatímco u slova „места" je trénovacích dat skoro 1,5x více než testovacích.

Slovo	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	3,13	5,29	6,25
Города	41,27	41,27	42,33
Дома	52,82	62,50	62,10
Места	59,73	63,09	62,42
Права	49,40	50,60	56,63
Самом	86,65	86,35	85,16
Слова	36,29	37,50	36,29
Стоит	48,34	45,39	46,86
Стороны	39,57	39,00	38,05
Страны	76,65	64,59	71,98

Tabulka 21: Výsledky klasifikace Logistic Regression, trénování na 9 slovech, testování na 10., %

Slovo	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	3,13	13,70	4,57
Города	40,74	41,80	40,74
Дома	50,40	59,68	63,31
Места	71,14	61,74	61,07
Права	40,96	50,60	46,99
Самом	69,73	77,15	82,79
Слова	36,29	36,69	36,69
Стоит	46,86	47,60	48,71
Стороны	86,59	41,08	36,54
Страны	76,65	54,86	56,81

Tabulka 22: Výsledky klasifikace SVM kernel = linear, trénování na 9 slovech, testování na 10., %

Tabulka 22 poskytuje výsledky pro klasifikátor SVM s lineárním jádrem. Oproti předchozímu klasifikátoru se lepší výsledky shodují v kontextech L – 20 P – 5 a L – 10 P – 10. Průměrná úspěšnost, stejně jako u Logistic Regression, se pohybuje kolem 50 %. Nejvyšší dosažená úspěšnost činí 86,59 % pro slovo „стороны“ a kontext L – 5 P – 3.

V tabulce 23 jsou výsledky pro klasifikátor SVM s rbf jádrem. Obdobně jako výsledky z kapitoly 6.1 a 6.2 tento klasifikátor neprokazuje moc dobré výsledky z důvodu popsaném v kapitole 6.1. Průměrná úspěšnost daného klasifikátoru je kolem 47 %. Nejvyšší dosažená úspěšnost činí 95,55 % pro slovo „самом“ pro všechny kontexty.

Slovo	L – 5 P – 3	L – 20 P – 5	L – 10 P – 10
Году	3,13	3,13	3,13
Города	40,74	40,74	40,74
Дома	24,19	39,92	39,92
Места	59,73	58,39	57,72
Права	40,96	40,96	40,96
Самом	95,55	95,55	95,55

Слова	35,89	36,29	36,69
Стоит	47,97	45,39	46,86
Стороны	39,57	37,49	35,98
Страны	76,65	75,88	75,88

Tabulka 23: Výsledky klasifikace SVM kernel = rbf, trénování na 9 slovech, testování na 10., %

Tabulka 24 uvádí výsledky trénování neuronové sítě se strukturou, která je stejná jako struktura v kapitole 6.2. Nejlepší výsledky neuronová síť udává pro kontext L – 10 P – 10, průměrná úspěšnost činí 55 %. Nejvyšší dosažená úspěšnost je ale nižší - 79,82 % pro slovo „самом“ a kontext L – 10 P – 10.

Slovo	**L – 5 P – 3**	**L – 20 P – 5**	**L – 10 P – 10**
Году	4,81	27,40	34,13
Города	44,44	77,25	77,25
Дома	37,50	59,27	54,44
Места	73,15	62,42	64,43
Права	49,40	45,78	48,19
Самом	79,23	69,14	79,82
Слова	36,69	43,55	53,23
Стоит	40,22	48,71	52,77
Стороны	33,71	57,70	49,76
Страны	72,76	69,65	58,37

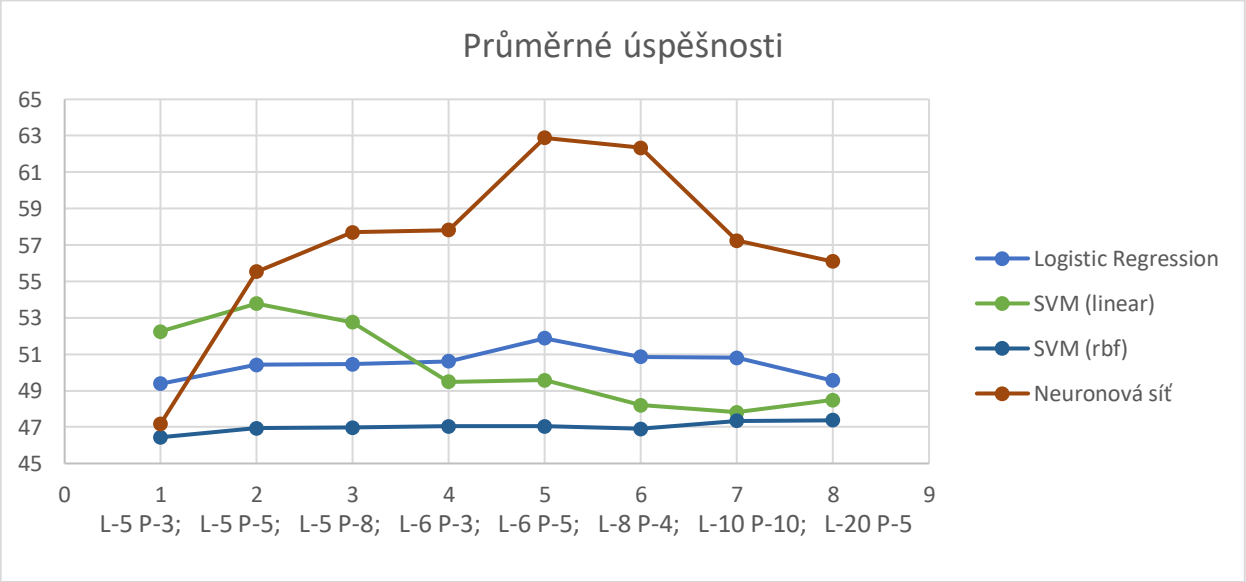
Tabulka 24: Výsledky trénování neuronové sítě, trénování na 9 slovech, testování na 10., %

Tabulka 25 znázorňuje průměrné výsledky pro všechna slova pro každou z vybraných metod strojového učení. Na závěr tohoto experimentu poznamenejme, že nejvyšší úspěšnost prokazuje neuronová síť – 62,88 % pro kontext L – 6 P – 5, která ve své podstatě umí lépe zobecňovat. Oproti prvním dvěma experimentům je průměrná úspěšnost v rozmezí 47 až 63 %. Tento průměr je skoro o 20 % menší než v předchozích experimentech, ale vzhledem k tomu, jak se úloha zkomplikovala, jedná se o relativně dobrý výsledek.

Případy kontextu	L – 5 P – 5	L – 8 P – 4	L – 5 P – 3	L – 6 P – 5	L – 20 P – 5	L – 5 P – 8	L – 10 P – 10	L – 6 P – 3
Logistic Regression	50,42	50,86	49,38	51,87	49,56	50,45	50,81	50,62
SVM, linear	53,78	48,20	52,25	49,57	48,49	52,75	47,82	49,49
SVM, rbf	46,94	46,91	46,44	47,05	47,37	46,98	47,34	47,04
Neuronová síť	55,53	62,34	47,19	62,88	56,09	57,70	57,24	57,82

Tabulka 25: Průměrné výsledky pro všechny slova, %

Změnu úspěšností s růstem počtu znaků zleva (následně zprava) můžeme sledovat na Obrázku 20. Na něm je dobře vidět, že každý ze způsobů predikce při některé délce kontextu dosahuje svého maxima, po kterém následuje pokles. Výjimkou je klasifikátor SVM s lineárním jádrem, který dosahuje svého maxima při nejkratší délce kontextu L – 5 P – 5 – 53,78 %, potom klesá. Ale také je dobře vidět poměrně velký růst úspěšnosti u neuronové sítě a také poměrné rychlý pokles, po dosaženi optimální délky přízvuku L – 6 P – 5.



Obrázek 20 Graf průměrných výsledků obecného trénování přes všechny slova, při vzrůstů délky kontextu, pro 4 způsoby predikce - Logistic Regression; SVM, linear; SVM, rbf; Neuronová síť

Tento experiment již částečně simuluje úlohu obecného trénováni klasifikátoru či neuronové sítě na velkém množství vstupních dat. Experiment potvrzuje možnost natrénování obecného prediktoru pozice slovního přízvuku v ruském jazyce.

6.4 Shrnutí výsledků

Postupně byly provedeny 3 experimenty. První (výsledky viz kapitola 6.1) byl zaměřen na trénování a testování (predikci) pouze na ručně vybraných datech, predikce byla provedena klasifikátory Logistic Regression a Support Vector Machine. Cílem prvního experimentu bylo zhodnotit úspěšnost predikce pozice přízvuku v testovaném slově a také stanovit změnu výsledků při dvojnásobném navýšení počtu trénovacích dat. Z důvodu poměrně malého počtu dat byla pro křížovou validaci dat využita metoda Leave-One-Out.Výsledky prokázaly, že průměrná úspěšnost je v rozmezí 70 až 85 % a nejlepší odhad dělá klasifikátor Logistic Regression. Bylo pozorováno zlepšení odhadu při zvýšení počtu trénovacích dat. Projevila se také tendence k růstu úspěšnosti odhadu při určité délce kontextu, zejména zleva - 6 až 8 znaků od samohlásky. Taková závislost může být vysvětlena tím, že často je predikované slovo závislé na koncovce slova, které se nachází přímo před ním, protože koncovka v mnoha případech rozhoduje o tom, v jakém pádě se nachází dané slovo, a tudíž i slovo na něm závislé.

Druhý experiment byl zaměřen na trénování na ručně vybraných datech a testování na vybraném množství dat z Ruského národního korpusu. Predikce byla provedena klasifikátory (viz výše) a neuronovou sítí. Cílem bylo zjistit úspěšnost predikce oproti prvnímu experimentu, kde trénovací a testovací data byla všechna získána ručně. Zde lepší výsledky také ukázal klasifikátor Logistic Regression, který udává nejlepší výsledky pro 5 kontextů, ale druhé nejlepší výsledky ukazuje neuronová síť, která udává nejlepší výsledky pro 3 kontexty. V druhém experimentu se průměrná úspěšnost při testování nachází mezi 70 a 80 %. Zde, podobně jako v prvním experimentu, byl sledován růst úspěšnosti odhadu při určité délce kontextu (zejména zleva) a následný pokles úspěšnosti odhadu při zbytečně velké délce kontextu.

Ve třetím experimentu bylo cílem zjistit úspěšnost jednotlivých metod strojového učení při jejich trénování na 9 slovech z 10 vybraných a následném testování na zbývajícím 10. slově. V tomto experimentu (stejně jako i ve druhém) byli pro trénování použity pouze ručně vybraná data a pro testování byly využity data z RNC (pro desáté testované slovo). U zobecněného trénování byla očekávána nižší úspěšnost než u trénování zaměřeného na konkrétní slovo, ale i přesto bylo dosaženo pro některé délky kontextů průměrné úspěšnosti větší než 60 %. U úkolu zobecňování prokázala očekávaně lepší úspěšnost neuronová síť.

Kapitola 7

Závěr

Tato bakalářská práce je věnována návrhu systému automatizace detekce přízvuku v ruském jazyce, konkrétně detekce přízvuku u homografů. V rámci řešení daného problému jsme teoreticky rozebrali principy, na kterých je postaven systém TTS, neboli syntezátor řeči. Bylo vysvětleno, že přízvuk je jedna z hlavních vlastností slova. Každý jazyk z historických důvodů svého vývoje přistupuje k různorodosti přízvuku jinak, například v angličtině a ruštině je přízvuk různorodý, ale v češtině, polštině a francouzštině není. K různorodosti se vztahuje pojem pohyblivosti přízvuku, což je důvodem pro jeho detekci u homografů. K řešení daného problému byly použity metody strojového učení – klasifikátory Logistic Regression a Support Vector Machine se 2 různými jádrovými funkcemi (linear a rbf) a jednoduchá neuronová síť. Jako příznaky pro trénování byl použit pouze text, tj. vybraná okolní slova.

Z výsledků experimentu lze usoudit, že postup nabízený v této práci obecně funguje nejlépe u klasifikátoru Logistic Regression a získané úspěšnosti dosahují více než 80 %, převážně s delším levým textovým kontextem daného slova. Dobrých výsledků lze také dosáhnout i použitím jednoduché neuronové sítě, převážně v úloze obecného trénovaní a to téměř 63 %. Obecné trénování je potřeba v reálných systémech syntézy řeči z textu. O něco horší výsledky jsme dostali od klasifikátoru SVM (linear). Nejslabší výsledky ukázal klasifikátor SVM (rbf), navíc byl ze všech testovaných klasifikátorů nejpomalejší, což ho dělá nevhodným pro použití oproti výše zmíněným. Postup navržený v této práci by při jeho aplikaci mohl mít pozitivní přínos a jednoznačně vylepšit přesnost a přirozenost TTS.

Literatura

1. Кратценштейн Христиан Готлиб — Биография (Christian Gottlieb Kratzenstein - Biografie). *Помни Про. Электронный мемориал (Pamatuj si. Elektronicky memoriál).* [Online] [Citace: 01. 05 2019.] http://pomnipro.ru/memorypage41160/biography.

2. Matoušek, J. Učební texty z předmětu Úvod do strojového vnímání prostředí. Plzeň : ZČU, 2017.

3. Psutka, J.; Pražák, A. Učební texty z předmětu Analýza a rozpoznávání řeči. Plzeň : ZČU, 2018.

4. Dutoit, T. *An Introduction to Text-to-Speech Synthesis.* Dordrecht : Kluwer Academic Publishers, 1997.

5. Psutka, J.; Müller, L.; Matoušek, J.; Radová, V. *Mluvíme s počítačem česky.* Praha : Academia, 2006. stránky 616-631.

6. Сколько слов в русском языке? (Kolik je slov v ruském jazyce?). *Science Debate научно-популярный блог (Science Debate populární vědecký blog).* [Online] [Citace: 22. 01 2019.] http://www.sciencedebate2008.com/how-many-russian-words/.

7. Těšitelová, M. *O češtině v číslech.* místo neznámé: Academia, 1987.

8. Psutka, J. Učební texty z předmětu Úvod do strojového vnímání prostředí. Plzeň : ZČU, 2017.

9. Bobrov, S.P. Prozódie, Literární encyklopedia: Slovník literárních pojmů: Ve dvou dílech. [Online] 1925. [Citace: 15. 10 2018.] http://feb-web.ru/feb/slt/abc/lt2/lt2-6671.htm.

10. Palková, Z. Intonace v popisu prozódie. Nový encyklopedický slovník češtiny. [Online] 18. 10 2018. https://www.czechency.org/slovnik/INTONACE V POPISU PROZODIE.

11. Avanesov, R. *O přízvuku v ruském jazyce.* Moskva: Rusky jazyk ve škole, 1984.

12. Skarnitzl, R.; Šturm, P.; Volín, J. Zvuková báze řečové komunikace. 2016.

13. Kadashevsky, S. *Diference slovních významů podle umístění přízvuku.* 1946.

14. Arthur, S. *Some Studies in Machine Learning Using the Game of Checkers.* 1959. stránky 210-229.

15. Bishop, C. *Pattern Recognition and Machine Learning.* 2006.

16. Kohavi, R.; Provost, F. *Glossary of terms. Machine Learning.* 1998. stránky 271-274.

17. Kropotov, D. a Vetrov, D. Лекция 1 Различные задачи машинного обучения (Přednáška 1 Různé úlohy strojového učení). *http://www.machinelearning.ru.* [Online] MSU. [Citace: 27. 02 2019.] http://www.machinelearning.ru/wiki/images/a/a1/BayesML-2009-1.pdf.

18. Psutka, J. Učební texty z předmětu Základy strojového učení a rozpoznávaní. Plzeň: ZČU, 2017.

19. Cox, D. *The Regression Analysis of Binary Sequences.* místo neznámé: Journal of the Royal Statistical Society. Series B (Methodological), 1958.

20. Vyugin, V. *Matematické základy teorie strojového učení a prognózování.* Moskva: MCNMV, 2013. stránky 88-105.

21. Radová, V. Učební texty z předmětu Neuronové sítě a evoluční strategie. Plzeň: ZČU, 2018.

22. Brownlee, J. *Deep Learning With Python.* 2018.

23. Национальный корпус русского языка. Состав и структура (Národní korpus ruského jazyka. Složení a struktura). *Национальный корпус русского языка (Národní korpus ruského jazyka).* [Online] [Citace: 09. 03 2019.] http://www.ruscorpora.ru/corpora-progr.html.

24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. [Online] 2011. [Citace: 19. 05 2019.]

25. Voroncov, K. Přednášky na téma SVM. *ВЦ РАН им. А.А. Дородницына Федерального исследовательского центра «Информатика и управление» РАН.* [Online] 21. 12 2007. [Citace: 09. 03 2019.] http://www.ccas.ru/voron/download/SVM.pdf.

26. *Keras: The Python Deep Learning library.* [Online] [Cited: 03 24, 2019.] http://keras.io.

27. Chizhova, A. *Detekce přízvuků v ruštině s použitím klasifikátoru.* Plzeň : Západočeská univerzita v Plzni, 2017.

28. —. *Detekce přízvuků na datech z Russian National Corpus.* Plzeň : Západočeská univerzita v Plzni, 2018.

Obsah přiloženého CD

Na přiloženém CD je text této bakalářské práce v elektronické podobě, skripty k jednotlivým experimentům a vstupní soubory pro experiment č.1. Experimenty č.2 a č.3 byli provedeny s použitím Ruského národního korpusu, který může být získán na žádost pro nekomerční použití na jejich stránkách (23). CD rovněž obsahuje soubory se získanými výsledky.

Složka "Skripty" – obsahuje všechny skripty vytvořené v rámci této bakalářské práce:

- Priprava_dat.py – kód určený pro přípravu trénovacích a testovacích dat ze vstupních souborů
- seznam_funkci.py – obsahuje všechny funkce, které byly použité v hlavní částí programu
- Prvni_experiment.py – kód prvního experimentu, trénovaní a testování na ručně vybraných datech metodou Leave-One-Out
- Druhy_a_treti_experiment_klasifikatory.py – kód druhého a zároveň třetího experimentu. Kód je určen pro klasifikátory, konkrétní klasifikátor se volí odkomentováním jednoho z řádků v hlavním programu. V programu probíhá trénováni na ručně vybraných datech a testování na datech z RNC.
- Druhy_a_treti_experiment_NN.py – kód druhého a zároveň třetího experimentu s použitím neuronové sítě. V programu probíhá trénování na ručně vybraných datech a testování na datech z RNC.
- 9_slov.py – pomocný kód k experimentu č.3, slouží k vytvoření všech možných kombinací trénovacích a testovacích slov pro vybrané kontexty.

Složka „Vstupní soubory" – obsahuje soubory s ručně vybranými daty pro trénování

- Году100.txt
- Города100.txt
- Дома100.txt
- Места100.txt
- Права100.txt
- Самом100.txt
- Слова100.txt
- Стоит100.txt
- Стороны100.txt
- Страны100.txt
- Году200.txt
- Города200.txt
- Дома200.txt
- Места200.txt
- Права200.txt
- Самом200.txt
- Слова200.txt
- Стоит200.txt
- Стороны200.txt
- Страны200.txt

Složka "Výsledky 1. experimentu" - obsahuje 6 tabulek v excelu s kompletními výsledky 1. experimentu:

- vysledky_LogReg_100.xlsl – Výsledky klasifikace Logistic Regression, 100 případů
- vysledky_LogReg_200.xlsl – Výsledky klasifikace Logistic Regression, 200 případů
- vysledky_SVM_linear_100.xlsl – Výsledky klasifikace SVM kernel = linear, 100 případů
- vysledky_SVM_linear_200.xlsl – Výsledky klasifikace SVM kernel = linear, 200 případů
- vysledky_SVM_rbf_100.xlsl – Výsledky klasifikace SVM kernel = rbf, 100 případů
- vysledky_SVM_rbf_200.xlsl – Výsledky klasifikace SVM kernel = rbf, 200 případů

Složka "Výsledky 2. experimentu" - obsahuje 4 tabulky v excelu s kompletními výsledky 2. experimentu:

- vysledky_LogReg.xlsl – Výsledky klasifikace Logistic Regression
- vysledky_SVM_linear.xlsl – Výsledky klasifikace SVM kernel = linear
- vysledky_SVM_rbf.xlsl – Výsledky klasifikace SVM kernel = rbf
- vysledky_NN.xlsl – Výsledky trénování neuronové sítě

Složka "Výsledky 3. experimentu" - obsahuje 4 tabulky v excelu s kompletními výsledky 3. experimentu:

- vysledky_LogReg.xlsl – Výsledky klasifikace Logistic Regression
- vysledky_SVM_linear.xlsl – Výsledky klasifikace SVM kernel = linear
- vysledky_SVM_rbf.xlsl – Výsledky klasifikace SVM kernel = rbf
- vysledky_NN.xlsl – Výsledky trénování neuronové sítě