VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIZERTAČNÁ PRÁCA

Brno, 2018 Ing. LUKÁŠ POVODA

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

DOLOVANIE ZNALOSTÍ Z TEXTOVÝCH DÁT POUŽITÍM METÓD UMELEJ INTELIGENCIE
TEXT MINING BASED ON ARTIFICIAL INTELLIGENCE METHODS

DIZERTAČNÁ PRÁCA
DOCTORAL THESIS

AUTOR PRÁCE — **Ing. Lukáš Povoda**
AUTHOR

VEDÚCI PRÁCE — **doc. Ing. Radim Burget, Ph.D.**
ADVISOR

BRNO 2018

ABSTRAKT

Práca sa zaoberá problémom dolovania znalostí z textových dát, ktorý je stále aktuálnejší vzhľadom na exponenciálny rast množstva uložených dát v elektronickej podobe, kde 80% týchto dát je v textovej podobe. Práca skúma súčasné metódy, ich možné zvýšenie presnosti vďaka optimalizačným metódam, ako aj nové metódy riešenia problému porozumenia textu s modelovaním kognitívneho správanie človeka pri spracovaní textových dát. Problém súčasných metód, ktorým je závislosť na konkrétnom jazyku textu, ako aj ich presnosť, ktorá nedosahuje úspešnosti človeka, rieši prostredníctvom troch smerov: tradičnými metódami a ich optimalizáciami, prístupom Big Data a abstrahovaním prostredníctvom minimalizácie jazykovo závislých častí, a prístupom hlbokého učenia. Hlavným cieľom dizertačnej práce bolo navrhnúť metódu pre strojové porozumenie neštruktúrovaným textovým dátam. Metóda bola experimentálne overená na probléme extrakcie jednoduchých informácií prostredníctvom klasifikácie textových dát v 5 jazykoch – čeština, angličtina, nemčina, španielčina a čínština, čím bola dokázaná možnosť aplikácie na rôzne rodiny jazykov. Pri validácií na databáze hodnotení Yelp bola dosiahnutá presnosť vyššia o 0,5% než poskytujú súčasné metódy.

KĽÚČOVÉ SLOVÁ

Analýza sentimentu, dolovanie znalostí, hlboké učenie, klasifikácia emócií, klasifikácia textu, optimalizácia genetickým programovaním, spracovanie prirodzeného jazyka, textové dáta, umelá inteligencia

ABSTRACT

This work deals with the problem of text mining which is becoming more popular due to exponential growth of the data in electronic form. The work explores contemporary methods and their improvement using optimization methods, as well as the problem of text data understanding in general. The work addresses the problem in three ways: using traditional methods and their optimizations, using Big Data in train phase and abstraction through the minimization of language-dependent parts, and introduction of the new method based on the deep learning which is closer to how human reads and understands text data. The main aim of the dissertation was to propose a method for machine understanding of unstructured text data. The method was experimentally verified by classification of text data on 5 different languages – Czech, English, German, Spanish and Chinese. This demonstrates possible application to different languages families. Validation on the Yelp evaluation database achieve accuracy higher by 0.5% than current methods.

KEYWORDS

Artificial intelligence, data mining, emotion classification, genetic programming optimization, natural language processing, sentiment analysis, text data, text mining

POVODA, Lukáš. *Dolovanie znalostí z textových dát použitím metód umelej inteligencie*. Brno, 2018, 100 s. Dizertačná práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Radim Burget, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju dizertačnú prácu na tému „Dolovanie znalostí z textových dát použitím metód umelej inteligencie“ vypracoval samostatne pod vedením vedúceho dizertačnej práce využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej dizertačnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto dizertačnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu dizertačnej práce pánovi doc. Ing. Radimovi Burgetovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno ....

podpis autora

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výskum popísaný v tejto dizertačnej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno ....

podpis autora

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Množstvo dát uložených v elektronickej podobe exponenciálne stúpa, pričom 80% týchto dát má textovú podobu a len 5% z týchto dát je možné považovať za hodnotné. [13] Predikcie publikované v štúdií [107] sľubujú, že v roku 2020 objem dát vytvorených človekom počas jedného roka dosiahne hodnotu 44 zettabajtov. Vďaka súčasným trendom v *Internete vecí – Internet of Things* (IoT), stále lacnejšiemu úložisku a dostupnejším rýchlejším pripojeniam je možné očakávať, že tento trend bude naďalej pokračovať. Tento trend má za následok zvýšený záujem rôznych firiem a vládnych organizácií o dolovanie hodnotných informácií z neštrukturovaných dát. Príkladom môže byť analýza sentimentu v textových správach, monitorovanie pacientov v reálnom čase, monitorovanie a riadenie skladov a iných výpočtových systémov, kontrola funkčnosti bežiacich úloh a predikcia budúcich problémov, ktoré by sa v systéme mohli vyskytnúť. Vďaka týmto informáciám môžu spoločnosti získať prehľad o trhu, spätnú väzbu na poskytované produkty či služby, študovať súčasný stav trhu, prípadne predikovať budúce správanie trhu.

Analýza textových dát, ako podmnožina neštrukturovaných dát, predstavuje problém, ktorý je ťažké algoritmizovať. Tieto dáta nemajú jasne definovaný dátový model, ktorý by mohol byť použitý pri hľadaní požadovanej informácie. Spracovať veľký objem textových dát je prakticky nemožné. Predstavovalo by to obrovské množstvo ľudí a vysoké finančné náklady. V súčasnosti existujú metódy, ktoré sa snažia tento problém riešiť – metódy dolovania znalostí (data-mining), resp. v tomto prípade dolovanie znalostí z textových dát (text-mining). Automatickou klasifikáciou textových dát môžeme vstupné dáta filtrovať a znížiť tak počet dát, ktoré je nutné spracovať, prípadne rovno požadovanú informáciu získať. Tieto metódy však v súčasnosti nedosahujú úspešnosti človeka.

Znalosť jazyka a gramatiky konkrétneho textu predstavuje základný predpoklad k pochopeniu textu. Ako uvádza štatistická štúdia Eurobarometer [26], v súčasnosti je na svete viac ako 7000 aktívne používaných jazykov. Takisto vďaka IoT vzniká množstvo strojových jazykov a formátov (gramatík), v ktorých sa uchovávajú záznamy z prevádzky serverov alebo IoT senzorov. Navrhnúť dostatočne univerzálne riešenie pomocou tradičných metód, ktoré by pokrylo diverzitu všetkých týchto gramatík a jazykov je obtiažne a časovo náročné. Predspracovanie vstupného textu sa v mnohých prípadoch spolieha na algoritmy pre určenie pôvodného (koreňového) tvaru slova, prípadne algoritmy pre opravu preklepov a gramatických chýb. Pre správnu funkčnosť potrebujú znalosť o jazyku, jeho štruktúre a tvarosloví. Teda do analytického systému vnášajú závislosť na danom jazyku vstupných textov. Zrušením jazykovo závislých častí predspracovania sa však rapídne znižuje úspešnosť takéhoto riešenia. Preto sa v poslednej dobe viacero výskumov usiluje o iný prístup

k tejto problematike – či už ide o využitie obrovských objemov dát (známych tiež ako Big Data[1]) alebo návrh metódy pre pochopenie skutočného významu textov (ako aj samotných slov a ich jednotlivých tvarov) prostredníctvom novších prístupov – prostredníctvom metód založených na hlbokých neurónových sieťach.

Práca sa zaoberá niekoľkými experimentami: A) problémom klasifikácie 5 emócií z českého a anglického textu, ktorej riešenie bolo postavené na tradičnom prístupe – relatívne malé množstvo dát, použitie jednoduchých klasifikátorov postavených na *metóde podporných vektorov – Support Vector Machines* (SVM), komplexné predspracovanie vstupného textu, rozšírenie o novo navrhnuté optimalizačné metódy, ako napr. sekvenčná eliminácia, ktorá je aplikovateľná na niekoľko paralelných vetiev klasifikačného algoritmu, ako aj ďalších optimalizačných metód publikovaných v článku [87] (IF pre rok 2016 = 0,945), ktoré zvýšili presnosť klasifikácie o 11,4%, B) abstrahovaním metódy pomocou Big Data prístupu, ktorý zvyšuje pamäťovú a časovú zložitosť trénovania a optimalizačných metód až do takej miery, že niektoré metódy optimalizácie nie je možné použiť (napr. metóda eliminácie vstupných parametrov), avšak bez jazykovo závislých častí systému dosahuje až o 11,0% vyššiu presnosť klasifikácie, než tradičný prístup s nízkym počtom trénovacích vzoriek (validované na 4 jazykoch – angličtina, nemčina, španielčina a čeština), C) problémom analýzy textových dát pomocou metód postavených na hlbokom učení s využitím komplexných štruktúr postavených na konvolučných alebo rekurentných sieťach, kde sa kládol dôraz na absolútnu jazykovú nezávislosť navrhnutej metódy, prekonanie úspešnosti metód súčasného stavu vedy o 0,5% (validované voči výsledkom výskumných tímov Google DeepMind [116] a Facebook Research [5] [18] na verejne dostupnej databáze Yelp reviews database zo súťaže z 1. 9. 2017), výsledné riešenie s novo navrhnutým jadrom (*Opakujúce sa jadro – Recurring Kernel* označené ako RCK) bolo v dobe písania práce v recenznom konaní k publikácií v časopise Cognitive Computation (IF pre rok 2016 = 3,441).

Hlavným prínosom tejto práce je návrh a experimentálne overenie univerzálnej metódy pre automatickú analýzu textových dát, ako aj metodiky pre analýzu dát bez predošlej znalosti jazyka či gramatiky. Výsledky obsiahnuté v tejto práci boli publikované na medzinárodných konferenciách a vedeckých časopisoch s impakt faktorom. Validácia prebiehala na privátnych databázach, ako aj na databázach dostupných pre vedecké účely, pre možnosť porovnania so súčasným stavom vedy a techniky.

Táto dizertačná práca je štruktúrovaná nasledovne: kap. 1 rozoberá súčasné prístupy v oblasti strojovej analýzy textu, strojového učenia a nového prístupu používaného v súčasnosti – hlbokého učenia, ktorého teória je vysvetlená na obrazových

[1] Termín zavedený v publikácií [88]

dátach, kap. 2 vytyčuje ciele dizertačnej práce, kde hlavným cieľom bol návrh metódy pre strojové porozumenie neštruktúrovaným textovým dátam bez predom pripravenej znalosti jazyka alebo gramatiky. Vlastné riešenie práce je popísane v kap. 3, kde v troch podkapitolách popisuje výskum a vývoj novej metódy. V kap. 4 sú uvedené výsledky overenia navrhnutých metód s porovnaním voči úspešnosti súčasných metód. Práca je uzavretá kap. 5.

1 Súčasný stav vedy a techniky

Témou dolovania znalostí z neštrukturovaných textových dát sa v súčastnosti zaoberajú výskumné inštitúcie ale aj veľké komerčné subjekty, akými sú napríklad *Google*, *Facebook*, *Amazon*, *Apple* alebo *Microsoft*. Pochopenie skutočného význam textu je často náročnou úlohou aj pre človeka. Význam textu môže autor meniť pomocou irónie alebo sarkazmu, texty môžu byť písane rôznymi štýlmi (zdvorilo, hovorovo, odborne apod.), rôzne jazyky môžu používať iné ustálené slovné spojenia. Tieto faktory naznačujú, že ide o jazykovo závislý problém – použitie rôznych slov, rôzne princípy vyjadrenia určitej myšlienky, rôzna syntax apod.

Súčasné riešenia problému dolovania znalostí z textových dát je možné z hľadiska vstupných dát rozdeliť do dvoch základných okruhov: 1) tradičné metódy, ktoré využívajú predspracované dáta na vstupe, majú k dispozícií vyextrahované funkčné vektory, ktoré vznikli na základe znalosti daného jazyka, jeho syntaxe a gramatiky, a 2) metódy založené na hlbokom učení, ktoré môžu pracovať so surovými dátami, teda bez predspracovania textu – bez znalosti daného jazyka, jeho syntaxe a gramatiky. Do druhej kategórie spadajú aj riešenia založené na modeloch zhlukovaní slov, ktoré sa využívajú na vstupe hlbokej neurónovej siete pre vytvorenie vektoru príznakov.

Pre objektívne vyhodnotenie úspešnosti rôznych algoritmov a prístupov k textovej analýze boli vytvorené databázy textov pre širokú vedeckú verejnosť. Práca sa zameriava na dve konkrétne databázy – databáza hodnotení Amazon [66] a databáza hodnotení Yelp, ktorých objem je vhodný pre všetky druhy metód. U databázy hodnotení Amazon ide o 82 miliónov záznamov, u databázy hodnotení Yelp ide o 4,7 milióna záznamov. Z týchto sú najčastejšie používané práve najvyššie hodnotenia (5 hviezdičiek) a najnižšie hodnotenia (1 hviezdička) u riešení problému analýzy sentimentu, resp. binárnej klasifikácie na pozitívny a negatívny text.

1.1 Tradičné metódy

Metódy označované ako „tradičné“ sú postavené na princípoch štatistickej analýzy. Tieto metódy využívajú komplexné predspracovanie textu, kde je znalosť jazyka a gramatiky zapracovaná vo forme stemmingu, lemmatizácie, odstraňovania stop slov apod. pre rozšírené jazyky ako je angličtina sú voľne k dispozícií knižnice, ktoré tieto úlohy plnia. Problém však nastáva u ostatných jazykov vo svete, pre ktoré nie sú dostupné riešenia komplexného predspracovania, prípadne u strojovo generovaných jazykoch, ktorých gramatika nemusí byť známa. Pri tradičných metódach je následne na predspracované dáta použitý jednoduchý model na štatistické vyhodnotenie vstupného vektoru príznakov a vyvodenie záveru – pri klasifikácii určenie triedy, do ktorej spadá vstupný text. Tradičné metódy nachádzajú v súčasnosti

uplatnenie najmä v prípadoch, kde nie je možné vytvoriť veľké objemy trénovacích dát, resp. manuálne vytvorenie databázy textov by bolo časovo a finančne náročné. Tradičné metódy je možné rozdeliť do 3 skupín:

1. metódy založené na detekcií kľúčových slov,
2. metódy postavené na strojovom učení,
3. hybridné metódy.

Metódy postavené na detekcií kľúčových slov sú relatívne jednoduché, výpočtovo nenáročné, avšak spoliehajú sa na to, že na vstupe sa objaví kľúčové slovo v presne danom tvare. Správne predspracovanie u takéhoto systému je kritické a každá drobná chyba môže znamenať výraznú odchýlku na celkovej úspešnosti. Tento princíp bol využitý v článku [112], kde s celkovou úspešnosťou 81,0% klasifikovali texty do 4 tried podľa emócie obsiahnutej v texte. Algoritmus detekcie založený na kľúčových slovách bol schopný vďaka definovaným pravidlám pochopiť sémantický význam danej vety a určiť tak skutočnú emóciu autora textu. Takýto prístup je však značne obmedzený na manuálne definovanú databázu kľúčových slov daného jazyka a danej emócie. Preto sa od tohto prístupu momentálne upúšťa, resp. využitie nájde už len pri jednoduchších problémoch, ako je napr. následná klasifikácia podľa kľúčových slov v rozpoznávanom rečovom signále, viac v článkoch [16] a [61].

Druhou veľkou skupinou sú metódy postavené na strojovom učení, označované tiež ako metódy umelej inteligencie. Narozdiel od predošlej skupiny, metódy strojového učenia dosahujú omnoho lepšie výsledky a svojim princípom sa približujú k strojovému pochopeniu textových dát, čo dokazuje aj článok [97]. Porovnanie úspešnosti klasifikácie anglických textov do 5 tried vykazuje až o 32,1% vyššiu úspešnosť pri Bayesových sieťach. Klasifikácia prebiehala na 5407 krátkych správach zo sociálnej siete. Uplatnenie v textovej analýze pomocou umelej inteligencie našiel najmä klasifikátor *Metóda podporných vektorov – Support Vector Machines* (SVM) s ktorým mnohé výskumné tímy dosahujú najvyššie presnosti klasifikácie spomedzi tradičných metód. V článku [45] bola publikovaná úspešnosť 85,5% pri klasifikácií do 6 tried, pričom pre trénovanie bola použitá malá databáza – 250 krátkych textov (titulky z novín). Kratšie texty a klasifikátor SVM boli využité aj v článku [9], resp. v rozšírenom článku [10], kde boli texty klasifikované podľa emócií s dosiahnutou úspešnosťou 80,3% pre 6 tried. Pre trénovanie bolo zvolených 1000 náhodných titulkov z novín. Tradičnými metódami postavenými na strojovom učení sa zaoberá aj komerčný sektor, kde v článku [116] bola dosiahnutá úspešnosť 86,0% pre binárnu klasifikáciu – článok je publikovaný tímom Google's DeepMind.

Poslednou skupinou tradičných metód sú hybridné metódy, ktoré sa snažia vyťažiť to najlepšie z predošlých dvoch spomínaných skupín. Často využívajú pomocné slovníky, či množiny reprezentujúce požadovanú informáciu (časť kľúčových slov) a

klasifikátory využívajúce predspracované textové dáta ako vstupné parametre umelej inteligencie. V článku [94] autori popisujú metódu pre detekciu emócií v textoch, založenú na automatickom generovaní pravidiel podľa už existujúcich metód ako je WordNet [71] a ConceptNet [59]. Trénovaný bol klasifikátor *k najbližších susedov – k Nearest Neighbors* (k-NN) relatívne malou množinou s veľkosťou 173 anglických správ (dokopy 5205 viet), pričom každá zo 6-tich tried bola zastúpená rovnomerne. Dosiahnutá úspešnosť klasifikácie 86% však bola nameraná na relatívne malej množine textov (570 viet). Autori článku [109] dokonca zapracovali do systému analýzy emócií z čínskych textov prekladač do angličtiny, slovníky pozitívnych a negatívnych slov pre oba jazyky, ako aj slovníky pre rozpoznanie negácie. Vyhodnocovaním dvoch rôznych výstupov klasifikátorov sa dostali na úspešnosť 86,1% pre rozpoznávanie pozitívnych a negatívnych textov.

1.1.1 Reprezentácia dát na vstupe

U tradičných metód sa na vstupe využíva transformácia vstupného textu do vektoru príznakov, ktoré reprezentujú zastúpenie jednotlivých tokenov (slov) v danom texte. Veľkosť takéhoto vektoru je definovaná celkovým počtom unikátnych slov v danom jazyku, respektíve slov, ktoré sa nachádzali v trénovacej množine.

Reprezentácia pomocou Bag-of-Words

Základným predstaviteľom techniky určovania zastúpenia jednotlivých slov je *batôžtek slov – Bag-of-Words* (BoW), známy tiež ako vektorový model slov. Podľa tohto modelu je text (veta, dokument) reprezentovaný ako množina slov prehliadajúc gramatiku (morfológia, v ideálnom prípade aj preklepy, synonymá a stop slová) a poradie slov, ponecháva sa však informácia o počte slov. BoW vytvára teda histogram slov z daného textu, pričom počet daného slova predstavuje jeden príznak. [30]

Ak teda klasifikačnú úlohu definujeme ako $y_t = f_t(T)$, kde T sú vstupné textové dáta a y_t je predikcia získaná prostredníctvom funkcie f_t, T môže byť reprezentované prostredníctvom BoW ako $T = \{w_1, w_2, \ldots, w_n\}$, kde w_i predstavuje počet výskytov daného slova. Funkcia f_t môže byť získaná prostredníctvom strojového učenia klasifikátoru ako je napr. SVM. [110]

Tab. 1.1 zobrazuje príklad zakódovania textových dát pomocou BoW, kde boli k dispozícií vety:

1. Boli to najlepšie časy.
2. Boli to najhoršie časy.
3. Bol to čas múdrosti.
4. Bol to čas pochabosti.

Tab. 1.1: Príklad zakódovania vstupného textu pomocou BoW

Veta	**Slovo**						
	byť	to	čas	dobrý	zlý	múdrosť	pochabosť
1.	1	1	1	1	0	0	0
2.	1	1	1	0	1	0	0
3.	1	1	1	0	0	1	0
4.	1	1	1	0	0	0	1

Z dostupných viet bol vytvorený slovník obsahujúci 7 slov („byť“, „to“, „čas“, „dobrý“, „zlý“, „múdrosť“ a „pochabosť“) z pôvodného korpusu pozostávajúceho z 16 slov. Pre vytvorenie BoW reprezentácie boli slová z každého dokumentu ohodnotené. Cieľom je transformovať voľný text do vektoru, ktorý je možné použiť na vstupe niektorého algoritmu strojového učenia. Dĺžka tohto vektoru vychádza z veľkosti vytvoreného slovníka, kde každá pozícia reprezentuje jedno slovo z extrahovaného slovníka. Najjednoduchšia varianta hodnotenia slov je určenie výskytu daného slova v transformovanom texte. [30]

So zväčšujúcim počtom slov v korpuse sa úmerne zväčšuje aj veľkosť slovníka, a teda aj vytvoreného vektoru. Preto je dobré aplikovať niektoré kroky predspracovania textu, ako sú: zmena písmen na písmena malej abecedy, ignorovanie často sa opakujúcich slov, ktoré nenesú žiadnu informáciu, oprava preklepov, oprava diakritiky, transformovanie slov do ich základnej podoby (ako bolo uvedené na príklade), prípadne sofistikovanejšie riešenia akým je napr. vyhľadávanie synoným.

Nevýhody tejto transformácie spočívajú: **v slovníku**, u ktorého je potreba dbať na správny návrh, správne zvoliť veľkosť slovníka vzhľadom na hustotu reprezentácie, **riedka reprezentácia** nie je vhodná pre hľadanie požadovanej funkcie strojovým učením vzhľadom na časovú a pamäťovú náročnosť, metóda **odstraňuje informácie o poradí slov**, čím dochádza k vypusteniu pôvodného kontextu, v ktorom bolo slovo použité. Kontext častokrát určuje skutočný význam slova.

Reprezentácia pomocou TF–IDF

Metóda *Term Frequency – Inverse Document Frequency* (TF–IDF) je štatistická metóda reflektujúca dôležitosť daného slova v dokumente. Ide o hodnotiacu funkciu používanú v BoW, ktorá rieši problém hodnotenia frekvencie slov, kde najfrekventovanejšie slová dominujú v reprezentácií dokumentu (majú vysoký počet výskytov), ale nemusia obsahovať hľadanú informáciu. TF–IDF sa zvyšuje priamoúmerne s počtom výskytov daného slova v dokumente (zložka TF – *Term Frequency*) a znižuje

sa s frekvenciou výskytu daného slova v celom korpuse (zložka IDF – *Inverse Document Frequency*). Vďaka tomu je možné detekovať slová, ktoré sa vyskytujú v textoch všeobecne častejšie. [15, 91]

Tab. 1.2 zobrazuje predošlý príklad textových dát zakódovaných prostredníctvom hodnotiacej funkcie TF–IDF, kde zložka TF je definovaná počtom výskytu (rovnako ako u BoW) a zložka IDF je určená vzorcom

$$idf_i = \log\left(\frac{|D|}{|\{d \in D : w_i \in d\}|}\right),$$

kde čitateľ predstavuje celkový počet dokumentov a menovateľ predstavuje počet dokumentov, ktoré obsahujú slovo w_i.

Tab. 1.2: Príklad zakódovania vstupného textu pomocou TF–IDF

Veta	**Slovo**						
	byť	to	čas	dobrý	zlý	múdrosť	pochabosť
1.	0	0	0	0,602	0	0	0
2.	0	0	0	0	0,602	0	0
3.	0	0	0	0	0	0,602	0
4.	0	0	0	0	0	0	0,602

Keďže každé slovo v texte má inú mieru dôležitosti, použitím metódy TF–IDF je možné určiť ich váhu. Z číselnej reprezentácie dôležitosti jednotlivých slov na príklade v Tab. 1.2 je vidieť, že slová, ktoré sa vyskytujú vo všetkých dokumentoch majú nulovú váhu. Ide o žiadaný efekt, ktorý má za následok zvýraznenie slov obsahujúcich užitočnú informáciu.

Rozšírenie o n-gramy

Keďže obe spomínané metódy (BoW a TF–IDF) spôsobujú stratu informácie o poradí slov, vytvorenie slovníka obsahujúceho dvojice slov môže pomôcť zachytiť význam zo širšieho kontextu. Ide napríklad o negáciu, kde práve dvojica (resp. n-tica) slov môže zachytiť tento jav. Zvolením vhodného n je možné zachytiť viaceré prvky analyzovaného jazyka, čím docieliť vyššiu presnosť následne natrénovaného modelu.

Kombináciou n-gramov a metódy TF–IDF bola v článku [118] dosiahnutá úspešnosť 91,54% pre binárnu klasifikáciu na databáze hodnotení Amazon a 95,44% na databáze hodnotení Yelp, pri použití $n = 5$. Takéto riešenie však predstavuje obrovské pamäťové nároky, keďže vstupný vektor môže obsahovať desiatky až stovky miliónov príznakov v závislosti od daného jazyka.

1.1.2 Metódy strojového učenia

Po reprezentácií vstupných dát pomocou vektoru, môže byť riešená úloha klasifikácie týchto dát. Pod klasifikáciou sa rozumie technika dolovania znalostí s dohľadom, ktorá zahŕňa priradenie triedy množine neoznačených objektov. Na základe počtu tried prítomných v riešenom probléme sa klasifikácia delí na: binárnu, kde klasifikácia prebieha do dvoch tried, a na klasifikáciu viacerých tried. Ako popisuje práca [97], časté problémy riešené pomocou klasifikácie textov sú:

- klasifikácia noviniek na triedy: politika, šport, svet, financie a životný štýl,
- rozpoznávanie nevyžiadanej elektronickej pošty,
- analýza sentimentu – pozitívna a negatívna správa, prípadne neutrálna.

Narozdiel od metód rozpoznávania kľúčových slov, metódy postavené na strojovom učení dosahujú omnoho lepšie výsledky a svojím princípom sa o niečo viac približujú k strojovému pochopeniu textu. Úloha automatickej textovej analýzy je v posledných rokoch rozšírená a študovaná. Vďaka popularite bol dosiahnutý značný pokrok v tejto oblasti s použitím algoritmov ako sú: Bayesovské siete, rozhodovacie stromy, k-NN, SVM apod. [49] Ako zobrazuje článok [87], najlepšie výsledky z testovaných dosahuje klasifikátor SVM.

Bayesovské siete

Bayesovské siete predstavujú významnú časť strojového učenia, ako reprezentant pravdepodobnostných modelov. Bayesovské siete využívajú grafovú reprezentáciu pre zobrazenie pravdepodobnostných vzťahov medzi jednotlivými javmi. Ide vlastne o orientované acyklické grafy, kde vrcholy odpovedajú náhodnej veličine a hrany vyjadrujú vzťahy podmienenej závislosti medzi premennými. [73, 77]

Bayesov teorém [74] vraví, že pravdepodobnosť, že náhodná veličina R má hodnotu r pri dôkaze e je možné vyjadriť ako

$$P(R = r|e) = \frac{P(e|R = r)P(R = r)}{P(e)},$$

kde čitateľ predstavuje súčin podmienenej pravdepodobnosti pri dôkaze e a pravdepodobnosti výskytu daného javu, menovateľ predstavuje pravdepodobnosť dôkazu e – teda predstavuje normalizačnú konštantu, ktorá zabezpečí, že pravdepodobnosť bude menšia alebo rovná 1. Pri komplikovaných pravdepodobnostných modeloch je výpočet normalizačnej konštanty $P(e)$ výpočtovo náročný, až nerealizovateľný z dôvodu vysoko-rozmerného vektoru. Grafová štruktúra pomáha riešiť tento problém prostredníctvom rozdelenia tejto pravdepodobnosti.

Príkladom na aplikovanie Bayesovho teorému môže byť požiadavka na určenie pravdepodobnosti skutočného výskytu určitého ochorenia, kedy pri testovaní dopadol test pozitívne. Tá je závislá od presnosti a senzitivity testu a od pravdepodobnosti ochorenia. Jav pozitívneho výsledku testu označíme ako T a jav ochorenia ako O, ak $P(T = 1|O = 1) = 0,95$ (teda testy vykazujú 5% chybu), a $P(T = 1|O = 0) = 0,05$, pri predpoklade, že sa jedná o vzácne ochorenie $P(D = 1) = 0,01$, potom

$$P(O = 1|T = 1) = \frac{P(T = 1|O = 1) \cdot P(D = 1)}{P(T = 1|D = 1) \cdot P(D = 1) + P(T = 1|D = 0) \cdot P(D = 0)}$$

$$= \frac{0,95 \cdot 0,01}{0,95 \cdot 0,01 + 0,05 \cdot 0,99} = 0,161 \ ,$$

takže pravdepodobnosť výskytu ochorenia predstavuje len 16%. [74]

Rozhodovacie stromy

Ďalšou z techník využívaných pri dolovaní znalostí z neštruktúrovaných dát je metóda rozhodovacieho stromu. Hlavnou výhodou tejto metódy je, že vzniká štruktúra, ktorá je jednoducho interpretovateľná a pochopiteľná človekom. Zároveň umožňuje vyvodzovať hneď niekoľko záverov. Základnou myšlienkou je roztrieštiť problém na niekoľko jednoduchých rozhodnutí, ktoré budú viesť k požadovanému výsledku. Uzly stromu teda predstavujú miesta, kde dochádza k rozhodnutiu na základe určitých príznakov, listy stromu sú výsledné závery. [90]

Hlavnou úlohou rozhodovacích stromov je správne klasifikovať dáta podla trénovacej množiny, pričom u návrhu tohto stromu je snaha generalizovať problém tak, aby aj vzorky, ktoré sa nenachádzajú v trénovacej množine mohli byť správne klasifikované. Snahou je tiež udržať štruktúru takéhoto stromu čo najjednoduchšiu. [50, 51]

Návrh vhodnej štruktúry rozhodovacieho stromu je možný chápať ako optimalizačnú úlohu, kde ide o snahu minimalizovať pravdepodobnosť chyby P_e. Táto je závislá od voľby štruktúry stromu T, zvolenej podmnožiny príznakov F a navrhnutých rozhodovacích pravidiel d. Pri limitovanej veľkosti trénovacej množiny je potreba poznamenať, že úspešnosť klasifikácie sa bude zhoršovať so zvyšujúcim sa počtom príznakov – tvrdenie známe aj ako Hughesov jav [40]. Definíciou množiny povolených príznakov N zo všetkých dostupných L, kde teda $N << L$, je možné tomuto javu zabrániť. Tento optimalizačný problém je teda možné vyriešiť, ako popisuje článok [51], pomocou dvoch krokov:

1. pre dané T a F nájdi také d^*, kde

$$p_e\left(T, F, d^*(T, F)\right) = \min_d P_e\left(T, F, d\right)$$

2. nájdi T^* a F^*, kde

$$P_e\left(T^*, F^*, d^*(T^*, F^*)\right) = \min_{T,F} p_e\left(T, F, d^*(T, F)\right)$$

Náhodné lesy

Náhodný les je kombinovaná metóda strojového učenia rozširujúca teóriu rozhodovacích stromov. Metóda využíva trénovanie niekoľkých rozhodovacích stromov, ktorých jednotlivé výsledky prostredníctvom štatistického modusu rozhodujú o výsledku klasifikácie. Náhodné lesy vznikli ako riešenie „pretrénovania“, ku ktorému často dochádza pri použití rozhodovacích stromov.

Problém „pretrénovania“ vychádza zo samotnej podstaty strojového učenia. Učiaci sa algoritmus je trénovaný na množine trénovacích dát, potom je však aplikovaný na novú množinu dát. Úlohou je maximalizoval úspešnosť predikcie na nových dátach. Nie je teda potrebné maximalizovať úspešnosť na trénovacej množine, keďže by pri hľadaní toho najlepšieho modelu došlo k učeniu na šume v týchto dátach, resp. k zapamätaniu zvláštností a drobností z trénovacej množiny, a nie hľadaniu všeobecných pravidiel na správnu predikciu výsledku. [21]

Definícia náhodného lesu sa skladá z množiny rozhodovacích stromov $T_1, \ldots, T_N$. Trénovacie množiny pre jednotlivé stromy T_i pozostáva z výberu z kompletnej množiny trénovacích vzoriek. Ide o výbery s náhodným opakovaním vzoriek, pričom je definovaná ich veľkosť n. Výsledok klasifikácie náhodného lesa je daný

$$C_r f = \text{modus}\left\{C_i(x)\right\}_1^N \ ,$$

kde $C_i(x)$ je výsledok klasifikácie i-teho stromu z celkového počtu N stromov. Jednotlivé stromy teda pracujú v podpriestoroch definovaných náhodne vybranou podmnožinou trénovacích dát. [36, 37]

k-NN klasifikátor

Algoritmus k-najbližších susedov (označovaný ako k-NN) predpokladá, že každá vzorka (inštancia) predstavuje bod v n-rozmernom priestore. Trénovacie vzorky predstavujú vektory v tomto priestore, kde každá ma priradenú triedu. Proces trénovania predstavuje uloženie týchto vektorov a im prislúchajúcich tried. V procese klasifikácie najbližší susedia klasifikovanej inštancie, ktorí sú určení prostredníctvom Euklidovskej vzdialenosti, hlasujú o zaradení inštancie do príslušnej triedy.

Ak vstupnú vzorku (inštanciu) definujeme ako n-ticu x_i a trénovaciu množinu ako dvojicu (x_i, ω_i), kde ω_i je trieda, do ktorej spadá daná vzorka, klasifikátor k-NN zaradí vstupnú vzorku x do rovnakej triedy ako jej najbližšia vzorka z trénovacej množiny podla funkcie $f(x) = \omega_i$, kde $i = \min_{j \in \{1,\ldots,m\}}(x - x_j)$. Ak je $k > 1$, dochádza ku klasifikácií pomocou viacerých susedných vzoriek.

Algoritmus sa dá jednoducho upraviť pre výpočet spojitej cieľovej funkcie, teda na tzv. riešenie regresného problému. Princíp spočíva v tom, že sa určí priemerná hodnota k-najbližších susedov z trénovacej množiny miesto najčastejšie sa vyskytujúcej hodnoty. Metóda k-NN disponuje jednoduchou implementáciou a relatívne vysokou účinnosťou. Je vhodná predovšetkým ku klasifikácií do nižšieho počtu tried. V prípade ďalších požiadaviek existujú rôzne modifikácie tejto metódy. [2, 24, 72]

SVM klasifikátor

Metóda SVM, označovaná aj ako metóda podporných vektorov, je určená na klasifikáciu a regresnú analýzu. Pri trénovaní klasifikátoru vzniká model, ktorý delí dáta do dvoch skupín prostredníctvom deliacej nadroviny v n-dimenzionálnom priestore. Výsledná nadrovina poskytuje vhodné rozdelenie vtedy, ak vzdialenosť k najbližšiemu bodu (vzorku z trénovacej množiny) je čo najväčšia, keďže vo všeobecnosti väčšia vzdialenosť okraja znamená nižšiu chybu klasifikácie. [100, 105]

Základná varianta SVM klasifikátoru sa označuje ako lineárny SVM. Vstupné dáta reprezentované pomocou vektoru $x_1, \ldots, x_n$ sú v tomto prípade reprezentované v pôvodnom n-dimenzionálnom priestore χ, kde $\chi \in \mathbb{R}^d$. Najväčším prínosom SVM klasifikátoru je však možnosť transformovať n-rozmerný χ priestor na priestor definovaný nelineárnou funkciou prostredníctvom tzv. Mercerovho operátoru K (resp. kernel funkcia), čím sa dosiahne lineárna separácia vstupných dát pôvodne lineárne neseparovateľných. Takéto správanie možno chápať ako množinu klasifikátorov a ich výstupnú funkciu

$$f(x) = \left(\sum_{i=1}^{n} \alpha_i K(x_i, x) \right) ,$$

kde K spĺňa Mercerovu podmineku [8] a $K(u, v) = \Phi(u) \cdot \Phi(v)$, kde Φ funkciu prevodu medzi pôvodným priestorom a novým priestorom. Dve najrozšírenejšie K funkcie sú: polynomický kernel $K(u, v) = (u \cdot v \ + 1)^p$, kde p predstavuje úroveň polynómu, a radiálny kernel $K(u, v) = (e^{-\gamma(u-v)\cdot(u-v)})$. [105]

Majorita analýzy sentimentu využívajúca tradičné metódy sa sústreďuje prevažne na klasifikátory postavené na metóde SVM, keďže ide o najúspešnejšiu metódu v tejto oblasti. Články [20, 45, 79, 106] zobrazujú omnoho vyššiu úspešnosť klasifikácie pomocou SVM metódy než u Bayesovských sieťach, rozhodovacích stromoch, náhodných lesoch či ďalších metódach strojového učenia. Aj z tohto dôvodu bola metóda SVM zvolená pri riešení tejto práce.

1.1.3 Hodnotenie výkonnosti

Všeobecne akceptované metódy hodnotenia úspešnosti metód strojového učenia predpokladajú nezávislosť od riešeného problému. Úspešnosť klasifikácie musí byť určená

bez závislosti od danej triedy resp. od počtu tried. Z tohto dôvodu je aj v súčastnosti najviac využívaná metrika „presnosti“ vychádzajúca z empírie nerozlišuje medzi počtom správnych predikcií rôznych tried. Metrika „presnosť“[1], označovaná aj termínom *accuracy*, je definovaná ako

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn} ,$$

v prípade, že ide o klasifikáciu tried „pozitívny“ a „negatívny“, potom tp predstavuje počet skutočne pozitívnych vzoriek, tn predstavuje počet skutočne negatívnych vzoriek, fp predstavuje počet predikovaných pozitívnych vzoriek a fn predstavuje počet predikovaných negatívnych vzoriek. Presnosť približuje pravdepodobnosť správneho výsledku klasifikácie. [96]

Metriky zamerané na danú triedu sú často využívané v oblasti klasifikácie textu, extrakcií informácií, spracovania prirodzeného jazyka apod. V týchto oblastiach môže byť počet prvkov jednej triedy výrazne nižší než u inej triedy. V týchto prípadoch sa využívajú metriky „precíznosť“ (anglicky *precision*) a metrika označená ako „senzitivita“ (anglicky *recall* alebo *sensitivity*), ktoré sú pre pozitívnu triedu definované ako

$$precision_p = \frac{tp}{tp + fp} , \quad recall_p = \frac{tp}{tp + fn} ,$$

a obdobne pre negatívnu triedu definované ako

$$precision_n = \frac{tn}{tn + fn} , \quad recall_n = \frac{tn}{tn + fp} ,$$

kde precíznosť odhaduje prediktívnu silu algoritmu a senzitivita odhaduje efektivitu algoritmu na danej triede. Z týchto metrík je možné následne určiť metriku „F–score“, ktoré je v prípade $\beta = 1$ počíta s vyváženými množinami, v prípade $\beta > 1$ uprednostňuje precíznosť, v opačnom prípade senzitivitu. [96]

$$F\text{–}score = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}$$

1.1.4 Optimalizačné metódy

Extrakcia informácií z viacdimenzionálnych dát predstavuje štatistický problém, ktorého riešenie má uplatnenie v mnohých aplikáciach. Dáta rozprestreté

[1] Ďalej v práci používaná už len ako „presnosť“ alebo „presnosť klasifikácie“.

v n-rozmernom priestore zahŕňajú množstvo príznakov, pričom nie každý príznak je skutočne dôležitý pre riešenie daného problému. Odstránenie irelevantných príznakov je pri návrhu predikčného modelu nevyhnutným krokom, keďže vysoký počet vstupných parametrov môže viesť k pretrénovaniu. Odstránením niektorých príznakov dôjde k odstráneniu šumu zo vstupných dát, ktorý môže viesť k zmäteniu predikčného modelu a zníženiu presnosti predikcie. Skúsiť však všetky možnosti kombinácií povolených príznakov nie je z časového hľadiska možné. [60, 102, 115]

Práca sa zameriava na najjednoduchšie metódy optimalizácie strojového učenia filtrovaním vstupných parametrov – doprednou selekciou a spätnou elimináciou príznakov, ktoré sú súčasne najrozšírenejšie metódy selekcie podmnožiny príznakov, ako popisuje [63]. Ďalšie metódy selekcie vychádzajú z logiky týchto metód, prípadne neboli považované za vhodné pre použitie na probléme dolovania znalostí z textových dát.

Optimalizačné metódy zamerané na textové dáta popisuje článok [115]. Ide o metódy, ktoré majú znalosť o danom texte (resp. korpuse textov) a táto sa využíva v ich behu. Ide napríklad o metódu prahovania prostredníctvom *minimálnej frekvencie v dokumentoch* (v práci ďalej označovaná ako MDF).

Dopredná selekcia príznakov

Najjednoduchším prístupom tvorenia modelu s minimom vstupných parametrov je metóda doprednej selekcie príznakov. Metóda pridáva parametre postupne, v každom kroku je pridaný len jeden. Každý parameter je teda postupne testovaný, či jeho začlenenie do množiny vstupných parametrov bude mať kladný vplyv na výslednú presnosť natrénovaného modelu, a či sa teda jedná o významný príznak. Parameter, ktorý spôsobil najvýraznejšiu kladnú zmenu presnosti modelu (napr. klasifikácie), bude použitý. Tento proces sa opakuje do doby, kedy žiaden zo zostávajúcich parametrov nepredstavuje významný príznak. [60]

Ide o výpočtovo náročnú operáciu, keďže stavový priestor možných kombinácií množín je prehľadávaný postupne. Metóda doprednej selekcie je však výpočtovo efektívnejšia než spätná eliminácia. Menšou podmnožinou vstupných parametrov však môže dôjsť k odstráneniu potenciálne potrebných parametrov v kontexte ostatných parametrov, ktoré doposiaľ neboli do podmnožiny zahrnuté. [34]

Spätná eliminácia príznakov

Narozdiel od doprednej selekcie, spätná eliminácia začína s plnou množinou vstupných parametrov. Následne iteratívne odstraňuje najmenej užitočné príznaky – vždy jeden v každej iterácií. Výsledkom tejto metódy je vo všeobecnosti model s vyššou dimenzionalitou.

1.2 Metódy hlbokého učenia

V posledných rokoch sa vďaka rýchlo sa rozvíjajúcej výpočtovej technike objavuje aj ďalšia skupina – skupina hlbokých neurónových sietí. Tieto siete počítajú s existenciou väčších objemov dát a zároveň s vyšším výpočtovým výkonom. Preto je nutné optimalizovať beh takýchto algoritmov prostredníctvom masívnej paralelizácie, ako popisujú články [64], [65] a [58], kde prvé pokusy s behom na grafických akcelerátoroch boli už v 90-tych rokoch [41]. Skutočné využitie masívneho paralelizmu a hlbokého učenia bolo popísané prvý krát v roku 2006 v článku [14], konkrétne pre *konvolučné neurónové siete – Convolutional Neural Network* (CNN). V článku [104] boli využité *rekurentné neurónové siete – Recurrent Neural Network* (RNN) siete pre analýzu emócií v textových dátach, konkrétne implementované pomocou *Long Short Term Memory* (LSTM) – predstavené v článkoch [38] a [28]. Tu bola dosiahnutá úspešnosť 67,6% pre klasifikáciu 5 tried práce pomocou LSTM. V porovnaní s konvolučnými sieťami, kde bola dosiahnutá presnosť 66,0%. Na druhej množine dát bola dosiahnutá úspešnosť klasifikácie 10 tried 45,3%. Extrakciou zložitejších informácií z prirodzeného jazyka sa zaoberá článok [68]. V článku pomocou RNN siete bola snaha o extrakciu textových sekvencií s úspešnosťou 93,81% (F-score = 93,98) pomocou metódy vyplňovania slotov (Slot Filling). V ďalšej práci rozšírením o *Conditional Random Field* (CRF) v článku [67] bolo dosiahnuté zlepšenie presnosti o 0,17%.

Súčasné metódy dosahujúce najlepšie výsledky (označované aj ako „state-of-the-art“ metódy) využívajú práve princípy hlbokého učenia a sú zhrnuté v článkoch [116] a [33]. Tab. 1.3 porovnáva tieto metódy na databáze hodnotení Yelp, ktorá bola použitá vo všetkých publikáciách, konkrétne pre problém klasifikácie pozitívneho a negatívneho textu.

Osobitým prístupom k danému problému sú metódy postavené na hlbokom učení a modeloch zhlukovania (v angličtine označované ako „embedding models“). Modely zhlukovania slov sú prvou skupinou, ktoré v sebe nesú informácie o jazyku a jeho vlastnostiach prostredníctvom zakódovania jednotlivých slov do vektoru, ktorý predstavuje vzťahy medzi jednotlivými slovami. Použitím tohto vektoru sa značne znižuje dimenzionalita vstupu a teda aj komplexnosť riešeného problému. Ako popisuje článok [53], použitím tejto techniky a jednoduchej logistickej regresie môže byť dosiahnutá úspešnosť až 80,4% na relatívne malej množine textov (použitých iba 2000 vzoriek). Experiment popísaný v článku [111] využíva model zhlukovania pre dosiahnutie úspešnosti 85,5% na databáze Google Snippets [81] a 96,8% úspešnosť na databáze TREC (10 tried, texty sú špecificky zamerané). Článok [27] dokazuje, že modely zhlukovania sú schopné výrazne prekonať tradičné metódy.

Modely zhlukovania viet sú navrhnuté na zakódovanie celej vety do vektoru prí-

znakov. Zhlukovanie viet má nevýhodu v tom, že nie je schopné zakódovať poradie slov. Tento nedostatok môže predstavovať značný problém hlavne pri generovaných strojových textov (logoch) a nie veľmi rozšírených jazykoch. Rovnaký problém sa vyskytuje u už spomínanej metódy TF–IDF, ktorú je možné pokladať za reprezentanta tradičných prístupov predspracovania textu. Pri zhlukovaní viet je teda hlavnou požiadavkou na sémanticky zviazané susedné vety v texte, ako popisujú články [55] a [46]. Modely zhlukovania postavené na RNN sieťach navyše požadujú, aby bola označená každá veta, čo môže predstavovať ďalšie problémy pri tvorení databázy textov. [78]

Oba prístupy – tradičný i prístup hlbokého učenia využívajúci modely zhlukovania – požadujú pre svoju správnu funkčnosť vopred pripravenú znalosť daného jazyka. U tradičných metód je to proces predspracovania. Tieto metódy však nedosahujú úspešnosť „state-of-the-art“ metód postavených na prístupe hlbokého učenia, ako zobrazujú výsledky tímu Google's DeepMind v článku [116] a výsledky tímu Facebook AI Research v článku [33]. Vytvorenie modelu zhlukovania pre ďalší jazyk predstavuje úlohu náročnú na výpočtové prostriedky a na dostatok kvalitných textových dát – na natrénovanie vyžadujú veľký korpus miliónov záznamov. [5, 69, 80] Problémom môže byť tiež veľkosť takéhoto modelu, ktorá napr. u Word2Vec dosahuje 1,5GB.

1.2.1 Základy hlbokého učenia

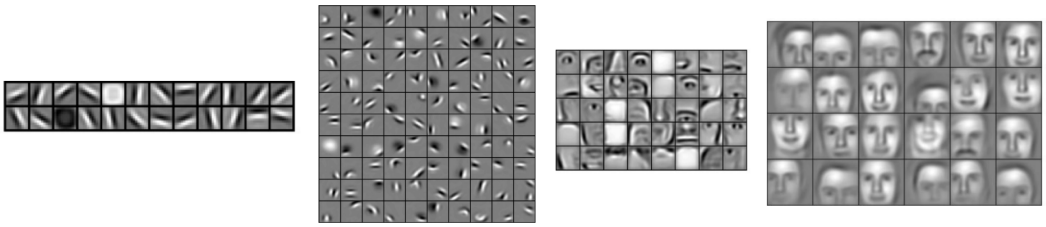
Štandardné plytké neurónové siete pozostávajú z veľkého počtu spojených jednotiek (neurónov) produkujúcich sekvenciu reálnych čísel pomocou ich aktivačnej funkcie. Vstupné neuróny sú aktivované vstupnými dátami, ďalšie neuróny sa aktivujú vá-

Tab. 1.3: Porovnanie súčasných metód binárnej klasifikácie databázy Yelp

Model	**Rok, publikácia**	**Úspešnosť**
Bag of Words	2015, [118]	92.2%
n-grams	2015, [118]	95.6%
Char-CNN	2015, [118]	94.7%
Char-CRNN	2016, [114]	94.5%
Very deep CNN	2016, [18]	95.7%
FastText	2016, [5]	93.8%
FastText with bigrams	2016, [5]	95.7%
Discriminative LSTM	2017, [116]	92.6%
Generative LSTM	2017, [116]	90.0%

hovaným spojením predošlých aktívnych neurónov. Učenie takejto siete predstavuje hľadanie takých váh, ktoré docielia požadované správanie siete. Vzhľadom na riešený problém a štruktúru zapojenia neurónov je učenie neurónovej siete výpočtovo náročný problém pozostávajúci z mnohých krokov, kde v každom kroku dochádza k transformáciám (často nelineárnym) agregovaných aktivácií. [92]

Hlboké učenie z veľkej časti pozostáva z učenia príznakov, resp. snaží sa riešiť problém extrakcie dôležitých príznakov pre riešený problém. Tento krok je možné chápať aj ako automatizáciu návrhu predspracovania dát. U obrazových dát sa v hlbokom učení za týmto účelom používajú konvolučné vrstvy, ktoré sú schopné nájsť príznaky v obraze pre ďalšie vrstvy a formovať tak hierarchiu nelineárnych príznakov, ktoré narastanú na komplexite (zhluky obrazových bodov, následne hrany, následne oči, nos, líca, z čoho je v ďalších vrstvách detekovaná tvár), ako zobrazuje Obr. 1.1 prevzatý z [57]. Posledné vrstvy hlbokých neurónových sietí sa zvyčajne zameriavajú na klasifikačný alebo regresný problém. [92, 113]



Obr. 1.1: Vizualizácia príznakov konvolučných vrstiev v rôznych hĺbkach siete [57]

V hierarchií hlbokej neurónovej sieti (DNN z anglického *Deep Neural Network*) dochádza k extrakcií príznakov v rôznych hĺbkach siete. Tieto sú následne spracované klasifikátorov, ktorý kombinuje všetky tieto príznaky a vyvodzuje záver – predikciu. Je teda potrebné tvoriť zložité štruktúry sietí, keďže z pár vrstiev nie je možné, aby sieť pochopila tak zložité príznaky ako napr. ľudská tvár. Vygenerovať príznaky u obrazových dát, ktoré by obsahovali komplexnejšie informácie nie je možné priamo zo vstupných dát. Je potreba transformovať už vygenerované príznaky (ako napr. zhluky obrazových bodov, hrany) pre získanie týchto zložitejších príznakov. Bolo vedecky dokázané, že ľudský mozog funguje na podobnom princípe – prvé neuróny hierarchie sú senzitívne na konkrétne hrany, zatiaľ čo hlbšie časti ľudského mozgu sú senzitívne na komplexnejšie štruktúry ako napr. tváre. [54, 57, 113]

Hierarchické učenie príznakov ešte pred existenciou princípu hlbokého učenia trpelo problémom miznúceho a explodujúceho gradientu, kde gradient sa zmenšil (zväčšil) natoľko, že ďalším hlbším vrstvám neposkytoval žiaden signál vhodný

na učenie. Takže funkčnosť týchto architektúr bola mnohonásobne horšia, než u tradičných metód (ako napr. SVM). Termín „hlboké učenie“ vznikol kvôli novým metódam a stratégiám navrhnutých na tvorbu týchto hlbokých štruktúr nelineárnych príznakov a prekonanie problému s miznúcim gradientom. V roku 2010 bolo dokázané, že kombináciou grafických akcelerátorov a aktivačných funkcií je možné trénovať hlboké architektúry sietí bez závažných problémov. [56, 92, 99, 108]

Stochastický gradientný zostup

Ako popisuje článok [56], najčastejšou metódou učenia hlbokých architektúr je metóda označovaná ako *Stochastický gradientný zostup – Stochastic Gradient Descent* (SGD). Po predložení vstupných vektorov z pár vybraných trénovacích vzoriek a určenia výstupu a chyby dochádza k výpočtu priemerného gradientu pre tieto vzorky a následnému zmenu váh v sieti. Proces je opakovaný toľko krát, pokiaľ sa priemerná hodnota objektívnej funkcie (funkcia určujúca chybu) neprestane zmenšovať. „Stochastický“ sa nazýva z toho dôvodu, že vybraná podmnožina trénovacích vzoriek poskytuje „zašumený“ odhad priemerného gradientu všetkých vzoriek. Táto jednoduchá procedúra poskytuje relatívne rýchly odhad váh, ktoré pre daný problém fungujú. [56]

Ak definujeme trénovaciu vzorku ako x_1, ktorá je privedená na vstup siete, dopredným prechodom siete získame x_L pre dosiahnutie predikcie. Táto hodnota musí byť porovnaná s očakávanou hodnotou odpovedajúcej vzorku x_1, z ktorej bude určená chyba z. Chyba z predstavuje učiaci signál, reprezentujúci ako veľmi sa musia váhy siete zmeniť. Modifikácia parametrov u metódy SGD prebieha pomocou

$$w_i \leftarrow w_i - \eta \frac{\partial z}{\partial w_i}$$

v jednotlivých krokov (v čase t). Ak by došlo k veľkej zmene gradientu, chybová funkcia sa zvýši. Pri každej zmene sa však parametry w_i menia len o malý krok, čo je kontrolované pomocou rýchlosti učenia η, kde $\eta > 0$, avšak zvyčajne predstavuje malé číslo (napr. $\eta = 0,001$). Po zmene váh siete na základe vzorky x_1 dôjde k zníženiu chyby pre túto konkrétnu vzorku. Tento krok však s veľkou pravdepodobnosťou zvýši chybu u iných trénovacích vzorkách. Z toho dôvodu je potreba váhy siete upravovať postupne prostredníctvom všetkých trénovacích vzoriek. Tento proces – jedna iterácia na všetkých dátach – sa nazýva v terminológií hlbokého učenia ako „epocha“. Jedna epocha vo všeobecnosti znižuje chybu na trénovacej množine, až pokiaľ nedôjde k pretrénovaniu siete na trénovacích dátach. Preto je potrebné tento proces včas zastaviť. [113]

Ostatné metódy ako sú napr. Adagrad [23], Adadelta [117], Adam [44, 89], Adamax [44], Nadam [22, 101], logicky vychádzajú z metódy SGD a snažia sa zjednodušiť a zrýchliť konvergenciu siete, prípadne zaistiť priebeh učenia.

Vrstvy

Stavebným blokom hlbokých neurónových sietí sú vrstvy, ktoré je možné chápať ako zapuzdrený objekt, ktorý transformuje váhovaný vstup pomocou množiny nelineárnych funkcií a predáva tieto hodnoty ďalšej vrstve. Vrstvy sú uniformné, vždy využívajú jeden typ aktivačnej funkcie. Prvá vrstva hlbokej neurónovej siete je nazývaná aj „vstupná“, posledná vrstva je „výstupná“. Počet vstupných a výstupných vrstiev môže byť v niektorých prípadoch vyšší než 1. Vrstvy nachádzajúce sa medzi vstupnou a výstupnou vrstvou sa označujú ako „skryté“.

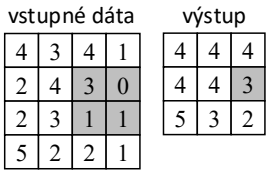
Plne prepojená vrstva predstavuje klasickú doprednú vrstvu pozostávajúcu z umelých neurónov, ktorých vstupy sú pripojené na predošlú vrstvu a výstupy sú pripojené na nasledujúcu vrstvu. Tieto vrstvy sa využívajú prevažne na posledných miestach architektúr DNN, kde zabezpečujú proces klasifikácie alebo regresie. [48, 54]

Konvolučná vrstva je inšpirovaná funkciou zrakového nervu, kde neuróny vrstvy reagujú na vstup aktivácií okolitých neurónov podľa zadanej veľkosti konvolučného jadra. Využívajú matematickú operáciu diskrétnej konvolúcie, ktorej dvojrozmerná varianta je znázornená príkladom na Obr. 1.2. [48, 113]

Obr. 1.2: Príklad 2D konvolúcie

Siete založené na konvolučných vrstvách (CNN) našli uplatnenie hlavne v klasifikácií obrazových dát. Dokážu generalizovať dáta pri omnoho nižšom počte neurónov, než by potrebovala plne prepojená sieť. [95, 43, 48]

Vrstva združovania označovaná aj ako „pooling vrstva“ poskytuje možnosť podvzorkovania dát, čo znamená zmenšenie množstva výstupov a výpočtovej náročnosti. Zvyčajne sa používajú hneď po konvolučných vrstvách, kde posunom konvolučných jadier po jednotlivých vstupoch vznikajú duplikované dáta. Najčastejšie sa využívajú vrstvy združovania pomocou hľadaní maxima alebo spriemerovaním hodnôt. Príklad pre maxpooling s jadrom veľkosti 2 x 2 a jednotkovým krokom je na Obr. 1.3. [48, 113]



Obr. 1.3: Príklad podvzorkovania 2D matice združovaním maxima

Vrstva vyradenia alebo tzv. „dropout vrstva“ využíva techniku náhodného zahadzovania častí dát, vďaka čomu dochádza ku zvýšeniu generalizácie metód hlbokého učenia. Napr. u siete VGG-16 [95], ktorá predstavuje sieť zloženú z konvolučných vrstiev dosahujúca „state-of-the-art“ úspešnosti na klasifikácií obrazových dát, je použité 50%-né vyradenie v dvoch vrstvách. Vďaka vrstve vyradenia je možné predísť preučeniu a natrénovať tak robustnejší model. [113]

1.2.2 Metódy založené na zhlukovaní

Zhlukovanie (anglicky „embedding“) možno vysvetliť na jave, kedy jedna inštancia matematickej štruktúry obsahuje inú inštanciu. Ak nejaký objekt X je vnorený v objekte Y, zhlukovanie je dané injekčnou funkciou a mapou zachovávajúcou štruktúru $f : X \to Y$. Metódy dolovania znalostí z textových dát často využívajú modely zhlukovania slov $W : slovo \to \mathbb{R}^n$, kde n definuje dimenzionalitu vektorového priestoru (napr. 100 vstupných parametrov DNN siete), pomocou ktorých sa vytvárajú modely jazyka, učenia príznakov textových dát a spracovania prirodzeného jazyka. Každá dimenzia takto vytvoreného vektoru môže predstavovať príznak témy, do ktorej dané slovo spadá, kontext v ktorom sa slovo vyskytuje, prípadne iné vlastnosti daného slova. [4, 70]

Znalosť jazyka a gramatiky, ako základný predpoklad k strojovému pochopeniu textu, môže byť zakódovaná prostredníctvom modelov zhlukovania, konkrétne modelov zhlukovania slov. Ako popisuje článok [52], metóda kódujúca každé slovo prostredníctvom viacrozmerného vektoru je často využívaná v súčasných „state-of-the-art“ metódach. Tieto vektory sú hľadané pre každé slovo s použitím obrovského množstva textových dát.

Word2vec je v súčasnosti najpopulárnejšia modelová architektúra pre výpočet vektorových reprezentácií slov, vytvorená výskumným tímom spoločnosti Google, neskôr analyzovaný a podrobne vysvetlený v článku [31]. Využíva architektúru BoW rozšírenú o kontext dvoch minulých a dvoch nasledujúcich slov, označovanú ako *kontinuálny batôžtek slov – Continous Bag-of-Words* (CBOG), a architektúru

tzv. „skipgramového“ modelu, ktorý využíva predikciu slov v určitej vzdialenosti od daného slova. [75]

GloVe využíva pre určenie sémantickej podobnosti Euklidovskú vzdialenosť. V niektorých testoch syntaktických a sémantických úloh dosahuje lepšie výsledky v porovnaní s modelom Word2vec. [75, 80, 118] Model ako taký vychádza však z logiky modelu Word2vec.

Štatistická analýza výskytu slov v korpuse je primárnym zdrojom informácií pri tvorbe metódy „bez učiteľa“, ako popisuje článok [80] s postupom návrhu modelu „GloVe“ pre vektorovú reprezentáciu pomocou tzv. globálnych vektorov (model zachytáva globálne štatistiky jednotlivých slov). Ak maticu výskytov jednotlivých slov v rovnakom kontexte označíme ako X, ktorej hodnory X_{ij} predstavujú, koľkokrát sa slovo j vyskytlo v rovnakom kontexte ako slovo i, potom $X_i = \sum_k X_{ik}$ bude počet výskytov slova i, môžeme určiť pravdepodobnosť výskytu slova j v kontexte slova i ako

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \ ,$$

čo predstavuje základný kameň pre modely zhlukovania slov. Príklad je znázornený v Tab. 1.4 Myšlienkou u modelu GloVe je zamerať sa na pomery pravdepodobností výskytu v rovnakom kontexte, než na pravdepodobnosti samé. Teda na pomere dvoch rôznych pravdepodobností P_{ik}/P_{jk}, kde dochádza k závislosti na troch slovách i, j a k. Najzákladnejší model môže byt definovaný ako

$$F(w_i, w_j, \widetilde{w_k}) = \frac{P_{ik}}{P_{jk}} \ ,$$

kde $w \in \mathbb{R}^d$ sú vektory slov a $\widetilde{w} \in \mathbb{R}^d$ je vektor kontextu slova. Pri tvorení matice X musí byť definovaná veľkosť kontextového okna d pre zabezpečenie znalosti, že veľmi vzdialené slová obsahujú len málo relevantných informácií o vzťahoch medzi slovami. Model GloVe teda vytvára dve množiny vektorov slov W a $\widetilde{W}$, ktoré sú zhodné v prípade symetrického X. V opačnom prípade je vhodné tieto vektory sčítať. [80]

Tab. 1.4: Príklad pravdepodobnostného zakódovania textu modelom GloVe [80]

Pravdepodobnosť	$k =$ pevný	$k =$ plynný	$k =$ voda	$k =$ móda
$P(k\|\text{ľad})$	$1,9 \cdot 10^{-4}$	$6,6 \cdot 10^{-5}$	$3,0 \cdot 10^{-3}$	$1,7 \cdot 10^{-5}$
$P(k\|\text{para})$	$2,2 \cdot 10^{-5}$	$7,8 \cdot 10^{-4}$	$2,2 \cdot 10^{-3}$	$1,8 \cdot 10^{-5}$

2 Ciele dizertačnej práce

Práca skúma súčasné metódy, ich možné zvýšenie presnosti vďaka optimalizačným metódam, ako aj nové metódy riešenia problému porozumenia textu s modelovaním kognitívneho správanie človeka pri spracovaní textových dát. Vďaka vývoju nových technológií a zdokonaľovaniu výrobných procesov dochádza k nárastu výpočtového výkonu, čo sa odzrkadľuje aj na smerovaní súčasných metód pre dolovanie znalostí z textových dát. Ich presnosť však stále nedosahuje presnosti človeka.

Hlavným cieľom dizertačnej práce je navrhnúť metódu pre strojové porozumenie neštruktúrovaným textovým dátam, teda bez ohľadu na charakter vstupných textových dát. Návrh metódy bol riadený niekoľkými požiadavkami: 1) dostatočná všeobecnosť, 2) znovu-použiteľnosť pre rôzne jazyky (prirodzené i strojové) bez nutnosti akéhokoľvek zásahu, z čoho vyplýva minimalizácia jazykovo závislých častí, v ideálnom prípade ich úplné odstránenie, a 3) dostatočnú presnosť validovanú na vybranom probléme.

Čiastkové ciele dizertačnej práce je možné rozdeliť do nasledujúcich bodov:

- vytvorenie databáz textových dát vhodných pre overenie a porovnanie úspešnosti súčasných metód,
- vytvorenie metódy pre klasifikáciu textu do viacerých emočných tried – návrh vychádza z tradičných metód, keďže k dispozícií môže byť len relatívne malé množstvo dát,
- návrh metódy trénovania viac-modelovej architektúry pre klasifikáciu viacerých tried s možnosťou implementácie optimalizačných metód,
- návrh optimalizačných metód s cieľom zvýšenia presnosti klasifikácie tradičných metód a minimalizácia možnosti pretrénovania,
- abstrahovanie metódy klasifikácie textov pomocou Big Data prístupu, ktorý zvyšuje pamäťovú a časovú náročnosť,
- vytvorenie optimalizačných metód použiteľných aj pre Big Data problém,
- výskum a vývoj metódy pre porozumenie textu založenej na hlbokom učení, so zameraním na všeobecnosť,
- experimentálne overenie navrhnutých metód na vytvorených databázach textových dát, ako aj experimentálne overenie navrhnutých optimalizačných metód a ich vplyv na úspešnosť klasifikačných metód.

Práca má za cieľ experimentálne overiť funkčnosť navrhnutých metód, pričom sa zameriava na extrakciu jednoduchých informácií prostredníctvom klasifikácie textových dát. V práci sú navrhnuté jazykovo nezávislé optimalizačné metódy a metódy na jazykovo nezávislé porozumenie textovým dátam validované na vybraných konkrétnych problémoch.

3 Navrhnuté metódy riešenia

Možné riešenia problému, ktoré táto práca predkladá, smerujú k úplnému odstráneniu jazykovo závislých častí. Výskum a vývoj všeobecnej metódy pre porozumenie textu prebiehal vo viacerých krokov. Jedná sa o návrh metódy pre klasifikáciu viacerých tried a návrh jazykovo nezávislých optimalizačných metód pre tento systém, zovšeobecnenie riešenia klasifikácie textu pomocou distribuovaných výpočtov a prístupu Big Data, čím došlo k odstráneniu niektorých jazykovo závislých častí, návrh optimalizačných metód v tomto systéme, a nakoniec návrh metódy pre porozumenie textu bez znalosti jazyka a gramatiky.

3.1 Návrh metódy klasifikácie emócií

Problém dolovania znalostí z textových dát je možné demonštrovať na probléme klasifikácie emócií autora textu. Emóciu autora je často možné pochopiť len z celého kontextu. Automatická extrakcia emócie je zložitá a jednoznačne určiť z pár slov zachytenú emóciu autora nie je možné, zvlášť v prípade sarkazmu a irónie.

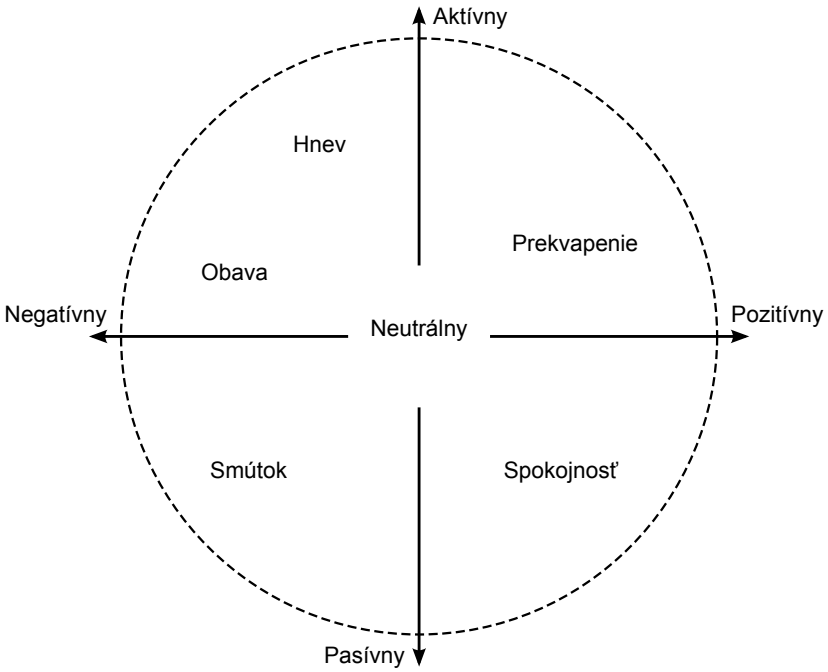
Veľké spoločnosti sa musia často vysporiadať na zákazníckych podporách s obrovským návalom textových správ, na ktoré je nutné včas odpovedať. Jednotlivé správy však nemajú rovnakú prioritu – niektoré je potreba uprednostniť (napr. správa od nespokojného zákazníka, u ktorého hrozí prechod ku konkurencií), iné správy nie sú v danom čase dôležité (napr. poďakovanie zákazníka). Filtrovanie správ, resp. určovanie priorít správ, je možné vďaka klasifikácií.

Cennou informáciou pre správne uprednostnenie textovej správy na zákazníckej podpore je emócia autora textu – vydolovaná emočná trieda, vďaka ktorej môže byť v systéme podpory zákazníkov pridelená priorita tejto správe.[1]

3.1.1 Definícia problému

Počas skúmania historických dát z podpory zákazníkov a pri tvorbe nových vzoriek do databáze textov bolo zistené, že najčastejšie sa vyskytujúce emócie v tomto systéme sú: hnev, smútok, spokojnosť, prekvapenie a obava. Tieto emócie sa javia ako dominantné z celkovej množiny. Jednotlivé emočné triedy sú analyzované a definované z pohľadu akustického modelu publikovaného v článku [19], zjednodušené grafické znázornenie tohto modelu je na obr. 3.1. Tento model popisuje niekoľko

[1]Náväznosť na projekt „Výzkum a vývoj technologie pro detekci emocí v nestrukturovaných datech“ so spoločnosťou Webnode pre detekciu nespokojných zákazníkov. Projekt „FR-TI4/151“ riešený v období 1.5.2012–31.12.2015, financovalo „Ministerstvo průmyslu a obchodu ČR“.



Obr. 3.1: Súradnicový systém v akustickom modele emócií

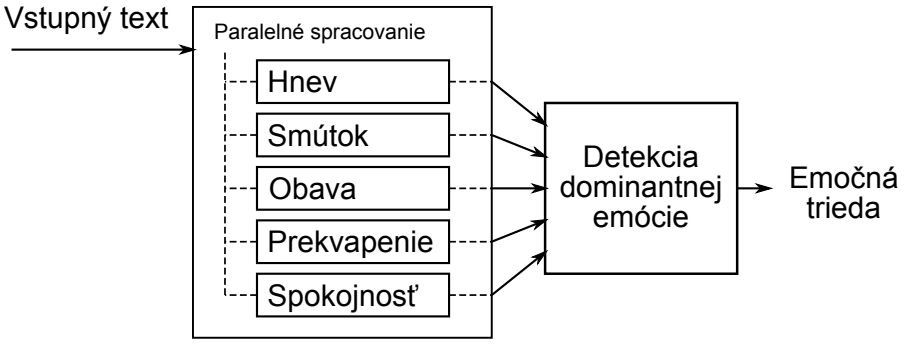
emočných tried, pričom sa vychádza z dvoch hlavných parametrov – miera pozitivity (horizontálna os) a miera aktivity (vertikálna os). Pri detekcií nespokojných zákazníkov má väčší význam horizontálna os, teda miera pozitivity alebo negativity daného textu.

Problémovou emóciou v tomto návrhu je emócia „prekvapenie“, keďže do tejto emócie možno zaradiť aj emóciu „milo prekvapený“. Táto emócia v textoch evokuje šťastie a teda patrí do emočnej triedy „spokojnosť“. Texty s týmto emočným nábojom má problém správne zaradiť do jednej z definovaných kategórií aj človek. Dosiahnuť pri takomto návrhu dostatočne kvalitnú databázu – a teda aj vysokú úspešnosť správnej klasifikácie vzoriek textu je možné pomocou extrakcie vzoriek s mylnou klasifikáciou emócie „milo prekvapený“ a presunutím týchto vzoriek do triedy „spokojnosť“.

Pre automatickú detekciu nespokojných zákazníkov je nutné navrhnúť metódu na detekciu emócie obsiahnutých vo vstupnom texte a na základe detekovanej emócie určiť prioritu správy. Metóda by mala mať na výstupe jednu z možných detekovaných emócií, resp. mieru istoty, že vstupný text spadá práve do danej triedy. Kapitola 1.1.2 popisuje, že SVM spadajúca pod tradičné metódy, vytvára nadrovinu, ktorá separuje dve triedy dát.

3.1.2 Popis navrhnutej štruktúry

Základnou myšlienkou, je natrénovať jeden model pre každú jednu emočnú triedu. Každý z týchto modelov bude rozhodovať, či vstupný text patrí do danej triedy (pozitívny výsledok) alebo nie (negatívny výsledok) – teda bude separovať dve triedy jednou nadrovinou. Výstupom takejto štruktúry je 5 reálnych čísel reprezentujúcich mieru istoty jednotlivých emočných tried (hnev, smútok, spokojnosť, prekvapenie a obava), z ktorých je následne potrebné určiť dominantnú emóciu vstupného textu. Popisovaná štruktúra sa nachádza na obr. 3.2.



Obr. 3.2: Štruktúra systému klasifikácie emócií

Funkčnosť systému pozostáva zo spoločného predspracovania vstupného textu, aplikovaním piatich klasifikačných modelov emócií a následnej detekcie dominantnej emócie. Trénovanie celého systému predstavuje komplexný proces, ktorý svojou zložitosťou prináša problémy s nasadením bežných optimalizačných metód, ako sú napr. redukcia dimenzionality (dopredný výber vstupných parametrov, postupná spätná eliminácia, prípadne ich kombinácia), odstránenie redundantných parametrov, apod.

Text musí do klasifikátorov vstupovať vo vhodnom formáte. O toto sa stará algoritmus predspracovania vstupného textu, ktorý transformuje neštrukturovaný text na vektor príznakov (parametrov). Vstupný text je rozdelený na slová (tokeny), následne sú skontrolované preklepy, prípadne doplnená diakritika, jednotlivé slová sú zmenené do pôvodného tvaru, synonymá sú nahradené hypernymom (resp. zastupujúcim slovom z danej množiny synoným – zvyčajne prvé slovo podľa abecedy) a výstupný text je prevedený pomocou metodiky hodnotenia relevancie TF–IDF na vektor reálnych čísel. [82]

V oboch prípadoch – pri trénovaní a pri nasadení natrénovaného systému – je nutné myslieť na náročnosť predspracovania vstupného textu. Pokiaľ má byť minimalizovaná časová a pamäťová náročnosť, algoritmus predspracovania musí byť

aplikovaný na vstupné dáta len raz. Teda ak by každý emočný model mal vlastný algoritmus predspracovania, časová a pamäťová náročnosť by vzrástla 5-násobne.

Vďaka paralelizácií trénovania klasifikátorov, prípadne distribúciou na viaceré výpočtové stanice, je možné dosiahnuť značné zníženie časovej náročnosti trénovania. Do procesu trénovania jednotlivých modelov vstupuje jedna množina dát – vopred predspracovaná, ktorá je ďalej rozdelená podľa pridelenej značky emócie. Množiny sú preskupené podľa významnosti k danej emočnej triede. Je to kvôli potrebe rozdelenia dát na množinu s pozitívnym a množinu s negatívnym výstupom.

Keďže spomínané predspracovanie textu je silno jazykovo závislé, prvým krokom k zníženiu tejto závislosti je smerovať aspoň novo navrhnuté optimalizačné metódy na úplnú jazykovú nezávislosť. Navrhnuté optimalizačné metódy musia byť dostatočne všeobecné a aplikovateľné na rôzne jazyky, aby nebolo potrebné pri tvorení rovnakého systému pre iný jazyk meniť chod prípadne štruktúru tohto systému.

3.1.3 Popis trénovacích a testovacích dát

Dáta boli manuálne zozbierané z reálneho systému podpory zákazníkov, pre český[2] a anglický[3] jazyk. Texty sú relatívne krátke a v mnohých prípadoch obsahujú len jednu vetu. Zvyčajne zachytávajú diskusiu k produktom a službám, prípadne technicky zamerané otázky na funkčnosť produktu, možné nastavenia, otázky smerované k existujúcemu manuálu apod. Pre zvýšenie všeobecnosti systému bola databáza rozšírená o frázy z verbálnej komunikácie s obsiahnutým emočným nábojom. Tieto texty boli zozbierané pre budúce možné nasadenie pre akýkoľvek systém podpory zákazníkov. Texty boli manuálne označkované jednou z emočných tried prostredníctvom skupiny ľudí, pričom priemerný počet značiek na jeden text bol 1,17. Tab. 3.1 zachytáva počty vzoriek jednotlivých emócií.

Tab. 3.1: Veľkosti databáz testovaných jazykov

Jazyk	**Počet vzoriek emócie**				
	hnev	smútok	obava	prekvapenie	spokojnosť
Čeština	351	441	241	56	992
Angličtina	114	106	104	110	112

[2]Databáza českých textov vznikla vďaka spolupráci so spoločnosťou Webnode CZ s.r.o. a projektu „Výzkum a vývoj technologie pro detekci emocí v nestrukturovaných datech“ vedeného pod identifikátorom „FR-TI4/151“.

[3]Databáza anglických textov vznikla vďaka spolupráci s „Amity University, Noida, Uttar Pradesh, India“, výsledky spolupráce sú publikované v článku [82].

Pre trénovanie jednotlivých emočných klasifikátorov bolo potrebné dáta rozdeliť tak, aby bolo zabezpečené vhodné zastúpenie jednotlivých vzoriek (emócií) pre pozitívnu a negatívnu množinu pre daný klasifikátor. V experimente bol použitý princíp vyváženia jednotlivých množín dát, keďže vytvoriť úspešný klasifikačný model na nevyváženej množine dát je omnoho zložitejšie a náročnejšie u tradičných metód – ako popisujú články [29] a [93] – tým viac u problému klasifikácie niekoľkých emócií. Preto pre zjednodušenie problému boli tieto množiny vyvážené.

Množiny určené pre trénovanie a testovanie klasifikátorov boli zostavené z vytvorenej databázy označených textov. Trénovacia a testovacia časť boli vyvážené u každej emócie, rovnako ako pozitívna a negatívna množina. Obr. 3.3 a Obr. 3.4 zobrazujú histogramy vyvážených množín a porovnávajú zastúpenie jednotlivých emócií. Vyváženie prebehlo podvzorkovaním väčšej množiny – množiny s vyšším obsahom vzoriek. V tomto prípade šlo vždy o negatívnu množinu, keďže negatívna časť sa skladá vždy zo všetkých ostatných emócií. Pri podvzorkovaní sa kládol dôraz na rovnomerné zastúpenie všetkých ostatných emócií v negatívnej množine. Napr. u klasifikátoru emócie „spokojnosť“ sú v pozitívnej množine dát ($\mathbf{X}_p$) texty s prevažujúcou emóciou „spokojnosť“, no v negatívnej množine dát ($\mathbf{X}_n$) musia byť zastúpené texty zo všetkých ostatných množín, ideálne s rovnomerným zastúpením ostatných emócií. Teda ak je množina všetkých dát definovaná ako $\mathbf{M}$, obsah pozitívnej množiny je možné definovať ako

$$\mathbf{X}_p = \mathbf{M}_{spokojnost}$$

a obsah negatívnej množiny

$$\mathbf{X}_n = \mathbf{A} \cup \mathbf{B} \cup \mathbf{C} \cup \mathbf{D},$$

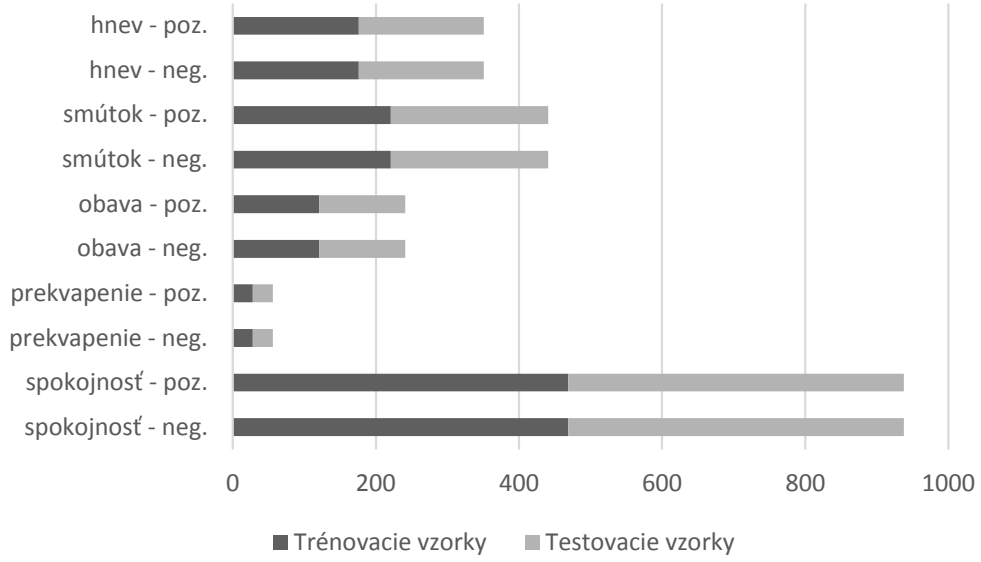
kde zložky $\mathbf{A}$ až $\mathbf{D}$ sú definované ako

$$\mathbf{A} \subset \mathbf{M}_{hnev}; \mathbf{B} \subset \mathbf{M}_{smutok}; \mathbf{C} \subset \mathbf{M}_{obava}; \mathbf{D} \subset \mathbf{M}_{prekvapenie},$$

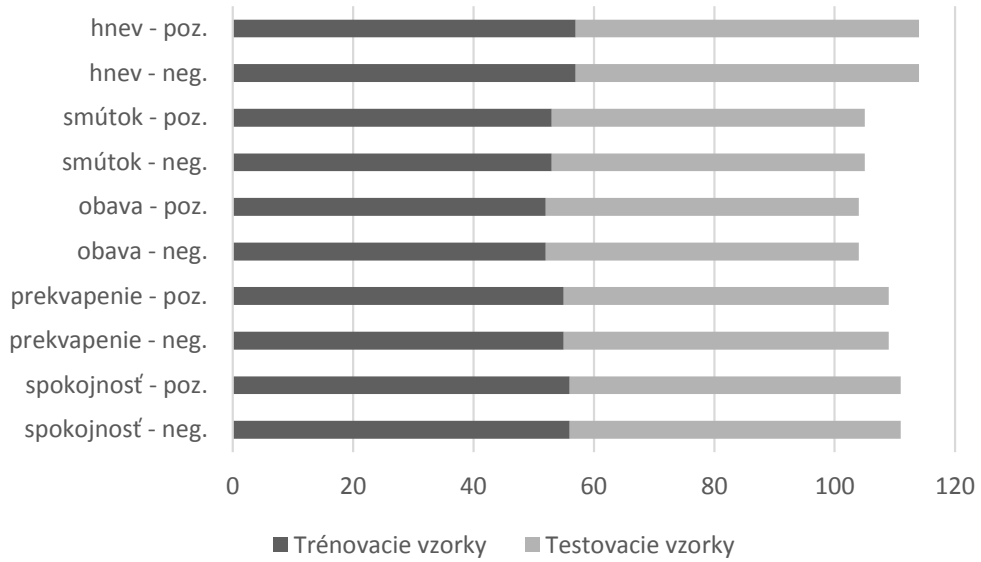
pri zachovaní podmienky

$$|\mathbf{X}_p| = |\mathbf{X}_n|.$$

Vytvorené databázy textov majú rôzne veľkosti (ako zobrazuje Tab. 3.1) a rôzne zastúpenie jednotlivých množín (ako zobrazujú histogramy na Obr. 3.3 a Obr. 3.4). Je to spôsobené tým, že databáza pre klasifikáciu českých textov vznikala zároveň s definíciou možných emočných tried, prvé pokusy prebiehali prostredníctvom definovania emócií v súradnicovom systéme (už bolo popísané v kapitole 3.1.1), kde sa postupne kryštalizovalo, ktoré emócie budú potrebné v tomto systéme. Databáza anglických textov bola následne vytvorená s vedomím o potrebných emočných triedach a potrebou vyvážených množín jednotlivých emócií a s úlohou definovať aspoň 100 vzoriek od každej emócie.



Obr. 3.3: Histogram vyvážených množín českého jazyka



Obr. 3.4: Histogram vyvážených množín anglického jazyka

K jednotlivým jazykom je potreba vytvoriť ďalšie pomocné databázy textov (resp. slov) potrebné k predspracovaniu vstupného textu. Tieto databázy sú potrebné ako pri trénovaní modelu, tak pri použití natrénovaného modelu. Model

založený na tradičnej metóde vyžaduje na vstupe predspracované dáta v rovnakej forme ako pri trénovaní. Pre rýchly beh je potreba držať tieto dáta neustále v pamäti, čím sa rapídne zvyšuje pamäťová náročnosť tohto riešenia. Ide o nasledujúce množiny textových reťazcov:

1. databáza kompletne všetkých slov daného jazyka, aby bolo možné vyhľadať preklepy na základe najmenšej vzdialenosti dvoch textových reťazcov,
2. databáza synoným pre zmenšenie stavového priestoru,
3. databáza stop slov pre odfiltrovanie slov bez významu, resp. slov, ktoré by mohli spôsobovať zbytočné zmätenie modelu – napr. spojky, predložky apod.

3.1.4 Popis predspracovania dát

Predspracovanie textu zabezpečuje transformáciu vstupného textu do vektoru príznakov reprezentujúcich zastúpenie jednotlivých slov. Ako už z popisu vyplýva, predspracovanie počíta s výskytom slov vo vstupnom texte. Teda je potreba vstupný reťazec znakov správne rozdeliť na slová (tokeny) pomocou tokenizácie. Tokenizácia sama o sebe nám vnáša podmienku na syntax jazyka vstupného textu. Teda takéto riešenie nebude možné aplikovať napr. na čínsky jazyk, kde syntax jazyka nie je založená na jednoznačne oddelených slovách.

Tokenizácia zabezpečuje rozdelenie textu na jednotlivé slová (tokeny). Pred tokenizáciou je však potreba detekovať špeciálne prípady, ktoré môžu byť z textu odstránené, pretože nemá zmysel, aby sa dostali do ďalšieho spracovania – napr. e-mailové adresy alebo http adresy, keďže nesúvisia s emóciou textu. Ďalej je potreba identifikovať prípadný výskyt emotikon, čísel a prípadne vulgarizmov, ktoré je potreba nahradiť zastupujúcou značkou. Následne takto vyčistený text môže byť rozdelený na jednotlivé slová pomocou bielych znakov a znakov interpunkcie.

Kontrola pravopisu je ďalším krokom, keďže vstupný text môže byť vkladaný človekom (ide o spracovanie prirodzeného jazyka), môžu sa teda v texte vyskytnúť pravopisné chyby a preklepy. Pri predspracovaní textu je potreba na toto myslieť, inak sa každá chyba môže prejaviť na ďalšom zmätení modelu. Kontrola pravopisu využíva už spomínanú databázu všetkých slov daného jazyka, ako aj všetkých možných tvarov. Každý token je porovnaný s touto databázou. Pokiaľ sa v databáze priamo nachádza, považuje sa za slovo bez chyby. Pokiaľ sa slovo v databáze nenachádza, hľadajú sa slová ktoré majú najmenšiu Levensteinovú vzdialenosť k hľadanému slovu. Podľa článku [76] je vzdialenosť dvoch slov určená ako počet písmen, ktoré musíte zmeniť v jednom slove pre dosiahnutie slova druhého. Algoritmus je

možné ďalej ozvláštniť napr. o zníženie vzdialenosti v prípade, ak ide len o zmenu v diakritike.

Lemmatizácia transformuje slová do „lemy“, teda do slovníkového tvaru tokenu. Aplikovaním lemmatizácie dochádza k výraznému zmenšeniu stavového priestoru, keďže každý možný tvar slova je nahradený zástupcom – základným tvarom slova. Lema sa uvádza vždy malým začiatočným písmenom, takže je vyriešený napr. aj problém začiatku vety, kde je u rovnakého slova použité veľké písmeno, teda by šlo o iný token. Pre svoju funkciu lemmatizácia využíva databázu tvarov a ich slovníkových tvarov, v ktorej prebieha vyhľadávanie. Často je lemmatizícia nahradzovaná efektívnejšou metódou – stemmingom, keďže ide o pamäťovo menej náročnú operáciu – využíva iba súbor pravidiel pomocou ktorých je možné dopracovať sa ku koreňu slova.

Nahradenie synonym zabezpečuje ďalšie zmenšenie stavového priestoru a teda i dimenzionality vstupu trénovaného modelu. Slová ktoré predstavujú rovnaký význam a teda i emocionálne zafarbenie môžu byť nahradené zástupcom z danej skupiny synoným – hypernymom. Tento zástupca zabezpečuje, že vo vstupnom texte sa nestratí žiadna informácia a zároveň že dimenzionalita vstupu bude skutočne znížená, keďže každé jedno slovo túto vlastnosť zvyšuje.

Stop slová slúžia na odstránenie slov, ktoré by mohli spôsobiť zmätenie modelu. To by malo za následok nižšiu úspešnosť klasifikácie. Najčastejšie ide o slová, ktoré sa v jazyku vyskytujú často. Samé o sebe však nenesú žiadnu významovú informáciu, zvyčajne v jazyku iba dotvárajú syntax. Do tejto kategórie spadajú aj slová ako predložky, spojky, zámená apod. V zložitejších implementáciach predspracovania však môžu predstavovať dodatočnú informáciu, ktorá by mohla zvýšiť presnosť klasifikácie (napr. implementáciou n-gramov). Databázy stop slov sú voľne dostupné takmer ku každému jazyku – môžu však spôsobiť aj zníženie úspešnosti, pokiaľ takáto databáza odstráni aj slová nesúce informáciu potrebnú ku klasifikácií.

3.1.5 Popis experimentu

Prvé pokusy prebiehali na klasifikácií do 2 tried, pričom bola snaha nájsť najvhodnejší klasifikátor pre textové dáta. V čase riešenia sa ako najvhodnejším zdal byť klasifikátor SVM. Vstupné dáta boli v experimente reprezentované váhami slov, teda hodnoty vypočítané pomocou TF–IDF. Experimenty zamerané na mieru pozitivity,

teda s klasifikáciou do 2 tried (pozitívny a negatívny), dosahovali na vybranom klasifikátore 87,0 % úspešnosť. Pre nasadenie v systéme podpory zákazníkov je však potreba rozlišovať viaceré emócie.

Existujúce implementácie pre klasifikáciu 5 rôznych emočných tried (plus prípadne ďalšej triedy – neutrálneho textu) pomocou jedného modelu nedosahovali príliš dobré výsledky. Šlo hlavne o implementácie rozhodovacích stromov, náhodných lesov apod., kde aj po niekoľkých úpravách bola dosiahnutá úspešnosť len 38,4 %. Po odstránení niektorých tried a klasifikovaní len do troch tried (hnev, spokojnosť a smútok) opäť pri použití jedného natrénovaného modelu, bola nameraná úspešnosť na testovacích dátach len 51,5 %. Po týchto experimentoch bol zvolený prístup, ktorý popisuje táto kapitola – natrénovanie jednoduchého modelu pre každú emočnú triedu tak, že každý model každý bude rozhodovať či vstupný text patrí do danej triedy alebo nie, resp. s akou pravdepodobnosťou, akou mierou istoty.

Ani riešenie pomocou viacerých modelov nedosahuje výsledky, ktoré by sa mohli rovnať s človekom. Zvýšiť úspešnosť klasifikácie je možné pomocou optimalizačných metód, ktoré boli navrhnuté pre tento problém.

3.1.6 Navrhnuté optimalizačné metódy

Úspešnosť klasifikácie bola zvýšená pomocou navrhnutých optimalizačných metód, ktoré boli následne publikované v článku [87] (IF pre rok 2016 = 0,945). Ide o metódy: a) sekvenčná eliminácia parametrov, ktorá je schopná bežať paralelne nad viacerými klasifikátormi (submodelmi emočných tried), b) metóda zoskupovania slov, ktorá zabezpečuje komprimáciu vstupných dát, a c) metóda rozširovania o staticky definované vzorky behom praktického testovania, ktorá je určená pre konečné ladenie systému. Všetky metódy boli navrhnuté ako jazykovo nezávislé, teda sú aplikovateľné na rôzne jazyky bez nutnosti akejkoľvek zmeny.

Predspracovanie vstupných dát predstavuje výpočtovo náročnú operáciu, preto je nutné minimalizovať počty volaní tohto procesu. Predspracovanie je volané pre všetky trénovacie dáta len raz, kde sú aplikované všetky časti z kap. 3.1.4, až po vygenerovanie vektoru zastúpenia jednotlivých slov. Zároveň je vytvorený zoznam povolených slov, čím je zabezpečené správne poradie neskôr pri testovaní alebo aplikovaní modelu, ako aj zabezpečenie odfiltrovania neznámych slov. Dáta sú následne uložené do dočasného úložiska pre minimalizovanie časovej náročnosti.

U každej zmeny v systéme je potreba dáta predspracovať znova. Až po tejto operácií, môže byť zmena evaluovaná a rozhodnuté, či je zmena akceptovateľná a či vedie k dosiahnutiu lepších výsledkov. Tento proces je o to náročnejší, lebo musí počítať s existenciou viacerých modelov (modelov jednotlivých emócií).

Úspešnosť bola zvýšená pomocou 3 novo navrhnutých optimalizačných metód

pre tento problém. Tieto metódy modifikujú priebeh trénovania a teda musia byť vhodne implementované v procese trénovania viacerých modelov. Ide o metódy:

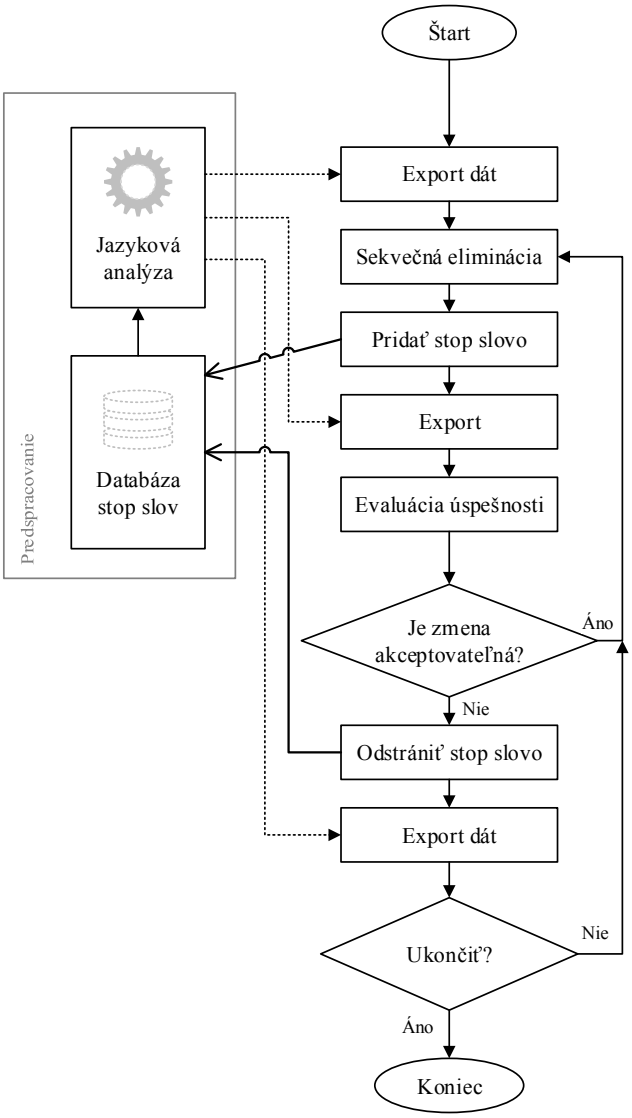
- **sekvenčná eliminácia parametrov**, ktorá zabezpečuje zníženie dimenzionality všetkých modelov, znižuje možnosť pretrénovania, zvyšuje úspešnosť vďaka eliminácií parametrov, ktoré spôsobujú zmätenie modelu,
- **zoskupovanie slov**, ktoré predstavuje možnosť definovať skupiny slov pre daný jazyk, čím sa znižuje dimenzionalita vstupu a možnosť pretrénovania,
- **rozširovanie o staticky definované vzorky**, pomocou ktorého je možné do trénovacej množiny pridať všeobecne známe frázy s daným emočným nábojom, cielene upravuje správanie modelov, ako aj ich výsledné úspešnosti, pôsobí ako dolaďovací nástroj pri nasadení systému do produkčného prostredia.

Tieto optimalizačné metódy ovplyvňujú aj ostatné modely emócií, nie len model, na ktorý bola daná zmena smerovaná. Napr. pri definícií novej frázy (v metóde rozširovania o staticky definované vzorky) do emócie hnev je táto fráza automaticky vložené do negatívnych množín ostatných emočných modelov. Týmto je cielené na dosiahnutie vyššej úspešnosti, resp. na vyriešenie konkrétneho nefunkčného vzorku. Naopak napr. u metódy sekvenčnej eliminácie parametrov môže dojsť k eliminácií parametru, ktorý spôsobuje zmätenie jedného modelu, ale u iného modelu môže byť tento parameter dôležitý pre správnu klasifikáciu. Toto prepojenie nie je nevyhnutné, ale oddelenie by spôsobilo omnoho vyššiu pamäťovú a výpočtovú náročnosť, keďže jednotlivé databázy by museli byť upravované pre jednotlivé modely oddelene, predspracovanie by tiež bolo nutné spustiť 5x (pre každý emočný model). Preto je jednoduchšie z hľadiska použitia tohto systému návrh smerovať na jednotné predspracovanie.

Sekvenčná eliminácia parametrov

Jednotlivé slová vstupného textu majú rôznu váhu v procese emočnej analýzy. Nie všetky parametre vstupu sú však pre umelú inteligenciu dôležité. Niektoré slová sú na vstupe prakticky úplne zbytočné – hlavne tie, ktoré neobsahujú žiaden emočný náboj. Naopak, niektoré slová spôsobujú len zmätenie modelu a znižujú výslednú úspešnosť modelu. Tieto slová môžu byť eliminovanú už pri predspracovaní.

Obr. 3.5 zjednodušene popisuje životný cyklus trénovania. Väčšina blokov pracuje pomocou paralelizácie pre zníženie časovej náročnosti. Časť so sekvenčnou elimináciou, ako názov napovedá, je spracovaná sekvenčne pre každý emočný model. Blok „Export dát“ predstavuje predspracovanie vstupných textových dát určených na trénovanie. Blok „Sekvenčná eliminácia“ je vykonaný iba pre jeden vybraný emočný model (vždy iný, postupne sa vystriedajú všetky) a je vyhľadaný presne jeden parameter určený k eliminácií. Tento parameter (slovo) je pridaný do databázy stop slov,



Obr. 3.5: Zjednodušený graf životného cyklu trénovania systému klasifikácie emócií

aby k eliminácií dochádzalo už pri predspracovaní. Následne je táto zmena evaluovaná na validačných dátach pre každý model, simulácia opäť beží paralelne. Kvôli závislosti medzi modelmi nie je možné elimináciu vykonávať paralelne. Po evaluácií sa porovná predošlá celková úspešnosť systému, ak došlo k zvýšeniu úspešnosti o nastavený prah (dynamicky menený) je zmena akceptovateľná a eliminované slovo je zapísané do databázy stop slov nastálo. V opačnom prípade sa eliminované slovo odstráni z databázy stop slov.

Metóda sekvenčnej eliminácie parametrov je založená na známej a rozšírenej metóde spätnej eliminácie parametrov[4]. Táto metóda znižuje redundanciu modelu,

[4]Viac informácií možno nájsť v článkoch [42], [47] a [102]

resp. dimenzionalitu vstupu, kde na vstupe sú ponechané len najdôležitejšie parametre. V každom kroku sekvenčnej eliminácie je vybraný jeden parameter (jedno slovo), ktoré spôsobuje najväčšie zmätenie daného emočného modelu. Eliminácia tohto parametru môže zvýšiť presnosť daného emočného modelu, ale zároveň znížiť presnosť iného emočného modelu. Napr. pri eliminácií slova „cena“ môže byť dosiahnutá vyššia prenosť klasifikácie emócie „spokojnosť“, ale spôsobí zhoršenie presnosti modelu emócie „obava“. Pre tento prípad je každý eliminovaný parameter skontrolovaný pomocou otestovania výsledného modelu a stanovovania výslednej presnosti celého systému. Eliminované slovo (parameter) je ponechané v databáze stop slov len v prípade, ak je presnosť zvýšená o stanovenú hranicu.

Hranica rozdielu presností (pred a po sekvenčnej eliminácií) pre elimináciu parametru je postupne znižovaná. Aby bolo slovo eliminované hneď v prvom cykle, je potrebné aby malo čo najväčší kladný dopad na výslednú úspešnosť celého systému.

Zoskupovanie slov

Po aplikovaní všetkých krokov predspracovania vstupného textu (tokenizácia, kontrola pravopisu, lemmatizácia, nahradenie synoným, odstránenie stop slov) je aplikovaná metóda zoskupovania slov. Metóda používa definovaný slovník pre každú skupinu, teda slovníky sú jazykovo závislou časťou a musia byť manuálne definované pre daný jazyk. Skupina je definovaná pomocou niekoľkých slov, ktoré budú nahradené identifikačným tokenom skupiny. Napr. každé vulgárne slovo bude nahradené tokenom „xvulgarx“ (v skutočnosti sú vulgárne slová rozdelené do 3 skupín podľa miery vulgárnosti). O slovách v skupine je možné povedať, že majú rovnaký význam – aspoň z emočného hľadiska.

Pre každú emočnú triedu je definovaná skupina slov, ktorá túto triedu jednoznačne vystihuje. Napr. v skupine „xsadx“ sú kľúčové slová, ktoré reprezentujú smútok – bohužiaľ, zle, sklamať, zarmútiť, apod. Tu je možno vidieť náväznosť na metódy založené na detekcií kľúčových slov z kap. 1.1.

Metóda znižuje dimenzionalitu vstupu vďaka nahradzovaniu slov s rovnakým emočným nábojom za jeden skupinový identifikátor. Podobný princíp je aplikovaný v metóde nahradzovania synoným, s tým rozdielom, že skupiny sú volené priamo pre konkrétny problém, je potreba definovať dostatočne unikátny identifikátor skupiny, ktorý nebude môcť byť nahraditeľný iným slovom a slová v skupinách nemusia mať rovnaký význam čo sa kontextu týka. Znížením počtu príznakov je možné predísť pretrénovaniu umelej inteligencie. Bolo definovaných 7 skupín – 4 skupiny pre emočné triedy „smútok“, „obava“, „prekvapenie“, „spokojnosť“ a 3 skupiny pre vulgárne slová s rôznom mierou emočného náboja hnevu. Napr. niektoré slová môžu v určitom kontexte znamenať urážku, v inom kontexte môžu reprezentovať

zástupcu zvieracej ríše. Tieto slová je potrebné separovať od ostatných vulgarizmov. Kompletný zoznam vytvorených skupín je v tab. 3.2.

Tab. 3.2: Veľkosti definovaných skupín príznakov

Skupina	**Počet príznakov**	
	Čeština	Angličtina
Smútok	13	43
Spokojnosť	33	88
Prekvapenie	2	74
Obava	9	59
Vulgárne 1	21	0
Vulgárne 2	81	0
Vulgárne 3	45	0

Metóda ako taká je jazykovo nezávislá. Je možné ju aplikovať na akýkoľvek jazyk, avšak je nutné definovať reprezentujúce skupinové príznaky a príznaky danej skupiny. Metóda spôsobuje vyrovnanie váh slov definovaných v skupine. Ak je niektoré slovo použité v trénovacej množine len sporadicky a iné slovo s rovnakým významom a rovnakým emočným nábojom používané často, budú mať vďaka rovnakej skupine rovnakú váhu na vstupe (ako je jasné z ich významu).

Rozširovanie o staticky definované vzorky

V praxi je nevyhnutné, aby bolo možné systém doladiť podla požiadaviek. Častokrát aj pri dosiahnutí vyššej úspešnosti klasifikácie, bol skutočný pocit z klasifikovania určitých fráz rozdielny voči očakávanému výsledku. Pre tieto prípady, ako aj pre ďalšie zvýšenie presnosti, bola navrhnutá metóda rozširovania trénovacej množiny o staticky definované vzorky, kde je trénovacia sada rozšírená frázami s jednoznačne určitým emočným nábojom – vzorkami, ktoré sa v trénovacej ani v testovacej sade nenachádzali. Vytvorená databáza vzoriek použitých na trénovanie a testovanie systému nemôže obsiahnuť všetky možné kombinácie rôznych slov a rôznych emočných nábojov. Aj preto je potrebné zabezpečiť možnosť definovať ďalšie vzorky, ktoré ovplyvnia správanie systému požadovaným smerom.

Frázy určené pre použitie v tejto metóde sú založené na spätnej väzbe užívateľov. Porovnaním emócií z výstupu modelu a reálnej emócie rozpoznanej človekom. Navrhnuté fráze sú skontrolované človekom (zamestnancom zákazníckeho servisu) a následne vložené do „ladiacej databázy“ s označenou emóciou. Ide o vzorky s jednoznačne definovanou emóciou, pričom musí obsahovať iba jednu emóciu. Databáza

vytvorená v tomto kroku je následne používaná pri trénovaní modelu. Množiny fráz sú však spracovávané samostatne – mimo ostatné vzorky, aby bolo zabezpečené, že všetky fráze budú použité pri ďalšom trénovaní. Rovnako môžu byť definované aj vety bez emočného náboja, teda tzv. neutrálne vzorky. Tieto sa použijú pri zostavovaní trénovacích množín len v negatívnej časti.

Príkladom takýchto fráz pre emóciu „smútok“ môžu byť vety: „Mám vysoké dlhy“ alebo „Kvalita vašich produktov je stále nižšia“. V týchto prípadoch ide o krátke vety s jedinou emóciou. S ďalšími možnými variáciami boli zaradené pre rozšírenie trénovacej množiny.

3.2 Klasifikácia pomocou „Big Data“

Navrhnutá metóda klasifikácie emócií je značne jazykovo závislá kvôli predspracovaniu, ktoré pozostáva z častí ako kontrola pravopisu, lemmatizácia, nahradenie synoným alebo odstránenie stop slov. Tieto časti vyžadujú znalosť daného jazyka pre správny návrh ich databáz, slovníkov, definíciu pravidiel apod. V súčasnosti je možné získať tieto slovníky a databázy relatívne jednoducho, keďže sú pre množstvo jazykov voľne dostupné na internete. Kvalita týchto databáz však nie je zaistená a nie je možné povedať, že bude dosiahnutá dostačujúca presnosť u každého jazyka. U niektorých jazykov dokonca určité slovníky ani nemusia existovať. Manuálne vytvorenie takýchto slovníkov je časovo náročné a nasadenie systému na väčší počet jazykov znamenalo nemalú investíciu. Aj z tohto dôvodu bol ďalší výskum smerovaný na abstrahovanie navrhnutej metódy.

Zovšeobecnenie metódy pre automatickú analýzu textových dát je založené na odstránení jazykovo závislých častí predspracovania a nahradenie týchto častí iným inteligentným systémom, ktorý bude schopný pochopiť základné závislosti v textových dátach (napr. negáciu) ale aj komplexné štruktúry (napr. frazeologizmy, ióniu, sarkazmus), určiť váhu určitých slov a slovných spojení, a vo výsledku extrahovať znalosť – emóciu autora textu.

Omnoho komplexnejšiu analýzu je možné vykonať pomocou distribuovaných výpočtov a prístupu Big Data. Zozbieraním veľkého objemu dát je možné dosiahnuť podobných výsledkov bez nutnosti komplexného predspracovania. Rozdelením výpočtového problému na viac výpočtových staníc je možné znížiť celkový čas výpočtu na prijateľnú úroveň. Hlavnou myšlienkou je teda vytvoriť automatický jazykovo nezávislý systém pre klasifikáciu textov, postavený na klasifikátore SVM, ktorý bude distribuovane trénovaný pomocou veľkého objemu dát. Pre zjednodušenie celého problému bude návrh sústredený na klasifikáciu textu do tried pozitívny a negatívny (možno chápať ako jeden emočný model vzhľadom na predošlý experiment). Tento prístup môže byť využitý v rôznych prípadoch, napr. pri klasifikácii leteckých

správ používaných v aeronautike – nazývaných *Poznámka pre letcov – Notice to Airmen* (NOTAM), kde je možné správy klasifikovať na dôležité správy (napr. ranvej je uzavretá) a správy, ktoré môžu byť odfiltrované, keďže pilot musí v predletovej príprave spracovať obrovský počet NOTAM správ.[5]

Tradičné optimalizačné metódy (napr. dopredný výber vstupných parametrov, postupná spätná eliminácia) u Big Data prístupu nie je možné použiť vzhľadom na časovú náročnosť a veľký počet vstupných parametrov. Pre zvýšenie presnosti klasifikácie je potreba optimalizačné metódy navrhnúť vzhľadom na obmedzenia u tohto prístupu. Kvôli časovo náročnému trénovaniu je kontrolovanie všetkých kombinácií vstupných parametrov prakticky nemožné. Preto optimalizácia takéhoto systému musí byť založená na inom princípe.

3.2.1 Popis trénovacích a testovacích dát

Vytvorenie kvalitnej databázy s dostatočným počtom záznamov je základnou požiadavkou pri návrhu metódy založenej na „Big Data“ prístupe. Manuálne vytvoriť databázu so státisícami záznamov je prakticky nemožné. Preto je potrebné databázu vygenerovať automaticky. Najjednoduchšou možnosťou je zozbierať dáta z dostupných webových diskusií a automaticky určiť triedu na základe hodnotenia daného produktu, recenzie danej reštaurácie alebo filmu. V týchto prípadoch je textová reprezentácia naviazaná na číselné hodnotenie priamo autorom textu. Zvyčajne ide o počet hviezdičiek, z ktorých sa následne počíta výsledné hodnotenie produktu. Zozbieraním textov a číselných hodnotení je možné relatívne jednoducho vytvoriť veľké databázy s dostačujúcou kvalitou. Takto vytvorené databázy textov s číselným hodnotením u konkrétneho textu sú tiež k dispozícií pre vedecké účely od spoločností ako sú Google, Yelp alebo Amazon, a sú používané výskumnými tímami na celom svete.

Určenie tried pre klasifikáciu textov prebieha na základe priradeného číselného hodnotenia k danému textu, pričom nie je vhodné použiť celý rozsah. Ak z rozsahu možného hodnotenia potrebujeme extrahovať negatívne texty, budeme svoju pozornosť upriamovať na texty s prideleným nízkym hodnotením. U pozitívneho textu naopak na hodnotenia vysoké. Medzi týmito dvomi množinami textov sú texty, ktorých triedu nie je možné jednoznačne určiť. Koľko hviezdičiek patrí k danému textu recenzie určuje priamo autor textu, takže texty môžu byť často plné rozporov. Pokiaľ ide napr. o text s recenziou filmu s hodnotením troch hviezdičiek z piatich, text bude

[5]Náväznosť na projekt „Pokročilé meteorologické informace pro letectví“ so spoločnosťami Honeywell International s.r.o., Český hydrometeorologický ústav, ABS Jets, a.s. a Řízení letového provozu České republiky. Projekt „TH01010503“ riešený v období 1.1.2015–31.12.2016, financovala „Technologická agentura České republiky“.

obsahovať jak pozitívne, tak negatívne emócie. Pre vytvorenie dostatočne kvalitnej databázy je nutné takéto texty odfiltrovať.

Takýmto spôsobom bola vytvorená databáza s celkovým počtom presahujúcim 2,4 milióna vzoriek so zameraním na 4 rôzne jazyky: český, anglický, nemecký a španielsky. Počty vzoriek pre jednotlivé jazyky však nie je vyrovnaný, hlavne z dôvodu nedostatku dát u niektorých diskusií, z ktorých boli dáta zozbierané. Prehľad počtov je znázornený v Tab. 3.3. V experimente bol opäť použitý princíp vyváženia jednotlivých množín dát, takže zo zozbieraných dát bolo nakoniec použitých len niečo cez 850-tisíc vzoriek. Práca s takýmito počtami dát však predstavuje obrovské nároky na operačnú pamäť.

Tab. 3.3: Veľkosti zozbieraných databáz testovaných jazykov

Jazyk	**Celkový počet vzoriek**	**Počet použitých vzoriek**	
		pozitívny	negatívny
Čeština	675 325	99 222	99 222
Angličtina	1 176 316	235 334	235 334
Nemčina	224 911	50 418	50 418
Španielčina	359 770	40 460	40 460

U každého textového vzorku bol uložený odtlačok dát (textový reťazec s výsledkom hašovacej funkcie MD5). Tento odtlačok sa počíta z normalizovaného textu zbaveného diakritiky, so všetkými písmenami prevedenými na malé, odstránené boli aj špeciálne znaky apod. Odtlačok slúži na zabezpečenie, že v databázy sa nevyskytnú 2 rovnaké vzorky. Slúži však aj k výslednému zoradeniu množín. Ako trénovacia časť bola použitá prvá polovica databázy, teda po zoradení podla odtlačku došlo k pseudonáhodnému premiešaniu dát medzi jednotlivými množinami (na základe hašovacej funkcie). Týmto je zároveň zabezpečené, že toto premiešanie bude u všetkých prevedených experimentov bez zmeny.

U rôznych aplikácií je možné voliť iný prístup ako pracovať s takýmto objemom dát, záleží od dostupného hardvéru, charakteru dát, objemov, ktoré sa snažíme spracovať apod. Každý z prístupov spracovania dát poskytuje určite výhody, ale prináša aj nevýhody. Základné prístupy k spracovaniu veľkých objemov dát je možné rozdeliť do nasledujúcich skupín:

- Práca s dátami priamo v operačnej pamäti je najrýchlejšou variantou, predstavuje však vysoké nároky na hardvér, ktorý značne obmedzuje maximálnu veľkosť množiny. Ide však o najrýchlejší prístup čo sa týka priepustnosti dát, ktorá u súčasných operačných pamätí dosahuje desiatky GB/s. Implementačne ide o najjednoduchšiu variantu.

- Dávková práca s dátami poskytuje možnosť spracovať väčšie množstvo dát, než je kapacita operačnej pamäte, keďže v pamäti sa v jednu dobu nachádza len časť dát (jedna dávka), ktorej veľkosť záleží od dostupnej operačnej pamäte. U tohto prístupu je najpomalšou časťou prístup k perzistentnému úložisku, kde sa nachádzajú všetky dáta. Implementačne ide o náročnejšie riešenie. Nevyžaduje však žiaden ďalší hardvér.
- Databázové systémy pracujú s dátami na základe vytvorených indexov a tak nám poskytujú rýchlejší prístup k určitým dátam. Je potreba vhodne navrhnúť štruktúru a indexy pre tieto dáta, inak môže byť systém značne pomalý, čo predstavuje určitý limit. Niektoré systémy však poskytujú možnosť rozloženia databázy na viacero jednotiek na základe definovaného kľúča. Takto je možné uskladniť obrovské objemy dát (stovky TB) bez straty rýchlosti prístupu k týmto dátam.
- Distribuované systémy na spracovanie dát poskytujú možnosť pracovať s celým objemom dát zároveň. Záťaž je rozdelená medzi viaceré výpočtové jednotky, kde každá jednotka pracuje len so svojou časťou dát. Tento prístup je najuniverzálnejší a so zväčšujúcim sa počtom výpočtových jednotiek sa limity takéhoto systému posúvajú.

3.2.2 Popis predspracovania dát

Predspracovanie dát je v porovnaní s predošlým experimentom omnoho jednoduchšie – pozostáva len z tokenizácie (rozdelenie vstupného textu na slová). Všetky jazykovo závislé časti predspracovania boli odstránené, čím bola jazyková závislosť výsledného riešenia minimalizovaná. Každý token (slovo) je vstupným parametrom SVM klasifikátoru, kde hodnota je reprezentovaná pomocou reálneho čísla vypočítaného metódou TF–IDF. Slová na vstupe klasifikátoru sú v tvare, v ktorom boli použité vo vstupnom texte. Teda pre určenie základného tvaru slova nebol aplikovaný žiaden stemming alebo lemmatizácia, tvar slova sa vôbec nemení. Vďaka tomuto prístupu je možné navrhnutý systém aplikovať na iné jazyky omnoho jednoduchšie – bez úpravy implementácie ktorejkoľvek časti, bez definovania slovníkov daného jazyka, vytvárania databáz slov, či definícií pravidiel daného jazyka. Zjednodušením predspracovania sa stavový priestor značne zväčšil, keďže každý tvar slova predstavuje nový parameter. Tento problém je potreba riešiť dodatočne, pomocou optimalizačných metód.

Vstupné parametry boli obohatené o vytvorené n-gramy (metóda predstavená v roku 1992 v článku [7], prípadne [12]), ktoré do systému vnášajú informácie o zastúpení slovných spojení v jednotlivých triedach, pričom tieto spojenia sa vytvárajú v rámci kontextu danej vety. Parametre n-gramov značne zvyšujú dimenzionalitu

vstupu. Napr. ako znázorňuje Tab. 3.4 pre český jazyk, počet vstupných parametrov bol zvýšený zo 164 tisíc na 2,1 milónu použitím 2-gramov a na 6 miliónov pri použití 3-gramov.[6] Táto metóda spôsobuje značné zvýšenie pamäťových nárokov pri použití na veľkom objeme dát. Z tohto dôvodu bola implementovaná *Minimálna frekvencia v dokumentoch – Minimum Document Frequency* (MDF). Určením hodnoty MDF je možné odfiltrovať parametre (v tomto prípade n-gramy), ktoré sa v databáze textov nevyskytli aspoň v danom počte (skúmané boli hodnoty 12 až 32, na databáze 198 tisíc vzoriek). Správnou konfiguráciou MDF je možné tiež odstrániť slová s preklepmi, citoslovcia, dlhé slová bez významu a podobné anomálie, ktoré vznikajú v textoch internetových diskusií. Ako zobrazuje Tab. 3.5, vďaka tejto metóde je možné vygenerovaný počet parametrov (s použitím n-gramov na Big Data probléme) dostaň na vhodnú a zvládnuteľnú úroveň – desiatky až stovky tisíc parametrov aj u vyššej hodnoty n.

Tab. 3.4: Počet príznakov v závislostí na úrovni n-gramu pre český jazyk

n	**Počet unikátnych príznakov**	
	pozitívnych	negatívnych
1	164 443	164 349
2	2 109 984	2 105 152
3	6 049 007	6 031 760

3.2.3 Popis experimentu

Navrhnuté riešenie vzhľadom na charakter dát a charakter riešeného problému (žiadne predspracovanie, implementovať optimalizácie vzhľadom na veľké objemy dát) bolo postavené na distribuovaných výpočtoch. Časovo zložitá výpočtová úloha je rozdelená na viacero výpočtových jednotiek. Vďaka tomuto prístupu je možné trénovať klasifikátor pomocou veľkých objemov dát (Big Data) v horizonte pár minút a tak validovať väčšie množstvo možných konfigurácií. Veľký objem dát je tiež rozdelený medzi výpočtové jednotky, kde každá jednotka pracuje len s určitou podmnožinou dát.

Z predošlých experimentov (publikovaných v [82], ale aj staršie [9] a [10]) sa javí, že klasifikátor SVM je z tradičných metód najvhodnejším na textovú analýzu. Predpokladom do tohto experimentu je, že pokiaľ bude dostatok dát, budú pokryté všetky možnosti ktoré nastávajú pri texte (s hľadaným emočným nábojom) v danom

[6]Pre vyššie hodnoty n nebolo možné zistiť počty parametrov z dôvodu vysokej pamäťovej náročnosti, šlo by však o desiatky miliónov parametrov.

Tab. 3.5: Počet príznakov v závislostí na MDF

MDF	**Úroveň n-gramu**					
	2		3		4	
	poz.	neg.	poz.	neg.	poz.	neg.
Čeština						
12	67 250	67 056	77 958	77 703	79 840	79 568
20	40 121	39 975	45 051	44 886	45 804	45 635
24	33 418	33 277	37 199	37 042	37 745	37 581
28	28 596	28 498	31 563	31 450	31 963	31 851
32	25 092	24 980	27 511	27 397	27 814	27 692
Angličtina						
12	385 489	385 491	760 797	760 800	944 965	944 965
20	255 279	255 279	467 970	467 962	560 399	560 387
24	220 626	220 626	393 922	393 918	466 153	466 149
28	194 971	194 973	340 736	340 741	399 336	399 343
32	175 312	175 310	300 817	300 809	349 626	349 614
Nemčina						
12	39 872	39 872	49 827	49 827	51 450	51 450
20	24 404	24 404	28 899	28 899	29 533	29 533
24	20 515	20 515	23 896	23 896	24 336	24 336
28	17 622	17 622	20 237	20 237	20 546	20 546
32	15 507	15 507	17 637	17 637	17 864	17 864
Španielčina						
12	24 758	24 758	33 571	33 571	35 651	35 651
20	15 556	15 556	19 831	19 831	20 668	20 668
24	13 170	13 170	16 519	16 519	17 122	17 122
28	11 409	11 409	14 082	14 082	14 556	14 556
32	10 052	10 052	12 273	12 273	12 654	12 654

jazyku. S využitím veľkých objemov (Big Data) by mala byť táto podmienka splnená a takéto riešenie bude dostatočne univerzálne a jazykovo nezávislé. Tradičný klasifikátor SVM však nie je možné v prípade distribuovaného trénovania použiť. Vhodným riešením je použitie distribuovanej varianty tohto klasifikátoru, kde implementácií je v súčasnej dobe dostupných veľa. Vhodnou môže byť napr. kaskádové SVM. [98]

Kaskádové SVM predstavené v článku [32] využíva rozdelenie trénovacej množiny na podmnožiny a rozdelenie trénovacej procedúry hierarchicky pomocou binárneho

stromu. Podmnožiny sú distribuované k listom stromu, kde prebieha trénovanie čiastkových častí klasifikátoru. V rodičovských uzloch stromu sú po natrénovaní podporné vektory zlúčené. Ako popisuje [98], aj pri zložitých problémoch môže byť čas trénovania redukovaný z celkovej doby jedného dňa na jednu hodinu vďaka použitiu distribuovaného prístupu.

Vzhľadom na odstránenie mnohých krokov predspracovania vstupných dát, na dosiahnutie vyššej úspešnosti je potreba ďalej zvoliť vhodnú optimalizačnú metódu. Tradičné optimalizačné metódy (ako dopredný výber vstupných parametrov, postupná spätná eliminácia) nie je možné použiť kvôli vysokému počtu vstupných parametrov a časovej náročnosti trénovania (narastá s veľkosťou trénovacej sady). Samotný beh trénovania a optimalizácie takéhoto systému prostredníctvom napr. postupnej spätnej eliminácie by zabral niekoľko desiatok rokov. Je preto potreba navrhnúť sofistikovanejšiu metódu na voľbu vhodných parametrov. Vhodnou cestou je prehľadanie stavového priestoru možných kombinácií príznakov pomocou genetického programovania.

3.2.4 Optimalizácia genetickým programovaním

V posledných rokoch boli algoritmy genetického programovania použité na riešenie rôznych komplexných problémov – napr. problém plánovania práce [86], využívajú sa v evolučnej robotike alebo bezpečnosti. U týchto problémov nie je možné vyskúšať všetky možné kombinácie pre nájdenie najvhodnejšieho riešenia, preto spadajú do triedy „NP“, resp. „NP-hard“ (neriešiteľné v polynomiálnom čase). Ako dokazuje článok [25], *Genetické programovanie* (GP) môže pomôcť v týchto prípadoch relatívne efektívne prehľadávať stavový priestor. Klasifikáciu textu a hľadanie vhodnej konfigurácie, resp. vhodnej sady príznakov, je možné pokladať za optimalizačný problém.

Keďže z dôvodu vysokého počtu príznakov nie je možné použiť doprednú selekciu ani spätnú elimináciu (viac informácií v článkoch [11] a [62]) je nutné pristupovať k problému inak – cestou prehľadávania stavového priestoru možných kombinácií príznakov pomocou genetického programovania. GP je inšpirované biologickou evolúciou zameranou na vytváranie programov (postupov), ktoré sa snažia riešiť danú úlohu čo najlepšie. Kombináciou heuristiky a náhodného prehľadávania stavového priestoru je možné sa dopracovať k zaujímavým výsledkom v relatívne krátkom čase.

Definícia algoritmu GP

Navrhnutá optimalizácia pre problém distribuovaného trénovania veľkými objemami dát využíva chromozómy, ktorých štruktúra pozostáva v našom prípade z množiny vybraných povolených parametrov (selekcia príznakov, resp. slov a slovných spojení,

ktoré môžu byť reprezentované na vstupe klasifikátoru), úrovne n-gramu a hodnoty MDF. Keďže pamäťové nároky rapídne vzrastajú s úrovňou n-gramu, bola definovaná maximálna hodnota pre tento parameter. Naopak, parameter MDF spôsobuje zníženie dimenzionality vstupu a teda aj pamäťových nárokov, takže tento nie je potreba limitovať.

Vygenerovanie prvej populácie bolo prevedené náhodnou voľbou úrovne n-gramu v povolených medziach a konštantnej hodnoty parametru MDF. U každého génu je tiež potreba definovať množinu povolených parametrov, kde bolo vybraných 99,5% náhodných parametrov z celkovej množiny, ktorú je možné pre danú konfiguráciu vygenerovať z trénovacích dát. Toto číslo bolo zistené empirickou metódou, kde práve táto hodnota u génov z inicializačnej populácie dosahovala najvyššiu hodnotu úspešnosti na testovacej množine.

Všeobecné princípy evolučného programovania [3] tvrdia, že každému génu musí byť priradená vhodnosť pomocou ohodnocujúcej (fitness) funkcie. Hodnotenie prebieha na základe štatistického zhodnotenia výsledného klasifikátoru s danou konfiguráciou a využívajúci len povolené parametre. Výsledná hodnota vhodnosti daného génu je reprezentovaná úspešnosťou klasifikácie na testovacej množine.

Navrhnutá optimalizácia využíva turnajovú selekciu, kde u každej generácie je zachovaný najlepší chromozóm. Tento je potom použitý u reprodukčných a mutačných operátorov s najvyššou pravdepodobnosťou.

Navrhnuté operátory GP

Pre zabezpečenie prehľadávania stavového priestoru je potreba aby dochádzalo k zmenám v obsahu chromozómov v danej populácií, teda k zmenám množín povolených parametrov a konfigurácií klasifikácie (pozostáva z parametrov MDF a úrovne n-gramu). Toto bolo dosiahnuté pomocou 6 navrhnutých operátorov: tri z nich pracujú s množinou povolených parametrov, 2 operátory menia konfiguráciu a 1 špeciálny zabezpečuje výmenu celej množiny medzi dvomi chromozómami (pokiaľ je to možné).

Operátory určené na zmenu povolených parametrov (slov) majú prístup ku kompletnej množine parametrov pre určenú konfiguráciu (množina je závislá na úrovni n-gramu), ktorú následne využívajú pri svojom behu. Mutačné operátory zabezpečujú, že GP neuviazne v lokálnom optime. Navrhnuté mutačné operátory tohto typu sú: Mutačný **operátor zmeny počtu povolených parametrov**, ktorý môže zvyšovať alebo znižovať počet povolených parametrov. Za rovnakým účelom bol navrhnutý mutačný **operátor nahradenia časti množiny**, ktorý nahradzuje časť množiny (percentuálne vyjadrenú) náhodne zvolenými parametrami z kompletnej množiny. Tento operátor bol použitý v procese dva krát – prvý nahradzuje 1% a druhý 0,01% z množiny povolených parametrov. Reprodukčný **operátor výmeny**

časti množiny povolených parametrov medzi sebou využívajú princíp vytvorenia nového chromozómu s časťami množín dvoch rôznych chromozómov v danej generácií. Takže nový skrížený chromozóm bude mať časť množiny povolených parametrov z prvého chromozómu a časť z druhého chromozómu. Touto operáciou konfigurácia klasifikácie ostáva nezmenená, opäť sa pracuje len s množinami parametrov.

Zmena konfigurácie klasifikácie a tak značné zväčšenie prehľadávaného stavového priestoru bola zabezpečená pomocou dvoch mutačných operátorov (mutujúcich dva možné parametre). Pre zmenu parametru MDF bol navrhnutý **operátor zmeny hodnoty MDF**, ktorý mení túto hodnotu v nastavených medziach (1 až 32). Zmeny druhého konfiguračného parametru má na starosti **operátor zmeny úrovne n-gramu**, ktorý takisto pracuje v nastavených medziach (2 až 3). Tento operátor má však vplyv na celkovú množinu povolených parametrov (slov), takže takouto mutáciou vznikajú takmer úplne nové chromozómy – s minimálnou podobnosťou k pôvodnému mutovanému chromozómu. Pokiaľ dochádza k zníženiu úrovne, je vygenerovaná úplne nová množina. Je to z dôvodu, že filtrovanie parametrov by bolo zbytočne zložité a časovo náročné. Hlavnou úlohou tohto operátora je vniesť do populácie čo najviac nových parametrov, ktoré sa v populácií pred tým nenachádzali.

Posledným navrhnutým reprodukčným operátorom je **operátor výmeny množín**. Pri aplikovaní dochádza k vzniku dvoch nových chromozómov, kde nové chromozómy majú len vymenené množiny povolených parametrov (slov) medzi sebou. Táto výmena môže nastať len u operátorov s rovnakou úrovňou n-gramu.

Tab. 3.6 zachytáva navrhnuté operátory, početnosť ich použitia v procese generovania novej populácie prostredníctvom pravdepodobnosti ich použitia a prípadné upresnenie ich správania.

3.3 Hlboké učenie pre klasifikáciu textu

Znalosť jazyka a gramatiky je základným predpokladom k pochopeniu textu a následnému extrahovaniu znalosti. Súčasné „state-of-the-art“ systémy využívajú modely zhlukovania slov, viet alebo dokumentov (v angličtine označované ako „embedding models“, viac informácií v [52]). Modely zhlukovania slov prevádzajú každé slovo z textu na vektor, ktorý predstavuje reprezentáciu daného slova s možnými významami, vzťahmi k iným slovám, kontextom apod. Takýto vektor je vytvorený s použitím veľkého objemu vzoriek, v ktorých sa toto slovo vyskytovalo. Model je teda zameraný vždy na konkrétny jazyk a predstavuje jazykovo závislú časť systému.

Vytvoriť model zhlukovania slov pre každý jazyk predstavuje úlohu náročnú na dáta a výpočtové zdroje, navyše použitie takéhoto modelu v procese klasifikácie

Tab. 3.6: Navrhnuté operátory GP, ich početnosť a konfigurácia

Operátor	**Typ**	**P.**	**Upresnenie**
výmena množín	reprodukcia	50%	
výmena časti množiny	reprodukcia	50%	
nahradenie časti množiny	mutácia	5%	nahr. 1% parametrov
nahradenie časti množiny	mutácia	5%	nahr. 0,01% parametrov
zmena počtu povolených parametrov	mutácia	1%	pridaj 1 parameter
zmena počtu povolených parametrov	mutácia	1%	pridaj 2 parametre
zmena počtu povolených parametrov	mutácia	1%	pridaj 5 parametrov
zmena počtu povolených parametrov	mutácia	1%	odober 1 parameter
zmena počtu povolených parametrov	mutácia	1%	odober 2 parametre
zmena počtu povolených parametrov	mutácia	1%	odober 5 parametrov
zmena úrovne n-gramu	mutácia	20%	
zmena hodnoty MDF	mutácia	20%	

textu predstavuje obrovské nároky na operačnú pamäť (stovky MB). Tieto modely musia byť k dispozícií (vopred pripravené alebo použitý online dostupný model) pred analýzou daného textu. Keďže ide o ďalší model, ktorý je potreba natrénovať, vnášajú do systému určitú chybu, keďže samé vykazujú určitú presnosť. Navyše prinášajú riziko, že niektorá z potrebných informácií pre extrakciu požadovanej znalosti bude už na vstupe odstránená.

Kognitívne správanie človeka pri čítaní textu sa však líši od súčasných metód automatického spracovania textu. Zvyčajne sú slová prevádzané na vektor príznakov a následne algoritmy strojového učenia extrahujú požadovanú znalosť. Prekážkou v tomto procese je nutnosť znalosti daného jazyka – či už vo forme komplexného predspracovania alebo vo forme modelu zhlukovania slov. Bez znalosti gramatiky a syntaxe daného jazyka nie je možné dosiahnuť pomocou súčasných systémov úspešnosť približujúcu sa človeku, a teda ani úplne jazykovo nezávislé riešenie.

Navrhnutá metóda postavená na hlbokom učení predstavuje alternatívu k súčasným metódam založených na modeloch zhlukovania. Metóda sa približuje k procesu, ktorým dochádza u človeka k porozumeniu textových dát – zo surových dát je pochopená typografia daného jazyka, sú detekované začiatky a konce slov (ak je to možné u daného jazyka), písmená a typografické príznaky sú formované do slabík a slov, slová do slovných spojení, následne do viet, z ktorých je potom pochopená hľadaná znalosť. Navrhnutá metóda teda simuluje celý proces čítania a pochopenia textu človekom. Za týmto účelom bola navrhnutá štruktúra hlbokej neurónovej siete, ktorá využíva niekoľko konvolučných vrstiev a komplexných štruktúr.

3.3.1 Popis trénovacích a testovacích dát

Pre overenie navrhnutej metódy bolo použitých 7 rôznych databáz textov. Prvých 5 databáz predstavujú dáta z predošlých experimentov – privátne množiny textových dát zhromaždené z rôznych webových zdrojov, pre 5 rôznych jazykov: angličtina, nemčina, čeština, španielčina a čínština. Keďže tieto množiny boli vytvorené automaticky na základe hodnotenia užívateľa, obsahujú často rôzne preklepy, sarkazmus, iróniu, v niektorých prípadoch hodnotenie priradené textu neodpovedá skutočnému emočnému náboju v texte (nízke hodnotenie ale pozitívny popis) apod. Ako popisuje článok [1], ide o časté problémy, ktoré sa v textových databázach vyskytujú a vnášajú určitú náročnosť do problému klasifikácie textu.

Každému textu bola priradená trieda – pozitívny alebo negatívny náboj. Pre zaistenie kvality databázy bol použitý rovnaký princíp ako pri predošlom návrhu systému, kde boli použité vzorky s najvyšším hodnotením (vrchných 20%) a s najnižším hodnotením (spodných 20%). Zozbierané databázy slúžia na demonštráciu funkčnosti navrhnutej metodológie a jej nezávislosť na jazyku vstupného textu. Zozbieraná databáza obsahuje aj texty v čínskom jazyku, ktorý má diametrálne odlišnú typografiu a syntax voči predošlým testovaným jazykom. Sumarizácia zozbieranej databázy textov a jej jednotlivých množín je v Tab. 3.7.

Tab. 3.7: Veľkosti zozbieraných databáz testovaných jazykov pre hlboké učenie

Jazyk	**Celkový počet vzoriek**	**Počet použitých vzoriek**	
		pozitívny	negatívny
Čeština	675 325	99 222	99 222
Angličtina	1 176 316	235 334	235 334
Nemčina	224 911	50 418	50 418
Španielčina	359 770	40 460	40 460
Čínština	1 571 156	30 305	30 305

Ako zobrazuje Tab. 3.7, iba časť zozbieraných vzoriek bola použitá. U čínskeho jazyka je tento rozdiel najviac viditeľný – zozbieraný počet negatívnych vzoriek bol 30 305, čo predstavuje zároveň maximálny počet pozitívnych vzoriek, ktoré môžu byť použité, aby boli množiny vyrovnané. Narozdiel od predošlých experimentov, databáza bola v tomto prípade rozdelená na 3 časti – na trénovaciu množinu (prvých 40% dát), validačnú množinu (nasledujúcich 30%) a testovaciu množinu (posledných 30%).

Pre možnosť objektívneho porovnania so súčasnými metódami bola navrhnutá metóda overená na verejne dostupných databázach:

- **Databáza hodnotení Yelp**[7] získaná z desiateho kola súťaže „Yelp challenge“ 1. Septembra 2017. Táto databáza obsahuje viac ako 4,7 milióna textových hodnotení označených počtom hviezdičiek 1 až 5. Avšak táto databáza nie je vyvážená a obsahuje duplicity. Tieto problémy boli odstránené pred použitím databázy. Výsledné veľkosti množín sú znázornené v Tab. 3.8.
- **Databáza hodnotení Amazon**[8] bola vytvorená projektom „Stanford Network Analysis Project (SNAP)“ zozbieraním hodnotení a užívateľských diskusií z webu Amazon.com do júla 2014, ako popisuje článok [66]. Táto databáza obsahuje 82 miliónov záznamov. Podobne ako databáza Yelp, obsahuje číselné hodnotenie v rozmedzí 1 až 5.

Tab. 3.8: Veľkosti testovaných verejne dostupných databáz

Databáza	**Yelp**	**Amazon**
Celkový počet	4 736 897	82 456 877
1 hviezdička	639 849	6 702 809
5 hviezdičiek	1 988 003	49 006 613
Použité vzorky	**1 277 462**	**13 405 618**

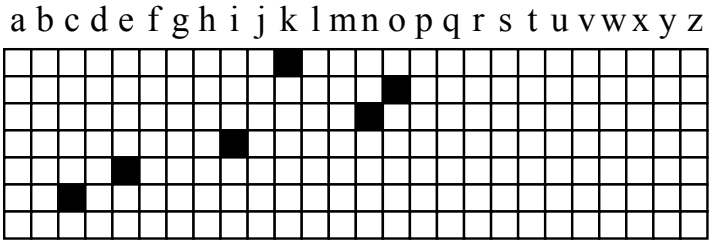
Ako znázorňuje Tab. 3.8, opäť ide o „Big Data“ problém. Navrhnutý systém však využíva masívny paralelizmus pomocou grafických kariet a dávkové spracovanie dát. Kvôli limitovanej pamäti súčasných grafických kariet, metóda bola navrhnutá pre využitie u kratších textov (limitované na dĺžku 512 znakov) ako sú krátke príspevky v diskusií, generované strojové texty (logy), titulky novín a časopisov, správy zo sociálnej siete Twitter apod.

3.3.2 Prevod vstupných dát

Prvým krokom v analýze textu pomocou hlbokého učenia je reprezentácia textových dát prostredníctvom číselnej matice. Čítaním textu písmeno za písmenom a zakódovanie každého znaku do vektoru je inšpirované prácou [118], kde bola prvý krát predstavená jednoduchá konverzia textového vstupu. Proces je postavený na definícií abecedy a kódovania „one-hot“.[9] Tento prístup vytvára maticu jednotiek a núl, ktorá môže byť považovaná za surovú reprezentáciu textových dát na vstupe. Toto kódovanie však môže byť optimalizované vypustením bielych (napr. medzery) a špeciálnych znakov, a teda ich zakódovaním pomocou nulového vektoru. Príklad

[7] Verejne dostupná na vedecké účely na adrese: https://www.yelp.com/dataset
[8] Dostupná na vedecké účely na adrese: http://jmcauley.ucsd.edu/data/amazon/
[9] Toto kódovanie bolo popísané už v roku 1954 v článku [39].

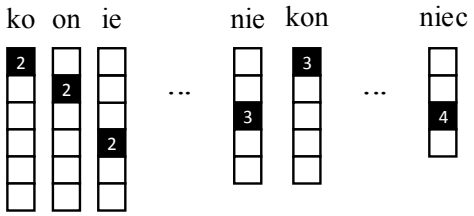
zakódovaného textu je znázornený na Obr. 3.6. Tento prístup je kompletne jazykovo nezávislý – jedinou jazykovo závislou časťou je abeceda, ktorá musí byť na začiatku definovaná. Táto informácia sa však dá relatívne jednoducho extrahovať z trénovacej množiny.



Obr. 3.6: Slovo „koniec“ reprezentované pomocou matice na vstupe

Obr. 3.6 zobrazuje slovo „koniec“ v surovom stave na vstupe hlbokej neurónovej siete. Prvý riadok reprezentuje písmeno „k“, druhý písmeno „o“ apod. Posledný riadok reprezentuje prázdny vektor, ktorý je použitý v prípade už spomínaných bielych znakov, špeciálnych znakov, znakov ktoré neboli v abecede definované a v prípade, že text nie je dostatočne dlhý sú ostatné riadky matice prázdne.

Takto vytvorená matica je následne privedená na vstup hlbokej neurónovej siete. Prvé vrstvy navrhnutej siete transformujú dáta do dimenzie kanálov, kde dochádza k detekcií slabík, krátkych slov, začiatkov a koncov slov, ako zobrazuje Obr. 3.7. Vďaka tomuto prístupu má neurónová sieť priamo kontakt so surovými dátami a teda s omnoho väčšou koncentráciou informácií, než pri použití modelu zhlukovania slov, kde tieto modely výrazne redukujú veľkosť reprezentácie vstupu (prostredníctvom pár reálnych čísel).



Obr. 3.7: Extrakcia prostredníctvom prvých konvolučných vrstiev

Konvolučné jadrá prvých konvolučných vrstiev môžu predstavovať napr. slabiky. Pomocou konvolúcie daného jadra a vstupných dát sú detekované podobnosti slabík

(krátkych slov, začiatkov a koncov slov apod.), ktoré sú následne reprezentované prostredníctvom aktivácie v danom kanáli, prípadne miery aktivácie.

Keďže text na vstupe je reprezentovaný pomocou dvojrozmerných dát (podobný princíp ako u obrázkov), spracovanie týchto dát prebieha podobne ako u obrazových dát pri obrazovej analýze alebo klasifikácií obrazu. Ako popisuje [18], pri trénovaní pomocou obrazových dát sa využívajú konkrétne hodnoty šedo-tónového kanálu, prípadne hodnoty jednotlivých farebných zložiek (červenej, zelenej a modrej) v matici pixelov. Pri 2D reprezentácií textových dát ide o podobný princíp ako u šedotónového obrazu. Sieť je schopná sa naučiť zo surových textových dát rozpoznať opakujúce sa vzory z danej textovej množiny, pochopiť štruktúru a špecifiká daného jazyka, rozpoznávať jednotlivé slová, prípadne detekovať podobnosť slov v prípade preklepu. Takýto postup sa približuje tomu, ako človek číta text a ľudský mozog chápe štruktúru a syntax daného jazyka.

3.3.3 Návrh RCK jadra

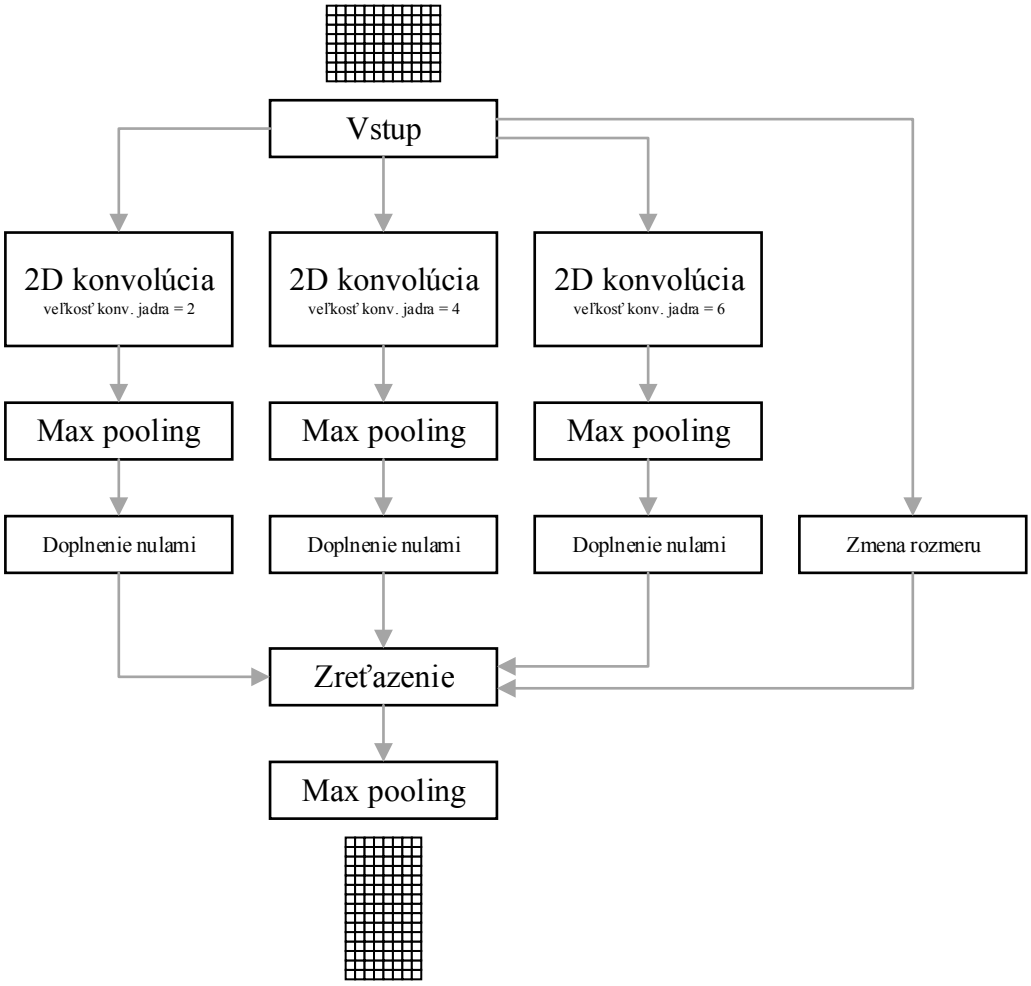
Pri návrhu štruktúry hlbokej neurónovej siete sa vychádzalo z už existujúcich štruktúr používaných pri obrazovom spracovaní. Ide prevažne o siete VGG (článok [18]), Inception (článok [103]), viac-stĺpcovú architektúru popísanú v článku [17] a reziduálne učenie popísané v článku [35]. V navrhnutej štruktúre hlbokej neurónovej siete boli identifikované opakujúce sa štruktúry, ktoré boli vyextrahované a označené ako *Opakujúce sa jadro – Recurring Kernel* (RCK), teda opakujúce sa jadrá, ktoré nesú rovnaké informácie rôznymi vetvami siete do hlbších vrstiev siete, čo umožňuje neurónovej sieti pohľad na danú vetu prostredníctvom rôznej miery abstrakcie.[10]

Toto jadro je navrhnuté tak, aby získalo čo najviac informácií dostupných v danej hĺbke siete. Zapojením niekoľkých jadier za sebou je možné vytvoriť hlbokú architektúru, ktorá zároveň zabezpečuje vysokú informačnú priepustnosť – neobmedzený tok informácií celou štruktúrou siete. Dokonca najhlbšie vrstvy môžu mať prístup k dátam z prvých vrstiev. Takéto siete môžu byť trénované priamo pomocou algoritmu SGD [6], prípadne iných optimizačných techník. Všetky dáta a parciálne informácie sú nesené do najnižších vrstiev, kde pomocou plne prepojených vrstiev môže dôjsť ku klasifikácií.

Použitie konvolučných vrstiev vytvára v jednotlivých úrovniach siete „odtlačky“ informácií v každom stĺpci tejto štruktúry. Tieto dátové odtlačky sú zozbierané a rozšírené o reziduálnu časť – o vstup daného jadra. Všetky výstupy sú považované za samostatné dátové kanály. Keďže ide o pamäťovo náročnú operáciu, pred výstupom dát z jadra dôjde k ich podvzorkovaniu. Túto operáciu je možné vykonať vďaka

[10]Idea „recurring“ jadier je v čase písania tejto práce v štádiu recenzného riadenia k publikácií v časopise Cognitive Computation (IF pre rok 2016 = 3,441).

tomu, že niektoré dáta sú vďaka vytvoreným odtlačkom duplikované. Štruktúra RCK jadra je znázornená na Obr. 3.8.



Obr. 3.8: Štruktúra RCK jadra

Na vstupe navrhnutej siete je text reprezentovaný ako matica jednotiek a núl, čo môže byť chápané aj ako čierno-biely obrázok (s jedným kanálom). Vstup je transformovaný pomocou RCK jadra na podvzorkovanú maticu pozostávajúcu z podvzorkovaných vstupných dát a parciálnych informácií extrahovaných konvolučnými vrstvami. Opakovaním tohto procesu niekoľkokrát za sebou dôjde k extrakcií požadovanej znalosti. Príklad takejto siete je načrtnutý v Tab. 3.9. Vďaka kapacite jednotlivých jadier a ich informačnej priepustnosti, takto navrhnutá sieť je schopná pochopiť daný jazyk – jeho syntax, štruktúru viet, často používané slovné spojenia v hľadaných prípadoch, detekovať potrebné informácie ako je napr. negácia, ale aj zložitejšie štruktúry, a vyviesť vhodný záver.

Tab. 3.9: Príklad siete pozostávajúcej z RCK jadier, sieť č. 2

Vrstva	**Parametre**	**Rozmer výstupu**
Vstup		(53, 512, 1)
RCK jadro	64 filtrov	(1, 256, 245)
RCK jadro	64 filtrov	(1, 128, 437)
RCK jadro	64 filtrov	(1, 64, 629)
RCK jadro	32 filtrov	(1, 32, 725)
Sploštenie		(23200)
Plne prepojená	256 neurónov, relu	(256)
Plne prepojená	128 neurónov, relu	(128)
Plne prepojená	2 neuróny, softmax	(2)

4 Overenie navrhnutých metód

Funkčnosť navrhnutých metód bola overená na vybraných príkladoch. Optimalizačné metódy boli overené na probléme klasifikácie 5 emočných tried z textových dát, kde boli aplikované na tradičný prístup klasifikácie – relatívne malé množstvo dát, použitie jednoduchých klasifikátorov postavených na metóde SVM, komplexné predspracovanie vstupných dát. Abstrahovaná metóda založená na Big Data prístupe bola overená na probléme valencie textu (klasifikácia do dvoch tried), ako aj optimalizačných metód vhodných pre tento prístup. Metóda postavená na hlbokom učení s novo navrhnutým jadrom pre účely analýzy textových dát bola overená na voľne dostupných databázach prístupných vedeckej verejnosti.

Výpočtové prostredie pre všetky experimenty popísané v tejto kapitole pozostávalo z osobného počítača s procesorom Intel Core i7–2600K s frekvenciou 3,4GHz, 32GB operačnej pamäte, a grafickým akcelerátorom GeForce 1080Ti s 12GB operačnej pamäte. Pri experimente s Big Data prístupom bola využitá učebňa s 24 výpočtovými jednotkami pre distribuované výpočty – rôzne konfigurácie, prevažne však Intel Core i5 a 8GB operačnej pamäte.

4.1 Tradičné metódy a ich optimalizácie

Prvým krokom pri návrhu metódy klasifikácie emócií vola voľba vhodného existujúceho klasifikátoru. Experimenty prebiehali na klasifikácií dvoch tried pre porovnanie rôznych klasifikátorov. Úspešnosť klasifikácie bola relatívne vysoká pre klasifikátor SVM – s hodnotou 87,00 % ako naznačuje tab. 4.1. Experiment prebiehal na databáze so 7000 vzorkami (5000 vzoriek určených pre trénovanie, 2000 vzoriek pre testovanie). Databáza bola vytvorená automaticky z textov hodnotenia produktov a komentárov s hodnotením, na základe ktorého bolo rozhodnuté, či ide o pozitívny alebo negatívny text.

Na základe týchto výsledkov, ale aj výsledkov popísaných v kap. 1.1, bol pre ďalšie experimenty zvolený práve klasifikátor SVM. Bohužiaľ, rozlišovanie valencie textu (pozitivity a negativity) sa ukázalo ako nepostačujúce pre systém podpory zákazníkov. Z tohto dôvodu bola navrhnutá metóda klasifikácie emócií pomocou viacerých modelov a následne metódy optimalizácie. Základnou myšlienkou bolo natrénovanie jednoduchého modelu pre každú jednu emočnú triedu tak, že každý bude rozhodovať či vstupný text patrí do danej triedy alebo nie, resp. s akou pravdepodobnosťou.

Pre overenie navrhnutej metódy klasifikácie piatich emócií bola využitá druhá polovica databázy – 1673 vzoriek pre český jazyk, 279 vzoriek pre anglický jazyk. Množiny pre jednotlivé modely boli vyvážené, teda negatívna množina bola zvolená

Tab. 4.1: Porovnanie tradičných metód na klasifikáciu textových dátach

	Skutočný neg.	Skutočný pos.	Predikcia triedy
FLM klasifikátor			
Predikovaný neg.	799	133	85.73%
Predikovaný pos.	201	867	81.18%
Vyťaženosť triedy	79.90%	86.70%	
Random Forest klasifikátor			
Predikovaný neg.	663	696	48.79%
Predikovaný pos.	337	304	47.43%
Vyťaženosť triedy	66.30%	30.40%	
***k*-NN klasifikátor**			
Predikovaný neg.	993	976	50.43%
Predikovaný pos.	7	24	77.42%
Vyťaženosť triedy	99.30%	2.40%	
SVM klasifikátor			
Predikovaný neg.	867	127	87.22%
Predikovaný pos.	113	873	86.78%
Vyťaženosť triedy	86.70%	87.30%	
		FLM presnosť	**83.30%**
		Random Forest presnosť	**48.35%**
		***k*-NN presnosť**	**50.85%**
		SVM presnosť	**87.00%**

tak, aby jej veľkosť bola rovná pozitívnej množine. Zároveň bolo dodržané rovnomerné zastúpenie textov ostatných emócií, ako popisuje kap. 3.1.3. Proces evaluácie má paralelný beh. Výsledná presnosť je počítaná z originálnej značky modelu a predikcie bez akéhokoľvek zakázaného pásma, pričom sa rozlišuje len pozitívny a negatívny výsledok. Niektoré metódy boli testované prevažne len pre český jazyk z dôvodu časovej náročnosti tejto operácie. Všetky uvedené výsledky boli prijaté vedeckou komunitou a publikované v časopise RadioEngineering (IF pre rok 2016 = 0,945) [87] a na medzinárodných konferenciách – články [82] a [83], prípadne v článku [85] v slovenskom jazyku.

Tab. 4.2 zobrazuje počiatočnú úspešnosť navrhnutého riešenia pred aplikovaním optimalizačných metód. Novo navrhnuté optimalizačné metódy a ich vplyv na úspešnosť klasifikácie budú demonštrované tak, ako boli aplikované v praxy – všetky implementované v jednom systéme. Výsledky jednotlivých metód sú následne znázornené na častiach procesu trénovania, kde sa ich vplyv prejavil. Celkový dopad

Tab. 4.2: Presnosť klasifikácie pred aplikovaním optimalizačných metód

Presnosť modelu emócie					**Presnosť**
obava	smútok	spokojnosť	hnev	prekvapenie	**klasifikácie**
Český jazyk					
91,11%	73,03%	71,54%	70,00%	71,74%	**75,49%**
Anglický jazyk					
79,81%	75,44%	56,73%	69,09%	76,85%	**71,58%**

optimalizácií predstavuje zvýšenie úspešnosti klasifikácie o 11,40 % pre český jazyk, ktorý je zhodnotený v záverečnej metóde – v metóde rozširovania trénovacej množiny o staticky definované vzorky.

4.1.1 Optimalizácia sekvenčnou elimináciou parametrov

Niektoré kroky sekvenčnej eliminácie pre český jazyk sú zachytené v Tab. 4.3, kde v poslednom stĺpci je uvedená presnosť v danom kroku. Tabuľka popisuje správanie modelov pri jednotlivých elimináciách parametru a ich prípadné zapísanie do databázy eliminovaných parametrov. V tabuľke je táto akcia zobrazená pomocou zaškrtnutia pri danom parametre.

Jednotlivé eliminované parametre ovplyvňujú úspešnosť klasifikácie viacerých modelov, keďže je medzi jednotlivými modelmi vzťah[1] – spoločné predspracovanie vstupných dát. Eliminácia jedného parametru môže zvýšiť presnosť daného emočného modelu, ale zároveň znížiť presnosť iného emočného modelu. Toto správanie je vidno napr. u 3. kroku, kde došlo k zvýšeniu presnosti u modelov emócií hnev a spokojnosť, ale k zníženiu úspešnosti klasifikácie emócie obava. V 5. kroku (teda po eliminovaní ďalších dvoch parametrov) sa však tento emočný model dostal späť na svoju pôvodnú hodnotu.

Na demonštráciu funkčnosti bola vybraná jedna sledovaná eliminácia s dostatočnou dĺžkou a s krokmi, ktoré ukážkovo popisujú možné situácie. Ide o elimináciu z etapy 4, u ktorej došlo k zvýšeniu presnosti o 3,07% pre český jazyk, pričom v 39. kroku došlo k ukončeniu behu eliminácie. Posledný eliminovaný príznak (slovo „inteligentní“) nepredstavoval pozitívnu zmenu správania systému, preto bol odobraný z množiny eliminovaných príznakov. Výsledná použitá konfigurácia bola

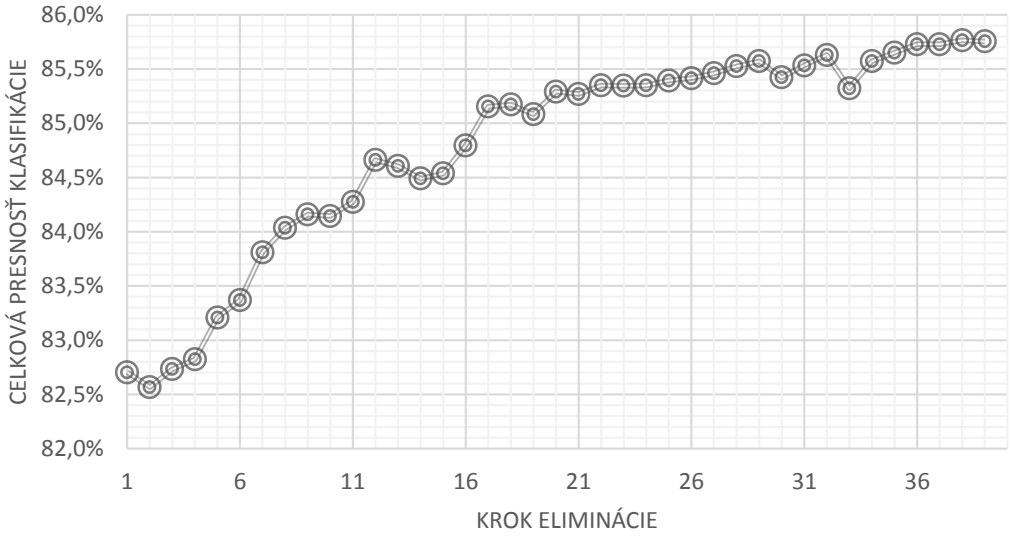
[1] Závislosť medzi modelmi emócií bola vysvetlená v kap. 3.1.6 – ide o nevyhnutnú optimalizáciu pamäťovej a výpočtovej náročnosti, keďže jednotlivé databázy by museli byť upravované pre emočné modely oddelene, predspracovanie by tiež bolo nutné spustiť 5x (pre každý emočný model).

Tab. 4.3: Vývoj hodnôt presnoti klasifikácie českého jazyka v priebehu eliminácie

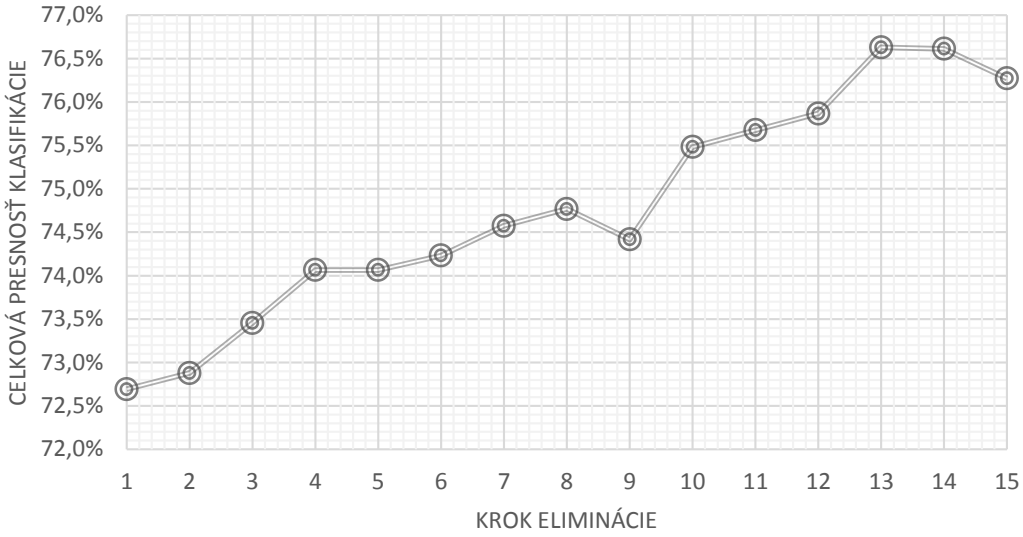
It.	**Presnosť modelu emócie** obava smútok spokojnosť hnev prekvapenie	**Eliminovaný parameter**	**Presnosť klasifikácie**
1	71.43% 92.08% 79.77% 81.66% 88.57%		82.70%
2	71.43% 92.08% 79.09% 81.66% 88.57%	efekt	82.57%
3	69.64% 92.92% 80.23% 82.30% 88.57%	sem ✓	82.73%
4	69.64% 92.92% 80.68% 82.30% 88.57%	akční ✓	82.82%
5	71.43% 92.92% 80.68% 82.73% 88.29%	dát ✓	83.21%
6	71.43% 92.92% 80.91% 82.73% 88.86%	americký ✓	83.37%
7	73.21% 92.92% 80.91% 83.16% 88.86%	autentický ✓	83.81%
8	73.21% 93.75% 80.91% 83.16% 89.14%	bezcenný ✓	84.03%
...	...	...	...
17	76.79% 93.75% 82.50% 82.73% 90.00%	evidentní ✓	85.15%
18	76.79% 93.75% 82.27% 83.05% 90.00%	český ✓	85.17%
19	76.79% 93.75% 82.05% 82.84% 90.00%	jistota	85.08%
20	76.79% 93.75% 82.73% 82.62% 90.57%	pletka ✓	85.29%
21	76.79% 93.75% 82.73% 82.52% 90.57%	esej	85.27%
22	76.79% 93.75% 82.73% 82.62% 90.86%	pokračování ✓	85.35%
23	76.79% 93.75% 82.50% 82.84% 90.86%	dávno	85.35%
24	76.79% 93.75% 82.73% 82.62% 90.86%	dement	85.35%
25	76.79% 93.75% 82.95% 82.62% 90.86%	celosvětový ✓	85.39%
26	76.79% 93.75% 82.95% 82.73% 90.86%	platný ✓	85.42%
27	76.79% 93.75% 83.18% 82.73% 90.86%	hit ✓	85.46%
28	76.79% 93.75% 83.18% 83.05% 90.86%	bezvýrazný ✓	85.52%
29	76.79% 94.17% 83.18% 83.16% 90.57%	moci ✓	85.57%
...	...	...	...
36	76.79% 94.17% 83.18% 83.37% 91.14%	dojít ✓	85.73%
37	76.79% 94.17% 83.18% 83.37% 91.14%	exaktní	85.73%
38	76.79% 94.17% 83.18% 83.26% 91.43%	drobek ✓	85.77%
39	76.79% 94.17% 82.95% 83.16% 91.71%	inteligentní	85.76%

z kroku 38. Priemerné zlepšenie úspešnosti pomocou sekvenčnej eliminácie bolo stanovené na 1,69%, ktoré bolo určené zo 6 meraných etáp pri nasadzovaní systému klasifikácie emócií.

Pre anglický jazyk došlo k zlepšeniu úspešnosti klasifikácie o 3,58% na sledovanej eliminácií, pričom výsledná úspešnosť dosahuje 76,63%. Priebehy eliminácií pre oba jazyky, so všetkými krokmi eliminácie, sú znázornené na Obr. 4.1 pre český jazyk a na Obr. 4.2 pre anglický jazyk.



Obr. 4.1: Priebeh sekvenčnej eliminácie pre klasifikáciu českého jazyka



Obr. 4.2: Priebeh sekvenčnej eliminácie pre klasifikáciu anglického jazyka

4.1.2 Optimalizácia metódou zoskupovania slov

Druhou meranou optimalizačnou metódou je metóda zoskupovania slov. Táto kompresia vstupných parametrov pomáha zabraňovať pretrénovaniu modelu, vďaka čomu je možné dosiahnuť vyššiu úspešnosť klasifikácie v reálnej prevádzke. Počty slov v jednotlivých definovaných skupinách sú znázornené v Tab. 3.2.

Tab. 4.4: Dopad optimalizácie zoskupovania slov na presnosť klasifikácie

Jazyk	**Presnosť modelu emócie** obava smútok spokojnosť hnev prekvapenie	**Presnosť klasifikácie**
Bez optimalizácie		
Čeština	75,00% 95,83% 80,91% 81,98% 89,71%	84,69%
Angličtina	79,81% 75,44% 56,73% 69,09% 76,85%	71,58%
S optimalizáciou		
Čeština	82,14% 96,25% 82,05% 83,16% 90,86%	86,89%
Angličtina	79,81% 75,44% 57,69% 70,00% 76,85%	71,96%

4.1.3 Optimalizácia rozširovaním trénovacej množiny

Treťou optimalizačnou metódou je metóda rozširovania trénovacej množiny o staticky definované vzorky. Ide o komplexnú metódu, ktorá mení správanie a prístup k trénovaniu klasifikátoru. Frázy určené pre použitie v tejto metóde sú založené na spätnej väzbe užívateľov, resp. metóda vyžaduje spoluprácu zamestnancov zákazníckej podpory, ktorí poskytujú spätnú väzbu, na základe ktorej bola zostavená množina textov pre túto optimalizačnú metódu. Z tohto dôvodu budú výsledky tejto metódy demonštrované len pre český jazyk.

Výsledky optimalizácie je možné rozdeliť do 6 testov, keďže množina bola rozširovaná postupne v 6 etapách. V každej etape pribudli nové vzorky, čo popisuje aj Tab. 4.5. Jednotlivé etapy sú označené číslami. Po definícií novej množiny textov je spustený celý proces trénovania ako aj spomínanej sekvenčnej eliminácie. Na začiatku eliminácie u každej etapy teda dôjde k zníženiu nameranej presnosti klasifikácie, pričom sa očakáva zvýšenie presnosti na konci eliminácie, a teda i zvýšenie presnosti klasifikácie celého modelu oproti predošlej etape. Pri jednotlivých testoch je zároveň uvedená aktuálna veľkosť množiny textov (v tomto prípade ide o jednoduché vety s jednoznačne obsiahnutou danou emóciou). Celkový priebeh eliminácie a úspešnosť celého modelu, ako aj jeho čiastkových častí je uvedený na Obr. 4.3.

Pre porovnanie bol prevedený experiment, kedy mal človek určiť správnu emóciu vopred označeného textu. Na 200 vybraných textoch z testovacej množiny dosiahol človek úspešnosť 88,50%. Test prebiehal na 5 ľuďoch, pričom každý zúčastnený mal určiť emóciu u 40 textov. Človek teda správne určil emočnú triedu v 177 prípadoch. Určovanie emócií textu človekom je ovplyvňované rôznymi faktormi ako sú únava, emočné a psychické rozpoloženie, kvôli ktorým môže emóciu v danom texte pochopiť inak.

Tab. 4.5: Vývoj presností v jednotlivých etapách rozširovania trénovacej množiny

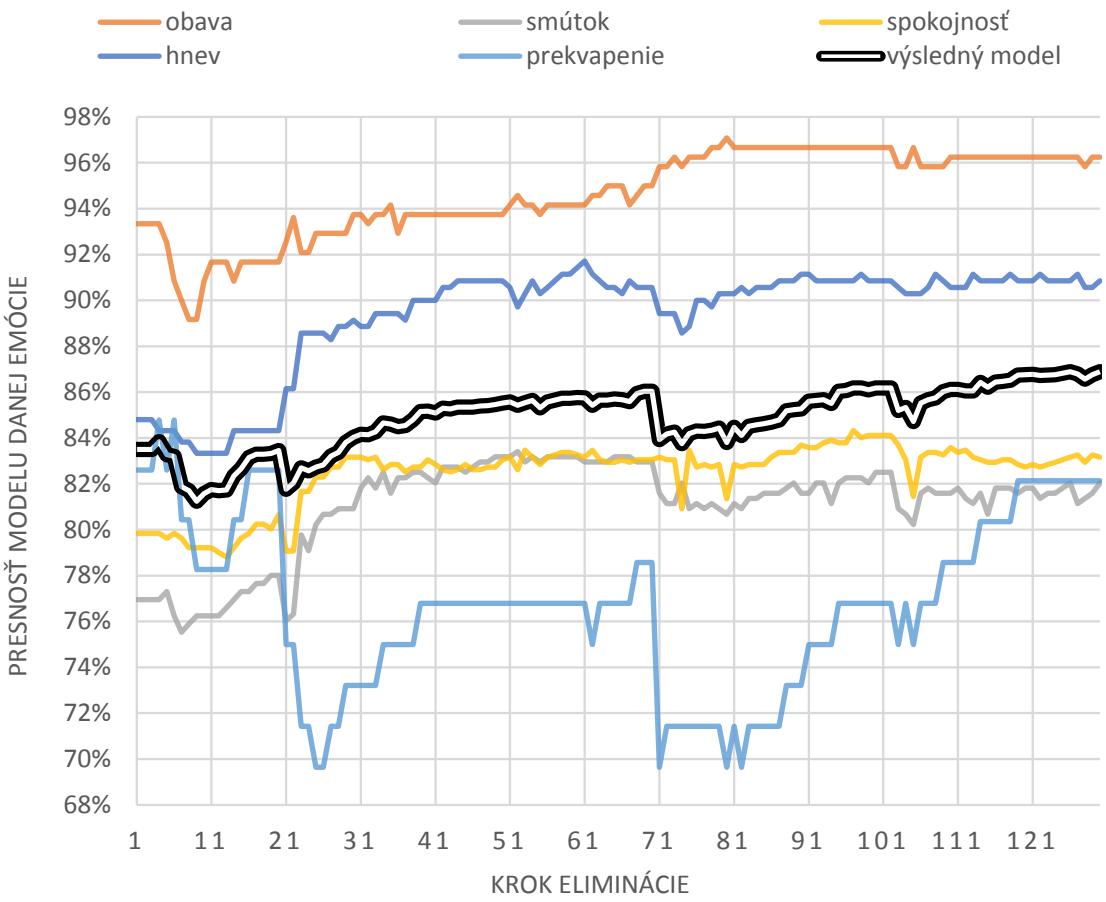
Etapa	1	2	3	4	5	6
Emócia	**Počet vzoriek rozširujúcej množiny**					
Obava	46	50	50	50	81	95
Hnev	61	61	63	63	92	103
Smútok	42	42	42	42	61	68
Spokojnosť	29	29	29	29	34	37
Prekvapenie	1	1	4	4	13	16
Neutrál	1	1	1	1	1	1
Suma	180	184	189	189	282	320
Presnosť klasifikácie						
na začiatku	75,49%	78,97%	81,70%	82,70%	83,93%	85,20%
na konci	76,44%	80,37%	83,30%	85,77%	86,18%	**86,89%**
zlepš. eliminácie	0,95%	1,40%	1,60%	**3,06%**	2,25%	1,69%
zlepš. v etape	—	3,93%	2,93%	2,47%	0,41%	0,71%

4.2 Abstrahovaný systém pre klasifikáciu textu

Minimalizácia jazykových závislostí bola docielená odstránením jazykovo závislých častí predspracovania. Takýto krok má za následok radikálne zníženie úspešnosti. Navrhnutý systém pracuje však s veľkým objemom dát (termín „Big Data“ zavedený v článku [88]), takže je možné daný klasifikátor naučiť základné črty jazyka – niektoré často sa vyskytujúce kombinácie slov, slovných spojení, tvarov slov s určitými predložkami, pochopiť negáciu apod. Predpokladom je, že dostatočným počtom dát je možné zvýšiť úspešnosť klasifikácie.

Tak ako v predošlom prípade, testy boli prevedené pomocou druhej polovice zozbieranej databázy (veľkosť databáz zachytáva Tab. 3.3). Hodnotenie správnosti klasifikácie do 2 tried (pozitívny a negatívny) prebiehal obdobne ako v predošlom prípade – prah klasifikácie nastavený na minimálnu istotu výsledku rovnú 0,5, pričom nebola použitá žiadna zakázaná oblasť.

Pomocou navrhnutého distribuovaného riešenia bolo validovaných 720 rôznych konfigurácií. Každá z konfigurácií bola trénovaná na vytvorenej databáze s veľkým objemom dát (napr. 235 tisíc vzoriek u anglického jazyka), ako popisuje kapitola 3.2.1. V experimente bol použitý princíp vyváženia jednotlivých množín (pozitívnej a negatívnej), trénovacia množina predstavovala prvých 50% dát, testovacia množina zostávajúcich 50%. Keďže jedna iterácie môže byť časovo náročná, výpočet bol distribuovaný na 112 výpočtových jednotiek (jadier). Vďaka tomu bolo možné dosiahnuť



Obr. 4.3: Priebeh sekvenčnej eliminácie a metódy rozširovania trénovacej množiny pre klasifikáciu emócií českého jazyka

výsledky v prijateľnom čase (rádovo jednotky hodín). Experiment porovnáva rôzne úrovne n-gramov, ktoré zvyšujú dimenzionalitu vstupu (slová sú spájané v kontexte viet a vznikajú nové vstupné parametre), ako aj rôzne hodnoty MDF, vďaka ktorým sa dimenzionalita znižuje (odfiltrovaním parametrov, ktoré sa v trénovacej množine nevyskytujú v dostatočnom množstve). Najlepšie dosiahnuté výsledky sú zhrnuté v Tab. 4.6, ktoré boli publikované na medzinárodnej konferencií a uverejnené v článku [83], výsledky boli využité v projekte „Pokročilé meteorologické informace pro letectví“ pre detekciu dôležitých a NOTAM správ a redukovanie počtu týchto správ, ktoré musí pilot prečítať behom svojej predletovej prípravy.

Z výsledkov v Tab. 4.6 vyplýva, že vo všeobecnosti pre klasifikáciu textových dát trénované veľkými objemami dát postačujú 2-gramy. Tab. 4.7 toto tvrdenie potvrdzuje, keďže najlepšie dosiahnuté úspešnosti z pomedzi meraných konfigurácií n-gramu a parametru MDF dosahovala úroveň n-gramu rovná 2. Táto konfigurácia sa zdá byť vhodná pre „Big Data“ prístup, keďže dokáže vyriešiť problém

Tab. 4.6: Porovnanie najlepších konfigurácií pre klasifikáciu Big Data

Jazyk	**Iterácií**	**MDF**	**Úroveň *n*-gramu**	**Presnosť**
Čeština	10	12	2	89.05%
Angličtina	100	32	2	**95.31%**
Nemčina	10	12	2	88.34%
Španielčina	50	12	2	93.23%

negovania. Úspešnosť klasifikácie dosiahnutá pomocou tri-gramov bola napr. u angličtiny o 1,15% nižšia, u 4-gramov dokonca o 1,61% nižšia. Vzhľadom na tieto výsledky boli v ďalších experimentoch použité len hodnoty n-gramu rovné 2 alebo 3, čím bolo dosiahnuté značné zníženie pamäťovej náročnosti ďalších meraní.

Tab. 4.7: Presnosť klasifikácie pre rôzne úrovne n-gramov

Jazyk	**Úroveň *n*-gramu**			
	bez	2	3	4
Čeština	88,72%	**89,05%**	88,66%	88,43%
Angličtina	93,76%	**95,31%**	94,16%	93,70%
Nemčina	87,11%	**88,34%**	87,86%	87,67%
Španielčina	91,41%	**93,23%**	93,22%	93,11%

Pamäťovú náročnosť znižuje vhodné nastavenie parametru MDF, ktorý spôsobuje odfiltrovanie vstupných parametrov na základe ich početnosti výskytu. Odfiltrovaním príznakov metódou MDF sa zbavíme zbytočných informácií a zároveň čiastočne predchádzame pretrénovaniu – napr. pre české texty by bol počet príznakov cez 2,1 milióna, po odfiltrovaní pomocou MDF=12 je počet príznakov už len 67 250, čo je vhodný počet pre veľkosť trénovacej databázy 99 222 vzoriek. Tab. 4.8 znázorňuje vplyv tohto parametru na úspešnosť klasifikácie. Tento konfiguračný parameter má u niektorých jazykoch len minimálny vplyv na výslednú úspešnosť – napr. u nemeckého jazyku je rozptyl hodnôt presnosti len 0,14%. U angličtiny je možné pozorovať iný jav – napriek tomu, že má omnoho nižší počet slov než čeština (len 55 tisíc voči 800 tisíc v českom jazyku), výskyt neodfiltrovaných slovných spojení je omnoho vyšší. Práve z tohto dôvodu vykazuje vyššiu presnosť klasifikácie pri vyšších hodnotách MDF. Tieto hodnoty sú však stále dostatočne nízke a teda k strate žiadnej dôležitej informácie v tak veľkej množine dát nedochádza.

Navrhnutý klasifikátor dosahuje najlepšiu úspešnosť klasifikácie na anglickom jazyku, kde presnosť 95,31% predstavuje zvýšenie presnosti až o 11% voči tradičnému

Tab. 4.8: Presnosť klasifikácie pre rôzne hodnoty MDF

Jazyk	**Hodnota MDF**				
	12	20	24	28	32
Čeština	**89,05%**	88,93%	88,79%	88,68%	88,51%
Angličtina	94,70%	95,07%	95,17%	95,25%	**95,31%**
Nemčina	88,34%	88,01%	87,78%	87,58%	87,37%
Španielčina	93,23%	93,08%	93,04%	92,97%	92,89%

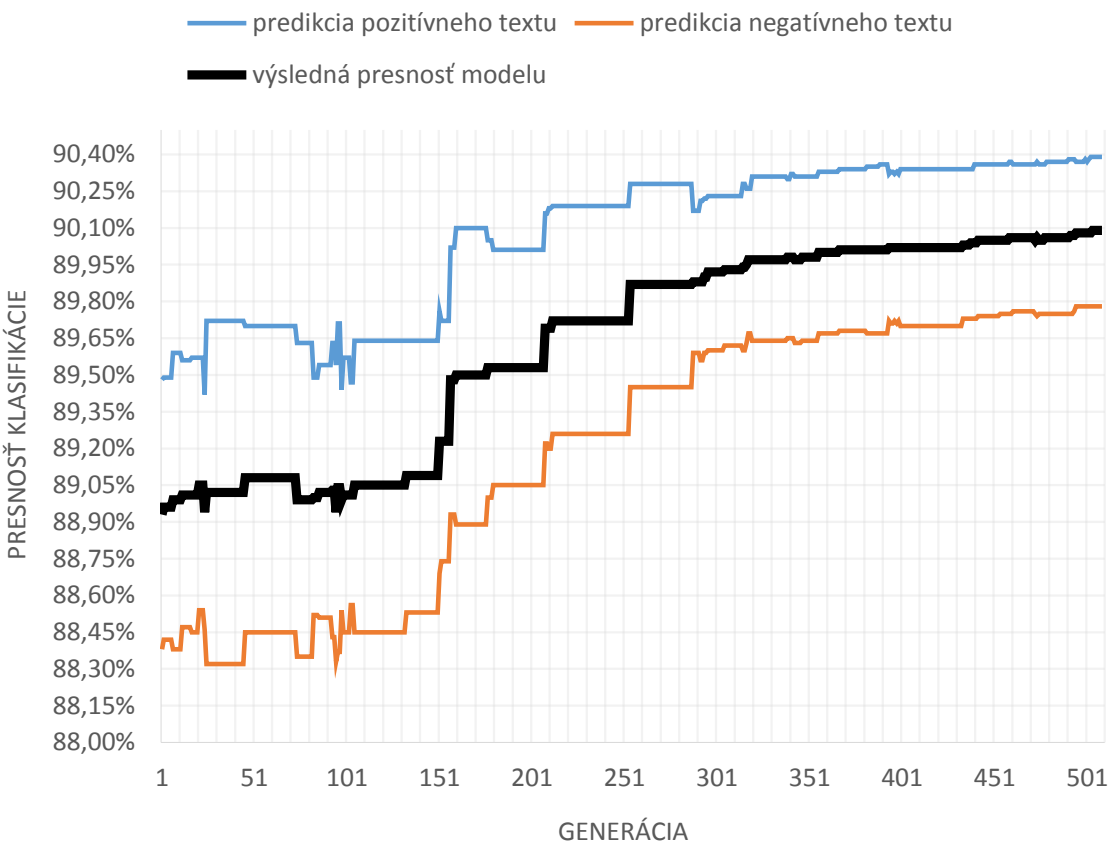
prístupu bez použitia „Big Data“. Navrhnuté riešenie je navyše jazykovo nezávislé, predspracovanie je časovo menej náročné a neobsahuje žiadne časti, ktoré by potrebovali úpravu pri nasadení klasifikátoru na iný jazyk.

4.2.1 Optimalizácia genetickým programovaním

Pre ďalšie zvýšenie presnosti klasifikácie bola navrhnutá optimalizačná metóda založená na algoritme genetického programovania. Metóda ma za úlohu nájsť optimálnu konfiguráciu spomínaných dvoch parametrov (úroveň n-gramu a hodnota MDF) a podmnožinu povolených vstupných parametrov (slov a slovných spojení), ktoré budú dosahovať na testovacej množine čo najvyššiu presnosť klasifikácie. Ohodnotenie prebieha pomocou druhej polovice vytvorenej databázy textov. Výsledky tejto optimalizačnej metódy boli publikované v článku [84] a demonštrované na medzinárodnej konferencií v Indii.

Konfigurácia využívajúca 2-gramy a MDF rovné 12 pre český jazyk a 32 pre anglický jazyk dosahovali najvyššiu úspešnosť v predošlých experimentoch. Tieto hodnoty boli použité v inicializačnej populácií algoritmu GP. Veľkosť množiny povolených parametrov predstavuje 99,5% z celkového počtu parametrov (závisí od úrovne n-gramu a hodnoty MDF, konkrétne počty sú v Tab. 3.5). Zbytok chromozómov do populácie bol vygenerovaný náhodne v rozsahoch 2 až 3 pre úroveň n-gramu a 1 až 32 pre hodnotu MDF. Vyššie hodnoty u týchto parametrov nepredstavovali možnosť zlepšenia úspešnosti klasifikácie.

Obr. 4.4 znázorňuje vývoj úspešnosti klasifikácie českých textov v jednotlivých generáciách behu algoritmu GP. Prvá generácia vykazuje úspešnosť 88,93%, čo je o 0,12% nižšie než systém bez selekcie príznakov (podľa množiny povolených parametrov). Tento pokles je spôsobený znížením počtu vstupných parametrov. Konfigurácia pozostávala z nájdených optimálnych hodnôt z predošlého experimentu a náhodne zvolených príznakov. Posledná dosiahnutá generácia má poradové číslo 509, kde bola dosiahnutá úspešnosť 90,09%, čo predstavuje zlepšenie presnosti



Obr. 4.4: Priebeh úspešnosti klasifikácie pri optimalizácií GP

klasifikácie o 1,04% voči predošlému riešeniu. Tabuľka s kompletným prehľadom výsledkov vývoja úspešnosti je v prílohe A v Tab. A.1 až Tab. A.4.

4.3 Hlboká neurónová sieť založená na RCK jadre

Odstránenie všetkých jazykovo závislých častí bolo možné vďaka návrhu založenom na hlbokom učení a novo navrhnutých RCK jadrách, ktorých štruktúra je navrhnutá priamo na použitie na textových dátach. Tieto jadrá disponujú veľkou schopnosťou učenia a vysokou informačnou priepustnosťou, takže sú schopné naučiť sa daný jazyk. Predpokladom je opäť dostatočný objem dát, aby navrhnutá neurónová sieť z predložených dát pochopila štruktúru a gramatiku daného jazyka. Validácia metódy prebiehala v dvoch samostatných experimentoch.

Prvý experiment demonštruje jazykovú nezávislosť navrhnutej metódy. U predošlého riešenia postaveného na distribuovaných výpočtoch a veľkom objeme dát nebola zabezpečená úplná jazyková nezávislosť, keďže navrhnutý klasifikátor počíta s existenciou oddelených slov vo vstupnom texte. Takéto riešenie nie je vhodné

pre jazyky ako je napr. čínština, kde slová nie sú oddelené prostredníctvom medzery (prípadne iného bieleho znaku). Experiment teda dokazuje, že navrhnuté riešenie je vhodné pre akýkoľvek druh jazyka. Testované boli tieto jazyky:

- angličtina – ako zástupca anglo-frízskeho jazyka, množina testovacích textov bola cielená na bežnú komunikáciu,
- nemčina – ako zástupca germánskych jazykov, testuje schopnosť spracovať dlhšie slová a zloženiny, komplexnú gramatiku,
- čeština – ako zástupca slovanských jazykov, zameriava sa na slová s diakritikou a bez, ako aj na časté chyby v pravopise a prípadné preklepy,
- španielčina – ako zástupca románskych jazykov, množina textov je zameraná prevažne na krátke texty,
- čínština – pre dokázanie, že metóda je schopná pracovať s rôznymi abecedami syntaxami, jazyk nespolieha na jednoznačné oddelenie slov.

Druhý experiment porovnáva úspešnosť navrhnutého riešenia so súčasnými metódami použitými v iných publikáciach. K týmto účelom boli využité verejné databázy textov – databáza hodnotení Yelp a databáza hodnotení Amazon, ktoré sú používané širokou vedeckou komunitou pri porovnávaní nových navrhnutých metód zameraných na dolovanie znalostí z textových dát. Tieto databázy poskytujú dostatok dát na to, aby boli použité na trénovanie a testovanie hlbokých neurónových sietí.

Oba experimenty používajú princíp rozdelenia databázy textov na 3 časti: na trénovaciu množinu, validačnú množinu a testovaciu množinu. Úspešnosti zobrazené v tabuľkách sú získané meraním na testovacej množine. Validačná množina bola použitá len na riadenie procesu trénovania. Text vstupuje do neurónovej siete v tzv. „surovom“ stave, žiadne predspracovanie nebolo aplikované na textové dáta. Klasifikácia prebiehala do dvoch tried – pozitívnej a negatívnej. U verejných databáz bol použitý rovnaký princíp ako u súčasných publikáciách, teda do pozitívnej triedy spadali len texty s najvyšším priradeným hodnotením, do negatívnej triedy texty s najnižším hodnotením, tak ako popisuje Tab. 3.8.

Experiment prebiehal na rôznych hlbokých neurónových sieťach pozostávajúcich z RCK jadier. Jednotlivé navrhnuté štruktúry využívajú sériové zapojenie týchto jadier, kde sa mení počet jadier a obmedzenie počtu konvolučných jadier v jednotlivých vetvách RCK jadra. Štruktúry obsahujú tiež plne prepojené vrstvy, kde počet neurónov bol volený (na základe ďalších experimentov) v rozmedzí 128 až 512. U všetkých konvolučných vrstiev boli použité „dropout“ vrstvy s pravdepodobnosťou 1%, u plne prepojených vrstiev s pravdepodobnosťou 10%. Tieto vrstvy zabezpečujú robustnosť natrénovaného modelu a zabraňujú pretrénovaniu modelu, čo by malo za následok zníženie úspešnosti na testovacej množine.

Tab. 4.9: Súhrn navrhnutých štruktúr hlbokých neurónových sietí

Č.	**Parametre RCK jadra**		**Počet neurónov plne prepojených**	**Trénovateľných parametrov**
	hĺbka	počet konv. jadier		
1	6	64, 64, 64, 32, 32, 32	256, 128	3 311 970
2	4	64, 64, 64, 32	256, 128	6 779 298
3	5	64, 64, 64, 32, 32	128, 128	2 783 490
4	4	64, 64, 64, 32	128, 128	3 793 186
5	5	64, 64, 128, 256, 256	512, 256	27 746 818
6	4	64, 64, 64, 64	256, 128	7 807 362
7	7	64, 64, 64, 64, 64, 32, 32	256, 128	3 761 410
8	7	64, 64, 64, 64, 64, 32, 32	128, 128	3 127 938

Tab. 4.9 zhrňuje štruktúry týchto sietí, v prílohe B je následne ukážková neurónová sieť (č. 6) vykreslená kompletne. Vybraná sieť má hĺbku 4 RCK jadier, je teda jednou z plytkých sietí, ktoré boli navrhnuté pre experiment. Hlbšia štruktúra by nemohla byť vložená priamo do tejto práce z dôvodu komplexnosti štruktúry a rozmerov jej grafického znázornenia. Z vytvorených štruktúr bola hľadaná tá, ktorá dosiahne najvyššiu úspešnosti klasifikácie, kde k porovnaniu slúžila práve testovacia množina dát.

Tab. 4.10 zobrazuje dosiahnuté úspešnosti klasifikácie u rôznych jazykov. Rozptyl presností 12,65% a priemerná úspešnosť klasifikácie 87,71% dokazujú, že metóda je aplikovateľná na rôznych jazykoch. Ako u každej metódy založenej na hlbokom učení, aj tu rozhoduje kvalita vytvorenej databázy a priamo ovplyvňuje možné dosiahnuteľné úspešnosti. Pre španielský a čínsky jazyk boli dosiahnuté najvyššie úspešnosti – 92,46% a 91,22%.

Výsledky z verejných databáz textov sú zhrnuté v Tab. 4.11, kde sieť č. 8 dosahuje najlepšie výsledky. Táto sieť pozostáva zo 7 RCK jadier a počet konvolučných jadier v jednotlivých úrovniach siete nepresahuje číslo 64. Natrénovaný model má 3,3 milióna trénovateľných parametrov, ktoré je možné reprezentovať v pamäti pomocou 24MB dát (o 85% nižšia než u najjednoduchšieho modelu GloVe [80]). Z toho vyplýva, že takýto model je možné použiť aj na jednoduchých vstavaných systémoch (tzv. „embedded“ systémy), ktoré figurujú nízkymi parametrami technického vybavenia – teda aj malou operačnou pamäťou. V porovnaní s riešeniami postavenými na modeloch zhlukovania slov ide o omnoho nižšiu pamäťovú náročnosť. Modely zhlukovania slov vyžadujú stovky MB – napr. najjednoduchší pred-trénovaný model Glove vytvorený Stanfordskou univerzitou [80] má 171MB, komplexnejšie modely zaberajú viac než 1GB pamäte, model Word2Vec [69] zaberá 1,5GB a model fastText [5] od spoločnosti Facebook má 280MB pre anglický

Tab. 4.10: Výsledky klasifikácie pomocou hlbokej neurónovej siete

Čeština model 1 s 3 493 602 trénovateľných parametrov

	skutočne pozitívny	skutočne negatívny	senzitivita
predikovaný pozitívny	12 422	2 578	82,81%
predikovaný negatívny	1 396	13 604	90,69%
precíznosť	89,90%	84,07%	
		presnosť klasifikácie	**86,75%**

Angličtina model 8 s 3 296 898 trénovateľných parametrov

	skutočne pozitívny	skutočne negatívny	senzitivita
predikovaný pozitívny	12 860	2 140	85,73%
predikovaný negatívny	2 435	12 565	83,77%
precíznosť	84,08%	85,45%	
		presnosť klasifikácie	**84,75%**

Nemčina model 5 s 28 321 282 trénovateľných parametrov

	skutočne pozitívny	skutočne negatívny	senzitivita
predikovaný pozitívny	11 974	3 026	79,83%
predikovaný negatívny	1 958	13 042	86,95%
precíznosť	85,95%	81,17%	
		presnosť klasifikácie	**83,39%**

Španielčina model 8 s 3 296 898 trénovateľných parametrov

	skutočne pozitívny	skutočne negatívny	senzitivita
predikovaný pozitívny	13 794	1 206	91,96%
predikovaný negatívny	1 055	13 945	92,97%
precíznosť	92,90%	92,04%	
		presnosť klasifikácie	**92,46%**

Čínština model 7 s 3 947 266 trénovateľných parametrov

	skutočne pozitívny	skutočne negatívny	senzitivita
predikovaný pozitívny	8 194	897	90,13%
predikovaný negatívny	699	8 392	92,31%
precíznosť	92,14%	90,34%	
		presnosť klasifikácie	**91,22%**

Tab. 4.11: Výsledky klasifikácie na verejných databázach

Databáza	**Štruktúra č.**	**Úspešnosť**	
		validačná	testovacia
Yelp	8	96.14%	**96.15%**
	2	96.10%	96.05%
	1	96.02%	96.03%
Amazon	3	94.74%	**94.59%**
	1	94.62%	94.57%
	2	94.19%	94.02%

jazyk. Natrénovať model pozostávajúci zo 7 RCK jadier na databáze hodnotení Yelp zabralo 38 hodín pri trénovaní na osobnom počítači s procesorom Intel Core i7-2600K s frekvenciou 3,4GHz, 32GB operačnej pamäte a grafickým akcelerátorom Nvidia GTX 1080Ti s 12GB operačnej pamäte. Tento čas je omnoho vyšší, než pri použití modelov zhlukovania slov, avšak navrhnuté riešenie je aplikovateľné na všetky jazyky vrátane strojových jazykov (strojovo generované logy apod.).

V porovnaní so súčasnými metódami je navrhnutá metóda kompletne jazykovo nezávislá (t.j. nepoužíva žiadne predspracovanie dát, nie je závislá na modeloch zhlukovania slov, nevyužíva žiadnu pripravenú znalosť o analyzovanom jazyku textu), má porovnateľnú úspešnosť než súčasné „state-of-the-art“ metódy (znázorňuje Tab. 4.12) a je 10-násobne pamäťovo efektívnejšia než riešenia postavené na modeloch zhlukovania slov. U databáze hodnotení Yelp dosahuje dokonca o 0,5% vyššiu úspešnosť klasifikácie, než súčasné metódy. Metóda je tiež vhodná na strojovo generované texty, u ktorých nie je známa gramatika. Keďže sa aj do budúcna predpokladá zvyšovanie výpočtovej kapacity ako aj pamäťových možností grafických akcelerátorov, navrhnutá metóda je vďaka masívnemu paralelizmu vhodná náhrada za modely zhlukovania slov.

Tab. 4.12: Porovnanie navrhnutej metódy so súčasnými riešeniami na databáze Yelp

Model	**Rok, publikácia**	**Úspešnosť**
Bag of Words	2015, [118]	92.2%
n-grams	2015, [118]	95.6%
Char-CNN	2015, [118]	94.7%
Char-CRNN	2016, [114]	94.5%
Very deep CNN	2016, [18]	95.7%
FastText	2016, [5]	93.8%
FastText with bigrams	2016, [5]	95.7%
Discriminative LSTM	2017, [116]	92.6%
Generative LSTM	2017, [116]	90.0%
RCK č. 8	2018, v recenzií	**96.2%**

5 Záver

Práca sa zaoberá problémom dolovania znalostí z textových dát, ktorý je stále aktuálnejší vzhľadom na exponenciálny rast množstva uložených dát v elektronickej podobe. Vďaka súčasným trendom je možné predpokladať zvýšený záujem o spracovanie týchto dát a ich analýzu. Vývoj nového výpočtového technického vybavenia so zameraním na masívny paralelizmus umožňuje túto analýzu značne urýchliť.

Časť riešenia práce sa venuje porovnaniu tradičných metód a súčasných nových metód založených na hlbokom učení, pričom sa zameriava na problém analýzy textových dát, ktorý bol demonštrovaný na probléme klasifikácie emócií autora daného textu. Hlavným problémom súčasných metód je závislosť na konkrétnom jazyku textu a ich presnosť, ktorá nedosahuje úspešnosti človeka.

V práci bolo navrhnuté riešenie pozostávajúce z tradičnej metódy a jej optimalizácie, abstrahovanie tejto metódy a jej optimalizácií, a vývoja novej metódy pre strojové porozumenie textu bez znalosti jazyka a gramatiky. U tradičných metód bola navrhnutá štruktúra pre detekciu 5 emočných tried, ktorá pozostávala z 5 samostatných klasifikátorov a jedného spoločného predspracovania textu. Pre tento systém boli navrhnuté jazykovo nezávislé optimalizačné metódy s cieľom zvýšiť úspešnosť klasifikácie. Abstrahovaním tejto metódy prostredníctvom odstránenia jazykovo závislejších častí a použitím prístupu „Big Data“ bol dosiahnutý návrh všeobecnejšieho systému, ktorý bol následne optimalizovaný prostredníctvom navrhnutého algoritmu genetického programovania. U tradičných metód nebolo možné dosiahnuť vyššiu mieru jazykovej nezávislosti riešenia (riešenie nebolo vhodné napr. pre čínsky jazyk, ktorého syntax nevyužíva slová), preto ďalší výskum smeroval na nové metódy založené na hlbokom učení. Po návrhu prevodu vstupných textových dát do maticovej podoby bola navrhnutá štruktúra hlbokej neurónovej siete so zameraním na textové dáta. Pri návrhu sa vychádzalo z existujúcich štruktúr často používaných pri obrazovom spracovaní. V navrhnutých štruktúrach boli identifikované opakujúce sa vzory, ktoré boli izolované do novej štruktúry nazvanej „RCK jadro“. Táto bola použitá pri návrhu komplexnejších hlbokých neurónových sietí vďaka jej schopnosti extrakcie informácií, vysokej informačnej priepustnosti a jej kapacite.

Hlavným prínosom tejto práce je navrhnutá metóda pre dolovanie znalostí z textových dát, ktorá bola experimentálne overená na vybranom prípade klasifikácie textu, kde bola dosiahnutá vyššia presnosť v porovnaní so súčasným stavom vedy a techniky (prekonanie o 0,5%) a zníženie pamäťovej náročnosti o 85% v porovnaní so súčasnými metódami postavenými na modeloch zhlukovania slov. Metóda je univerzálna pre všetky jazyky (prirodzené či strojové), pre svoju funkčnosť nevyžaduje predošlú znalosť jazyka a gramatiky. Výsledky obsiahnuté v tejto práci boli publikované v časopisoch s impakt faktorom [87] (IF pre rok 2016 = 0,945), riešenie

založené na novo-navrhnutom RCK jadre bolo v dobe písania práce v recenznom konaní k publikácií v časopise Cognitive Computation (IF pre rok 2016 = 3,441).

Prvá časť návrhu sa zaoberala metódou rozpoznávania 5 emočných tried (hnev, smútok, spokojnosť, prekvapenie a obava) v textových správach zákazníckej podpory. Overenie metódy prebiehalo na dvoch jazykoch – čeština (1673 vzoriek textu) a angličtina (279 vzoriek textu), kde bola dosiahnutá presnosť klasifikácie 75,49% a 71,58%. Použitím troch navrhnutých optimalizačných metód došlo k zvýšeniu presnosti pre český jazyk o 11,40% (teda na 86,89%) a pre anglický jazyk o 5,05% (teda na 76,63%). Výstupom tejto časti je návrh štruktúry jazykovo nezávislého systému, ktorý môže byť použitý pre prioritizáciu textových správ zákazníckej podpory.

Ďalšia časť sa venovala abstrahovanej metóde pre valenciu textu založenej na SVM klasifikátore trénovaného prostredníctvom veľkých objemov dát (zvaný tiež „Big Data“ prístup). Klasifikátor nevyužíva žiadne jazykovo závislé predspracovanie vstupného textu, ani žiadnu jazykovo závislú optimalizačnú metódu. Funkčnosť bola overená na 4 jazykoch, kde boli dosiahnuté úspešnosti 89,05% pre český jazyk, 95,31% pre anglický jazyk, 88,34% pre nemecký jazyk a 93,23% pre španielsky jazyk. U angličtiny teda došlo k zlepšeniu o 18,68% a u češtiny o 2,16%. Ďalšou navrhnutou optimalizáciou postavenou na algoritme genetického programovania bola dosiahnutá úspešnosť klasifikácie 90,09% pre český jazyk, čo predstavuje ďalšie zvýšenie presnosti o 1,04%.

Posledná časť sa venovala výskumu metódy pre porozumenie textu založenej na hlbokom učení, so zameraním na všeobecnosť navrhnutej metódy a úplne odstránenie jazykovo závislých častí systému. Metóda nepoužíva žiadne predspracovanie dát, nie je závislá na modeloch zhlukovania slov a nepoužíva žiadnu vopred pripravenú znalosť o analyzovanom jazyku. Metóda využíva hlbokú neurónovú sieť skladajúcu sa z nových navrhnutých RCK jadier určených na extrakciu parciálnych informácií v rôznych úrovniach siete, zároveň poskytujú dostatočnú informačnú priepustnosť, čo umožňuje neurónovej sieti pohľad na dané textové dáta v rôznych mierach abstrakcie. Overenie metódy prebiehalo na 5 jazykoch – čeština, angličtina, nemčina, španielčina a čínština. Rozptyl presností 12,65% a priemerná úspešnosť klasifikácie 87,71% dokazujú možnosť aplikácie metódy na rôzne jazyky. Výsledky presností sú však z dôvodu nedostatku dát (60 tis. až 500 tis. vzoriek) nižšie než u predošlého navrhnutého systému. Funkčnosť a schopnosť presnej klasifikácie je však demonštrovaná na voľne dostupných databázach hodnotení Yelp a Amazon, kde boli dosiahnuté úspešnosti klasifikácie 96,15% a 94,59%. U databáze hodnotení Yelp bola dosiahnutá presnosť vyššia o 0,5% než súčasné metódy publikované výskumnými tímami „Facebook AI Research“ a „Google DeepMind“.

Literatúra

[1] Abdi, A.; Shamsuddin, S. M.; Aliguliyev, R. M.: QMOS: Query-based multi-documents opinion-oriented summarization. *Information Processing & Management*, ročník 54, č. 2, 2018: s. 318–338.

[2] Altman, N. S.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, ročník 46, č. 3, 1992: s. 175–185.

[3] Back, T.: *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[4] Bishop, R. L.; Goldberg, S. I.: *Tensor analysis on manifolds*. Courier Corporation, 2012.

[5] Bojanowski, P.; Grave, E.; Joulin, A.; aj.: Enriching Word Vectors with Subword Information. *Facebook AI Research, ArXiv e-prints*, Červenec 2016, `1607.04606`.

[6] Bottou, L.: Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, Springer, 2010, s. 177–186.

[7] Brown, P. F.; Desouza, P. V.; Mercer, R. L.; aj.: Class-based n-gram models of natural language. *Computational linguistics*, ročník 18, č. 4, 1992: s. 467–479.

[8] Burges, C. J.: A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, ročník 2, č. 2, 1998: s. 121–167.

[9] Burget, R.; Karasek, J.; Smekal, Z.: Classification and detection of emotions in czech news headlines. In *The 33rd International Conference on Telecommunication and Signal Processing, TSP*, ročník 2010, 2010, s. 1–5.

[10] Burget, R.; Karasek, J.; Smekal, Z.: Recognition of emotions in Czech newspaper headlines. *Radioengineering*, ročník 20, č. 1, 2011: s. 39–47.

[11] Caruana, R.; Freitag, D.: Greedy attribute selection. In *Machine Learning Proceedings 1994*, Elsevier, 1994, s. 28–36.

[12] Cavnar, W. B.; Trenkle, J. M.; aj.: N-gram-based text categorization. *Ann arbor mi*, ročník 48113, č. 2, 1994: s. 161–175.

[13] Chakraborty, G.; Krishna, M.: Analysis of unstructured data: Applications of text analytics and sentiment mining. In *SAS global forum*, 2014, s. 1288–2014.

[14] Chellapilla, K.; Puri, S.; Simard, P.: High performance convolutional neural networks for document processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, Suvisoft, 2006.

[15] Christopher, D. M.; Prabhakar, R.; Hinrich, S.: Introduction to information retrieval. *An Introduction To Information Retrieval*, ročník 151, č. 177, 2008: str. 5.

[16] Chuang, Z.-J.; Wu, C.-H.: Multi-modal emotion recognition from speech and text. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 9, Number 2, August 2004: Special Issue on New Trends of Speech and Language Processing*, ročník 9, č. 2, 2004: s. 45–62.

[17] Cireşan, D.; Meier, U.; Schmidhuber, J.: Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*, 2012.

[18] Conneau, A.; Schwenk, H.; Barrault, L.; aj.: Very Deep Convolutional Networks for Text Classification. *Facebook AI Research, ArXiv e-prints*, Červen 2016, `1606.01781`.

[19] Cowie, R.; Douglas-Cowie, E.; Savvidou*, S.; aj.: 'FEELTRACE': An instrument for recording perceived emotion in real time. In *ISCA tutorial and research workshop (ITRW) on speech and emotion*, 2000.

[20] De Silva, J.; Haddela, P. S.: A term weighting method for identifying emotions from text content. In *2013 8th IEEE International Conference on Industrial and Information Systems (ICIIS)*, IEEE, 2013, s. 381–386.

[21] Dietterich, T.: Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, ročník 27, č. 3, 1995: s. 326–327.

[22] Dozat, T.: Incorporating nesterov momentum into adam. 2016.

[23] Duchi, J.; Hazan, E.; Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, ročník 12, č. Jul, 2011: s. 2121–2159.

[24] Elkan, C.: Nearest neighbor classification. *elkan cs. ucsd. edu‖,‖ January*, ročník 11, 2011: str. 3.

[25] Espejo, P. G.; Ventura, S.; Herrera, F.: A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, ročník 40, č. 2, 2010: s. 121–144.

[26] Eurobarometer, S.: 386. 2012. Europeans and their languages. *European Commission*, 2017.

[27] Feng, S.; Wang, Y.; Song, K.; aj.: Detecting Multiple Coexisting Emotions in Microblogs with Convolutional Neural Networks. *Cognitive Computation*, ročník 10, č. 1, 2018: s. 136–155.

[28] Gers, F. A.; Schmidhuber, J.; Cummins, F.: Learning to forget: Continual prediction with LSTM. 1999.

[29] Gokulakrishnan, B.; Priyanthan, P.; Ragavan, T.; aj.: Opinion mining and sentiment analysis on a twitter data stream. In *Advances in ICT for emerging regions (ICTer), 2012 International Conference on*, IEEE, 2012, s. 182–188.

[30] Goldberg, Y.: *Neural network methods for natural language processing*, ročník 10. Morgan & Claypool Publishers, 2017, 1–309 s.

[31] Goldberg, Y.; Levy, O.: word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[32] Graf, H. P.; Cosatto, E.; Bottou, L.; aj.: Parallel support vector machines: The cascade svm. In *Advances in neural information processing systems*, 2005, s. 521–528.

[33] Grave, E.; Mikolov, T.; Joulin, A.; aj.: Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, 2017, s. 3–7.

[34] Guyon, I.; Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research*, ročník 3, č. Mar, 2003: s. 1157–1182.

[35] He, K.; Zhang, X.; Ren, S.; aj.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, s. 770–778.

[36] Ho, T. K.: Recognition of handwritten digits by combining independent learning vector quantizations. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, IEEE, 1993, s. 818–821.

[37] Ho, T. K.: Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, ročník 1, IEEE, 1995, s. 278–282.

[38] Hochreiter, S.; Schmidhuber, J.: Long short-term memory. *Neural computation*, ročník 9, č. 8, 1997: s. 1735–1780.

[39] Huffman, D. A.: The synthesis of sequential switching circuits. 1954.

[40] Hughes, G.: On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, ročník 14, č. 1, 1968: s. 55–63.

[41] Jackel, L.; Boser, B.; Graf, H.; aj.: VLSI implementations of electronic neural networks: An example in character recognition. In *IEEE International Conference onSystems, Man and Cybernetics, 1990. Conference Proceedings.*, IEEE, 1990, s. 320–322.

[42] Jain, A.; Zongker, D.: Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, ročník 19, č. 2, 1997: s. 153–158.

[43] Karpathy, A.; Toderici, G.; Shetty, S.; aj.: Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, s. 1725–1732.

[44] Kingma, D. P.; Ba, L.: J. ADAM: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[45] Kirange, D.; aj.: Emotion classification of news headlines using SVM. *Asian Journal of Computer Science & Information Technology*, ročník 2, č. 5, 2013.

[46] Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; aj.: Skip-thought vectors. In *Advances in neural information processing systems*, 2015, s. 3294–3302.

[47] Kohavi, R.; John, G. H.: Wrappers for feature subset selection. *Artificial intelligence*, ročník 97, č. 1-2, 1997: s. 273–324.

[48] Kolařík, M.: *Hluboké učení pro klasifikaci textů*. Brno, 2017.

[49] Korde, V.; Mahender, C. N.: Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, ročník 3, č. 2, 2012: str. 85.

[50] Kurzyński, M. W.: Decision rules for a hierarchical classifier. *Pattern Recognition Letters*, ročník 1, č. 5-6, 1983: s. 305–310.

[51] Kurzyński, M. W.: The optimal strategy of a tree classifier. *Pattern Recognition*, ročník 16, č. 1, 1983: s. 81–87.

[52] Lai, S.; Liu, K.; He, S.; aj.: How to generate a good word embedding. *IEEE Intelligent Systems*, ročník 31, č. 6, 2016: s. 5–14.

[53] Lauren, P.; Qu, G.; Yang, J.; aj.: Generating Word Embeddings from an Extreme Learning Machine for Sentiment Analysis and Sequence Labeling Tasks. *Cognitive Computation*, 2018: s. 1–14.

[54] Lauzon, F. Q.: An introduction to deep learning. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, IEEE, 2012, s. 1438–1439.

[55] Le, Q.; Mikolov, T.: Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 2014, s. 1188–1196.

[56] LeCun, Y.; Bengio, Y.; Hinton, G.: Deep learning. *nature*, ročník 521, č. 7553, 2015: str. 436.

[57] Lee, H.; Grosse, R.; Ranganath, R.; aj.: Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, ročník 54, č. 10, 2011: s. 95–103.

[58] Li, Q.; Salman, R.; Test, E.; aj.: Parallel multitask cross validation for support vector machine using GPU. *Journal of Parallel and Distributed Computing*, ročník 73, č. 3, 2013: s. 293–302.

[59] Liu, H.; Singh, P.: ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal*, ročník 22, č. 4, 2004: s. 211–226.

[60] Luo, S.; Ghosal, S.: Forward selection and estimation in high dimensional single index models. *Statistical Methodology*, ročník 33, 2016: s. 172–179.

[61] Ma, C.; Prendinger, H.; Ishizuka, M.: Emotion estimation and reasoning based on affective textual interaction. In *International Conference on Affective Computing and Intelligent Interaction*, Springer, 2005, s. 622–628.

[62] Maldonado, S.; Weber, R.; Famili, F.: Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Information Sciences*, ročník 286, 2014: s. 228–246.

[63] Mao, K. Z.: Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, ročník 34, č. 1, 2004: s. 629–634.

[64] Masek, J.; Burget, R.; Karasek, J.; aj.: Multi-GPU implementation of k-nearest neighbor algorithm. In *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*, IEEE, 2015, s. 764–767.

[65] Masek, J.; Burget, R.; Povoda, L.; aj.: Multi–GPU Implementation of Machine Learning Algorithm using CUDA and OpenCL. *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, ročník 5, č. 2, 2016: s. 101–107.

[66] McAuley, J.; Yang, A.: Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2016, s. 625–635.

[67] Mesnil, G.; Dauphin, Y.; Yao, K.; aj.: Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, ročník 23, č. 3, 2015: s. 530–539.

[68] Mesnil, G.; He, X.; Deng, L.; aj.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, 2013, s. 3771–3775.

[69] Mikolov, T.; Chen, K.; Corrado, G.; aj.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[70] Mikolov, T.; Sutskever, I.; Chen, K.; aj.: Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013, s. 3111–3119.

[71] Miller, G. A.: WordNet: a lexical database for English. *Communications of the ACM*, ročník 38, č. 11, 1995: s. 39–41.

[72] Mrňous, J.: *Porovnání vybraných metod strojového učení pro klasifikaci textů*. Brno, 2004.

[73] Murphy, K.: A brief introduction to graphical models and Bayesian networks. 1998.

[74] Murphy, K.: A brief introduction to Bayes' Rule. 2010.

[75] Myška, V.: *Rekurentní neuronové sítě pro klasifikaci textů*. Brno, 2017.

[76] Navarro, G.: A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, ročník 33, č. 1, 2001: s. 31–88.

[77] Niedermayer, D.: An introduction to Bayesian networks and their contemporary applications. In *Innovations in Bayesian networks*, Springer, 2008, s. 117–130.

[78] Palangi, H.; Deng, L.; Shen, Y.; aj.: Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, ročník 24, č. 4, 2016: s. 694–707.

[79] Patil, C. G.; Patil, S. S.: Use of Porter stemming algorithm and SVM for emotion extraction from news headlines. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, ročník 2, č. 7, 2013: str. 9.

[80] Pennington, J.; Socher, R.; Manning, C.: Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, s. 1532–1543.

[81] Phan, X.-H.; Nguyen, L.-M.; Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, ACM, 2008, s. 91–100.

[82] Povoda, L.; Arora, A.; Singh, S.; aj.: Emotion Recognition from Helpdesk Messages. In *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2015, s. 310–313.

[83] Povoda, L.; Burget, R.; Dutta, M. K.: Sentiment analysis based on Support Vector Machine and Big Data. In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2016, s. 543–545.

[84] Povoda, L.; Burget, R.; Dutta, M. K.; aj.: Genetic Optimization of Big Data Sentiment Analysis. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2017, s. 141–144.

[85] Povoda, L.; Burget, R.; Mašek, J.; aj.: Automatické rozpoznávanie emócií z českého textu pomocou umelej inteligencie. In *Elektrorevue - Internetový časopis (http://www.elektrorevue.cz)*, 17.1, 2015, s. 15–18.

[86] Povoda, L.; Burget, R.; Masek, J.; aj.: Job shop scheduling problem with heuristic genetic programming operators. In *Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on*, IEEE, 2015, s. 702–707.

[87] Povoda, L.; Burget, R.; Masek, J.; aj.: Optimization Methods in Emotion Recognition System. *Radioengineering*, ročník 25, č. 3, 2016: s. 565–572.

[88] Power, D. J.: Using Big Data for analytics and decision support. *Journal of Decision Systems*, ročník 23, č. 2, 2014: s. 222–228.

[89] Reddi, S. J.; Kale, S.; Kumar, S.: On the convergence of adam and beyond. 2018.

[90] Safavian, S. R.; Landgrebe, D.: A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, ročník 21, č. 3, 1991: s. 660–674.

[91] Salton, G.; Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information processing & management*, ročník 24, č. 5, 1988: s. 513–523.

[92] Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural networks*, ročník 61, 2015: s. 85–117.

[93] Seiffert, C.; Khoshgoftaar, T. M.; Van Hulse, J.; aj.: Building Useful Models from Imbalanced Data with Sampling and Boosting. In *FLAIRS conference*, 2008, s. 306–311.

[94] Shaheen, S.; El-Hajj, W.; Hajj, H.; aj.: Emotion recognition from text based on automatically generated rules. In *2014 IEEE International Conference on Data Mining Workshop (ICDMW)*, IEEE, 2014, s. 383–392.

[95] Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] Sokolova, M.; Japkowicz, N.; Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, Springer, 2006, s. 1015–1021.

[97] Sriram, B.; Fuhry, D.; Demir, E.; aj.: Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2010, s. 841–842.

[98] Stolpe, M.; Bhaduri, K.; Das, K.: *Distributed Support Vector Machines: An Overview*. Cham: Springer International Publishing, 2016, ISBN 978-3-319-41706-6, s. 109–138, doi:10.1007/978-3-319-41706-6_5.

[99] Strigl, D.; Kofler, K.; Podlipnig, S.: Performance and scalability of GPU-based convolutional neural networks. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, IEEE, 2010, s. 317–324.

[100] Stříteský, R.: *Sémantické rozpoznávání komentářů na webu*. Brno, 2017.

[101] Sutskever, I.; Martens, J.; Dahl, G.; aj.: On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, 2013, s. 1139–1147.

[102] Sutter, J. M.; Kalivas, J. H.: Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection. *Microchemical journal*, ročník 47, č. 1-2, 1993: s. 60–66.

[103] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; aj.: Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, s. 2818–2826.

[104] Tang, D.; Qin, B.; Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, s. 1422–1432.

[105] Tong, S.; Koller, D.: Support vector machine active learning with applications to text classification. *Journal of machine learning research*, ročník 2, č. Nov, 2001: s. 45–66.

[106] Tripathy, A.; Agrawal, A.; Rath, S. K.: Classification of sentimental reviews using machine learning techniques. *Procedia Computer Science*, ročník 57, 2015: s. 821–829.

[107] Turner, V.; Gantz, J.; Reinsel, D.; aj.: The digital universe: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 2014.

[108] Vuduc, R.; Chandramowlishwaran, A.; Choi, J.; aj.: On the limits of GPU acceleration. In *Proceedings of the 2nd USENIX conference on Hot topics in parallelism*, ročník 13, USENIX Association, 2010.

[109] Wan, X.: Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing*, Association for Computational Linguistics, 2008, s. 553–561.

[110] Wang, M.; Cao, D.; Li, L.; aj.: Microblog sentiment analysis based on cross-media bag-of-words model. In *Proceedings of international conference on internet multimedia computing and service*, ACM, 2014, str. 76.

[111] Wang, P.; Xu, B.; Xu, J.; aj.: Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, ročník 174, 2016: s. 806–814.

[112] Wu, C.-H.; Chuang, Z.-J.; Lin, Y.-C.: Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, ročník 5, č. 2, 2006: s. 165–183.

[113] Wu, J.: Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 2017.

[114] Xiao, Y.; Cho, K.: Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*, 2016.

[115] Yang, Y.; Pedersen, J. O.: A comparative study on feature selection in text categorization. In *Icml*, ročník 97, 1997, s. 412–420.

[116] Yogatama, D.; Dyer, C.; Ling, W.; aj.: Generative and Discriminative Text Classification with Recurrent Neural Networks. *Google DeepMind, ArXiv e-prints*, Březen 2017, `1703.01898`.

[117] Zeiler, M. D.: ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[118] Zhang, X.; Zhao, J.; LeCun, Y.: Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 2015, s. 649–657.

Zoznam symbolov, veličín a skratiek

BoW	batôžtek slov – Bag-of-Words
CBOG	kontinuálny batôžtek slov – Continous Bag-of-Words
CNN	konvolučné neurónové siete – Convolutional Neural Network
CRF	Conditional Random Field
DNN	hlboká neurónová sieť – Deep Neural Network
GP	Genetické programovanie
IoT	Internet vecí – Internet of Things
k**-NN**	k najbližších susedov – k Nearest Neighbors
LSTM	Long Short Term Memory
MDF	Minimálna frekvencia v dokumentoch – Minimum Document Frequency
NOTAM	Poznámka pre letcov – Notice to Airmen
RCK	Opakujúce sa jadro – Recurring Kernel
RNN	rekurentné neurónové siete – Recurrent Neural Network
SGD	Stochastický gradientný zostup – Stochastic Gradient Descent
SVM	Metóda podporných vektorov – Support Vector Machines
TF–IDF	Term Frequency – Inverse Document Frequency

Zoznam príloh

A Tabuľky kompletných výsledkov

Tab. A.1: Vývoj úspešnosti klasifikácie na začiatku optimalizácie GP pre český jazyk

Gen.	**Presnosť**			**Gen.**	**Presnosť**		
	poz.	neg.	výsl.		poz.	neg.	výsl.
1	89,48%	88,38%	**88,93%**	41	89,72%	88,32%	89,02%
2	89,49%	88,42%	88,96%	42	89,72%	88,32%	89,02%
3	89,49%	88,42%	88,96%	43	89,72%	88,32%	89,02%
4	89,49%	88,42%	88,96%	44	89,72%	88,32%	89,02%
5	89,49%	88,42%	88,96%	45	89,72%	88,32%	89,02%
6	89,49%	88,42%	88,96%	46	89,70%	88,45%	89,08%
7	89,59%	88,38%	88,99%	47	89,70%	88,45%	89,08%
8	89,59%	88,38%	88,99%	48	89,70%	88,45%	89,08%
9	89,59%	88,38%	88,99%	49	89,70%	88,45%	89,08%
10	89,59%	88,38%	88,99%	50	89,70%	88,45%	89,08%
11	89,59%	88,38%	88,99%	51	89,70%	88,45%	89,08%
12	89,56%	88,47%	89,01%	52	89,70%	88,45%	89,08%
13	89,56%	88,47%	89,01%	53	89,70%	88,45%	89,08%
14	89,56%	88,47%	89,01%	54	89,70%	88,45%	89,08%
15	89,56%	88,47%	89,01%	55	89,70%	88,45%	89,08%
16	89,56%	88,47%	89,01%	56	89,70%	88,45%	89,08%
17	89,57%	88,45%	89,01%	57	89,70%	88,45%	89,08%
18	89,57%	88,45%	89,01%	58	89,70%	88,45%	89,08%
19	89,57%	88,45%	89,01%	59	89,70%	88,45%	89,08%
20	89,57%	88,45%	89,01%	60	89,70%	88,45%	89,08%
21	89,57%	88,54%	89,05%	61	89,70%	88,45%	89,08%
22	89,57%	88,54%	89,05%	62	89,70%	88,45%	89,08%
23	89,57%	88,54%	89,05%	63	89,70%	88,45%	89,08%
24	89,42%	88,46%	88,94%	64	89,70%	88,45%	89,08%
25	89,72%	88,32%	89,02%	65	89,70%	88,45%	89,08%
26	89,72%	88,32%	89,02%	66	89,70%	88,45%	89,08%
27	89,72%	88,32%	89,02%	67	89,70%	88,45%	89,08%
28	89,72%	88,32%	89,02%	68	89,70%	88,45%	89,08%
29	89,72%	88,32%	89,02%	69	89,70%	88,45%	89,08%
30	89,72%	88,32%	89,02%	70	89,70%	88,45%	89,08%
31	89,72%	88,32%	89,02%	71	89,70%	88,45%	89,08%
32	89,72%	88,32%	89,02%	72	89,70%	88,45%	89,08%
33	89,72%	88,32%	89,02%	73	89,70%	88,45%	89,08%
34	89,72%	88,32%	89,02%	74	89,63%	88,35%	88,99%
35	89,72%	88,32%	89,02%	75	89,63%	88,35%	88,99%
36	89,72%	88,32%	89,02%	76	89,63%	88,35%	88,99%
37	89,72%	88,32%	89,02%	77	89,63%	88,35%	88,99%
38	89,72%	88,32%	89,02%	78	89,63%	88,35%	88,99%
39	89,72%	88,32%	89,02%	79	89,63%	88,35%	88,99%
40	89,72%	88,32%	89,02%	80	89,63%	88,35%	88,99%

Tab. A.1 zachytáva vývoj úspešnosti na začiatku optimalizácie pomocou algoritmu GP. Na výsledky sa odvoláva kap. 4.2.1, ktorá porovnáva úspešnosť klasifikácie pred a po optimalizácií GP.

Tab. A.2: Vývoj úspešnosti klasifikácie po 80. gen. optimalizácie GP pre český jazyk

G.	Presnosť			G.	Presnosť			G.	Presnosť		
	poz.	neg.	výsl.		poz.	neg.	výsl.		poz.	neg.	výsl.
81	89,63%	88,35%	88,99%	151	89,76%	88,69%	89,23%	221	90,19%	89,26%	89,72%
82	89,63%	88,35%	88,99%	152	89,72%	88,74%	89,23%	222	90,19%	89,26%	89,72%
83	89,49%	88,52%	89,00%	153	89,72%	88,74%	89,23%	223	90,19%	89,26%	89,72%
84	89,49%	88,52%	89,00%	154	89,72%	88,74%	89,23%	224	90,19%	89,26%	89,72%
85	89,49%	88,52%	89,00%	155	89,72%	88,74%	89,23%	225	90,19%	89,26%	89,72%
86	89,54%	88,51%	89,02%	156	89,72%	88,74%	89,23%	226	90,19%	89,26%	89,72%
87	89,54%	88,51%	89,02%	157	90,02%	88,93%	89,48%	227	90,19%	89,26%	89,72%
88	89,54%	88,51%	89,02%	158	90,02%	88,93%	89,48%	228	90,19%	89,26%	89,72%
89	89,54%	88,51%	89,02%	159	90,02%	88,93%	89,48%	229	90,19%	89,26%	89,72%
90	89,54%	88,51%	89,02%	160	90,10%	88,89%	89,50%	230	90,19%	89,26%	89,72%
91	89,54%	88,51%	89,02%	161	90,10%	88,89%	89,50%	231	90,19%	89,26%	89,72%
92	89,54%	88,51%	89,02%	162	90,10%	88,89%	89,50%	232	90,19%	89,26%	89,72%
93	89,63%	88,43%	89,03%	163	90,10%	88,89%	89,50%	233	90,19%	89,26%	89,72%
94	89,63%	88,43%	89,03%	164	90,10%	88,89%	89,50%	234	90,19%	89,26%	89,72%
95	89,54%	88,33%	88,94%	165	90,10%	88,89%	89,50%	235	90,19%	89,26%	89,72%
96	89,71%	88,37%	89,04%	166	90,10%	88,89%	89,50%	236	90,19%	89,26%	89,72%
97	89,71%	88,37%	89,04%	167	90,10%	88,89%	89,50%	237	90,19%	89,26%	89,72%
98	89,44%	88,54%	88,99%	168	90,10%	88,89%	89,50%	238	90,19%	89,26%	89,72%
99	89,57%	88,45%	89,01%	169	90,10%	88,89%	89,50%	239	90,19%	89,26%	89,72%
100	89,57%	88,45%	89,01%	170	90,10%	88,89%	89,50%	240	90,19%	89,26%	89,72%
101	89,57%	88,45%	89,01%	171	90,10%	88,89%	89,50%	241	90,19%	89,26%	89,72%
102	89,57%	88,45%	89,01%	172	90,10%	88,89%	89,50%	242	90,19%	89,26%	89,72%
103	89,47%	88,56%	89,01%	173	90,10%	88,89%	89,50%	243	90,19%	89,26%	89,72%
104	89,47%	88,56%	89,01%	174	90,10%	88,89%	89,50%	244	90,19%	89,26%	89,72%
105	89,64%	88,45%	89,05%	175	90,10%	88,89%	89,50%	245	90,19%	89,26%	89,72%
106	89,64%	88,45%	89,05%	176	90,10%	88,89%	89,50%	246	90,19%	89,26%	89,72%
107	89,64%	88,45%	89,05%	177	90,05%	89,00%	89,53%	247	90,19%	89,26%	89,72%
108	89,64%	88,45%	89,05%	178	90,05%	89,00%	89,53%	248	90,19%	89,26%	89,72%
109	89,64%	88,45%	89,05%	179	90,05%	89,00%	89,53%	249	90,19%	89,26%	89,72%
110	89,64%	88,45%	89,05%	180	90,01%	89,05%	89,53%	250	90,19%	89,26%	89,72%
111	89,64%	88,45%	89,05%	181	90,01%	89,05%	89,53%	251	90,19%	89,26%	89,72%
112	89,64%	88,45%	89,05%	182	90,01%	89,05%	89,53%	252	90,19%	89,26%	89,72%
113	89,64%	88,45%	89,05%	183	90,01%	89,05%	89,53%	253	90,19%	89,26%	89,72%
114	89,64%	88,45%	89,05%	184	90,01%	89,05%	89,53%	254	90,28%	89,45%	89,87%
115	89,64%	88,45%	89,05%	185	90,01%	89,05%	89,53%	255	90,28%	89,45%	89,87%
116	89,64%	88,45%	89,05%	186	90,01%	89,05%	89,53%	256	90,28%	89,45%	89,87%
117	89,64%	88,45%	89,05%	187	90,01%	89,05%	89,53%	257	90,28%	89,45%	89,87%
118	89,64%	88,45%	89,05%	188	90,01%	89,05%	89,53%	258	90,28%	89,45%	89,87%
119	89,64%	88,45%	89,05%	189	90,01%	89,05%	89,53%	259	90,28%	89,45%	89,87%
120	89,64%	88,45%	89,05%	190	90,01%	89,05%	89,53%	260	90,28%	89,45%	89,87%
121	89,64%	88,45%	89,05%	191	90,01%	89,05%	89,53%	261	90,28%	89,45%	89,87%
122	89,64%	88,45%	89,05%	192	90,01%	89,05%	89,53%	262	90,28%	89,45%	89,87%
123	89,64%	88,45%	89,05%	193	90,01%	89,05%	89,53%	263	90,28%	89,45%	89,87%
124	89,64%	88,45%	89,05%	194	90,01%	89,05%	89,53%	264	90,28%	89,45%	89,87%
125	89,64%	88,45%	89,05%	195	90,01%	89,05%	89,53%	265	90,28%	89,45%	89,87%
126	89,64%	88,45%	89,05%	196	90,01%	89,05%	89,53%	266	90,28%	89,45%	89,87%
127	89,64%	88,45%	89,05%	197	90,01%	89,05%	89,53%	267	90,28%	89,45%	89,87%
128	89,64%	88,45%	89,05%	198	90,01%	89,05%	89,53%	268	90,28%	89,45%	89,87%
129	89,64%	88,45%	89,05%	199	90,01%	89,05%	89,53%	269	90,28%	89,45%	89,87%
130	89,64%	88,45%	89,05%	200	90,01%	89,05%	89,53%	270	90,28%	89,45%	89,87%
131	89,64%	88,45%	89,05%	201	90,01%	89,05%	89,53%	271	90,28%	89,45%	89,87%
132	89,64%	88,45%	89,05%	202	90,01%	89,05%	89,53%	272	90,28%	89,45%	89,87%
133	89,64%	88,53%	89,09%	203	90,01%	89,05%	89,53%	273	90,28%	89,45%	89,87%
134	89,64%	88,53%	89,09%	204	90,01%	89,05%	89,53%	274	90,28%	89,45%	89,87%
135	89,64%	88,53%	89,09%	205	90,01%	89,05%	89,53%	275	90,28%	89,45%	89,87%
136	89,64%	88,53%	89,09%	206	90,01%	89,05%	89,53%	276	90,28%	89,45%	89,87%
137	89,64%	88,53%	89,09%	207	90,01%	89,05%	89,53%	277	90,28%	89,45%	89,87%
138	89,64%	88,53%	89,09%	208	90,16%	89,22%	89,69%	278	90,28%	89,45%	89,87%
139	89,64%	88,53%	89,09%	209	90,16%	89,22%	89,69%	279	90,28%	89,45%	89,87%
140	89,64%	88,53%	89,09%	210	90,18%	89,20%	89,69%	280	90,28%	89,45%	89,87%
141	89,64%	88,53%	89,09%	211	90,18%	89,20%	89,69%	281	90,28%	89,45%	89,87%
142	89,64%	88,53%	89,09%	212	90,19%	89,26%	89,72%	282	90,28%	89,45%	89,87%
143	89,64%	88,53%	89,09%	213	90,19%	89,26%	89,72%	283	90,28%	89,45%	89,87%
144	89,64%	88,53%	89,09%	214	90,19%	89,26%	89,72%	284	90,28%	89,45%	89,87%
145	89,64%	88,53%	89,09%	215	90,19%	89,26%	89,72%	285	90,28%	89,45%	89,87%
146	89,64%	88,53%	89,09%	216	90,19%	89,26%	89,72%	286	90,28%	89,45%	89,87%
147	89,64%	88,53%	89,09%	217	90,19%	89,26%	89,72%	287	90,28%	89,45%	89,87%
148	89,64%	88,53%	89,09%	218	90,19%	89,26%	89,72%	288	90,17%	89,59%	89,88%
149	89,64%	88,53%	89,09%	219	90,19%	89,26%	89,72%	289	90,17%	89,59%	89,88%
150	89,64%	88,53%	89,09%	220	90,19%	89,26%	89,72%	290	90,17%	89,59%	89,88%

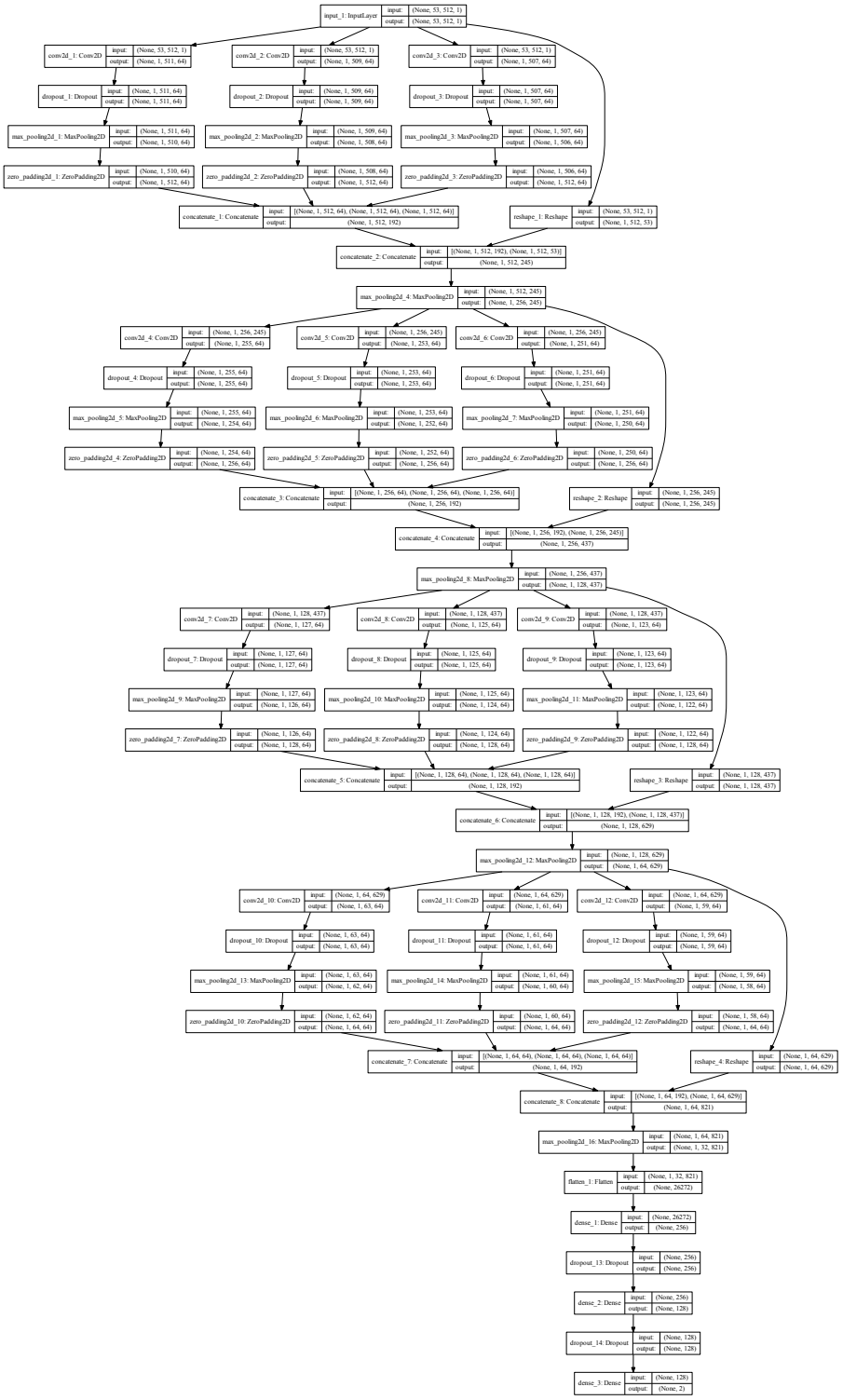
Tab. A.3: Vývoj úspešnosti klasifikácie po 290. gen. optimalizácie GP pre český jazyk

G.	Presnosť			G.	Presnosť			G.	Presnosť		
	poz.	neg.	výsl.		poz.	neg.	výsl.		poz.	neg.	výsl.
291	90,17%	89,59%	89,88%	361	90,33%	89,67%	90,00%	431	90,34%	89,70%	90,02%
292	90,21%	89,56%	89,88%	362	90,33%	89,67%	90,00%	432	90,34%	89,70%	90,02%
293	90,21%	89,56%	89,88%	363	90,33%	89,67%	90,00%	433	90,34%	89,70%	90,02%
294	90,22%	89,59%	89,90%	364	90,33%	89,67%	90,00%	434	90,34%	89,73%	90,03%
295	90,22%	89,59%	89,90%	365	90,33%	89,67%	90,00%	435	90,34%	89,73%	90,03%
296	90,23%	89,60%	89,92%	366	90,33%	89,67%	90,00%	436	90,34%	89,73%	90,03%
297	90,23%	89,60%	89,92%	367	90,34%	89,68%	90,01%	437	90,34%	89,73%	90,03%
298	90,23%	89,60%	89,92%	368	90,34%	89,68%	90,01%	438	90,34%	89,73%	90,04%
299	90,23%	89,60%	89,92%	369	90,34%	89,68%	90,01%	439	90,34%	89,73%	90,04%
300	90,23%	89,60%	89,92%	370	90,34%	89,68%	90,01%	440	90,36%	89,73%	90,04%
301	90,23%	89,60%	89,92%	371	90,34%	89,68%	90,01%	441	90,36%	89,73%	90,04%
302	90,23%	89,60%	89,92%	372	90,34%	89,68%	90,01%	442	90,36%	89,74%	90,05%
303	90,23%	89,60%	89,92%	373	90,34%	89,68%	90,01%	443	90,36%	89,74%	90,05%
304	90,23%	89,60%	89,92%	374	90,34%	89,68%	90,01%	444	90,36%	89,74%	90,05%
305	90,23%	89,62%	89,93%	375	90,34%	89,68%	90,01%	445	90,36%	89,74%	90,05%
306	90,23%	89,62%	89,93%	376	90,34%	89,68%	90,01%	446	90,36%	89,74%	90,05%
307	90,23%	89,62%	89,93%	377	90,34%	89,68%	90,01%	447	90,36%	89,74%	90,05%
308	90,23%	89,62%	89,93%	378	90,34%	89,68%	90,01%	448	90,36%	89,74%	90,05%
309	90,23%	89,62%	89,93%	379	90,34%	89,68%	90,01%	449	90,36%	89,74%	90,05%
310	90,23%	89,62%	89,93%	380	90,34%	89,68%	90,01%	450	90,36%	89,74%	90,05%
311	90,23%	89,62%	89,93%	381	90,34%	89,68%	90,01%	451	90,36%	89,74%	90,05%
312	90,23%	89,62%	89,93%	382	90,35%	89,67%	90,01%	452	90,36%	89,74%	90,05%
313	90,23%	89,62%	89,93%	383	90,35%	89,67%	90,01%	453	90,36%	89,74%	90,05%
314	90,23%	89,62%	89,93%	384	90,35%	89,67%	90,01%	454	90,36%	89,75%	90,05%
315	90,28%	89,60%	89,94%	385	90,35%	89,67%	90,01%	455	90,36%	89,75%	90,05%
316	90,28%	89,60%	89,94%	386	90,35%	89,67%	90,01%	456	90,36%	89,75%	90,05%
317	90,26%	89,63%	89,95%	387	90,35%	89,67%	90,01%	457	90,36%	89,75%	90,05%
318	90,26%	89,67%	89,97%	388	90,35%	89,67%	90,01%	458	90,36%	89,75%	90,05%
319	90,26%	89,67%	89,97%	389	90,36%	89,67%	90,01%	459	90,37%	89,75%	90,06%
320	90,31%	89,64%	89,97%	390	90,36%	89,67%	90,01%	460	90,37%	89,75%	90,06%
321	90,31%	89,64%	89,97%	391	90,36%	89,67%	90,01%	461	90,36%	89,76%	90,06%
322	90,31%	89,64%	89,97%	392	90,36%	89,67%	90,01%	462	90,36%	89,76%	90,06%
323	90,31%	89,64%	89,97%	393	90,36%	89,67%	90,01%	463	90,36%	89,76%	90,06%
324	90,31%	89,64%	89,97%	394	90,32%	89,72%	90,02%	464	90,36%	89,76%	90,06%
325	90,31%	89,64%	89,97%	395	90,33%	89,71%	90,02%	465	90,36%	89,76%	90,06%
326	90,31%	89,64%	89,97%	396	90,33%	89,71%	90,02%	466	90,36%	89,76%	90,06%
327	90,31%	89,64%	89,97%	397	90,32%	89,72%	90,02%	467	90,36%	89,76%	90,06%
328	90,31%	89,64%	89,97%	398	90,33%	89,71%	90,02%	468	90,36%	89,76%	90,06%
329	90,31%	89,64%	89,97%	399	90,32%	89,72%	90,02%	469	90,36%	89,76%	90,06%
330	90,31%	89,64%	89,97%	400	90,34%	89,70%	90,02%	470	90,36%	89,76%	90,06%
331	90,31%	89,64%	89,97%	401	90,34%	89,70%	90,02%	471	90,36%	89,76%	90,06%
332	90,31%	89,64%	89,97%	402	90,34%	89,70%	90,02%	472	90,36%	89,76%	90,06%
333	90,31%	89,64%	89,97%	403	90,34%	89,70%	90,02%	473	90,36%	89,75%	90,05%
334	90,31%	89,64%	89,97%	404	90,34%	89,70%	90,02%	474	90,37%	89,74%	90,06%
335	90,31%	89,64%	89,97%	405	90,34%	89,70%	90,02%	475	90,36%	89,75%	90,05%
336	90,31%	89,64%	89,97%	406	90,34%	89,70%	90,02%	476	90,36%	89,75%	90,05%
337	90,31%	89,64%	89,97%	407	90,34%	89,70%	90,02%	477	90,36%	89,75%	90,05%
338	90,31%	89,64%	89,97%	408	90,34%	89,70%	90,02%	478	90,36%	89,75%	90,06%
339	90,30%	89,65%	89,98%	409	90,34%	89,70%	90,02%	479	90,37%	89,75%	90,06%
340	90,30%	89,65%	89,98%	410	90,34%	89,70%	90,02%	480	90,37%	89,75%	90,06%
341	90,32%	89,65%	89,98%	411	90,34%	89,70%	90,02%	481	90,37%	89,75%	90,06%
342	90,32%	89,65%	89,98%	412	90,34%	89,70%	90,02%	482	90,37%	89,75%	90,06%
343	90,31%	89,63%	89,97%	413	90,34%	89,70%	90,02%	483	90,37%	89,75%	90,06%
344	90,31%	89,63%	89,97%	414	90,34%	89,70%	90,02%	484	90,37%	89,75%	90,06%
345	90,31%	89,63%	89,97%	415	90,34%	89,70%	90,02%	485	90,37%	89,75%	90,06%
346	90,31%	89,63%	89,97%	416	90,34%	89,70%	90,02%	486	90,37%	89,75%	90,06%
347	90,31%	89,64%	89,98%	417	90,34%	89,70%	90,02%	487	90,37%	89,75%	90,06%
348	90,31%	89,64%	89,98%	418	90,34%	89,70%	90,02%	488	90,37%	89,75%	90,06%
349	90,31%	89,64%	89,98%	419	90,34%	89,70%	90,02%	489	90,37%	89,75%	90,06%
350	90,31%	89,64%	89,98%	420	90,34%	89,70%	90,02%	490	90,37%	89,75%	90,06%
351	90,31%	89,64%	89,98%	421	90,34%	89,70%	90,02%	491	90,38%	89,75%	90,06%
352	90,31%	89,64%	89,98%	422	90,34%	89,70%	90,02%	492	90,38%	89,75%	90,07%
353	90,31%	89,64%	89,98%	423	90,34%	89,70%	90,02%	493	90,38%	89,75%	90,07%
354	90,31%	89,64%	89,98%	424	90,34%	89,70%	90,02%	494	90,38%	89,76%	90,07%
355	90,31%	89,64%	89,98%	425	90,34%	89,70%	90,02%	495	90,37%	89,78%	90,08%
356	90,33%	89,67%	90,00%	426	90,34%	89,70%	90,02%	496	90,37%	89,78%	90,08%
357	90,33%	89,67%	90,00%	427	90,34%	89,70%	90,02%	497	90,37%	89,78%	90,08%
358	90,33%	89,67%	90,00%	428	90,34%	89,70%	90,02%	498	90,37%	89,78%	90,08%
359	90,33%	89,67%	90,00%	429	90,34%	89,70%	90,02%	499	90,37%	89,78%	90,08%
360	90,33%	89,67%	90,00%	430	90,34%	89,70%	90,02%	500	90,38%	89,78%	90,08%

Tab. A.4: Vývoj úspešnosti klasifikácie na konci optimalizácie GP pre český jazyk

Gen.	**Presnosť**		
	poz.	neg.	výsl.
501	90,37%	89,78%	90,08%
502	90,38%	89,78%	90,08%
503	90,39%	89,78%	90,08%
504	90,39%	89,78%	**90,09%**
505	90,39%	89,78%	90,09%
506	90,39%	89,78%	90,09%
507	90,39%	89,78%	90,09%
508	90,39%	89,78%	90,09%
509	90,39%	89,78%	90,09%

B Štruktúra hlbokej neurónovej siete



Obr. B.1: Štruktúra neurónovej siete č. 6

Publikácie autora

Publikácie v časopisoch s impakt faktorom

- ([87]) Povoda, L.; Burget, R.; Mašek, J.; Uher, V.; Dutta, M. K.: Optimization Methods in Emotion Recognition System. *Radioengineering*, ročník 25, č. 3, 2016: s. 565–572.
  (IF pre rok 2016 = 0,945, podiel 59%)
- Povoda, L.; Burget, R.; Travieso, C. M.: Grammarless and Language-Independent Text Sentiment Analysis Based on Recurring Kernels. *Cognitive Computation*, 2018, **(ku dňu 23.8.2018 v recenznom konaní)**.
  (IF pre rok 2016 = 3,441, podiel 79%)

Publikácie v časopisoch bez impakt faktoru

- ([65]) Masek, J.; Burget, R.; Povoda, L.; Dutta, M. K.: Multi–GPU Implementation of Machine Learning Algorithm using CUDA and OpenCL. *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, ročník 5, č. 2, 2016: s. 101–107.
  (podiel 10%)
- ([85]) Povoda, L.; Burget, R.; Mašek, J.; Cvrk, L.: Automatické rozpoznávanie emócií z českého textu pomocou umelej inteligencie. In *Elektrorevue - Internetový časopis (http://www.elektrorevue.cz)*, 17.1, 2015, s. 15–18.
  (podiel 79%)

Publikácie v konferenčných zborníkoch

- Rajnoha, M.; Povoda, L.; Mašek, J.; Burget, R.; Dutta, M. K.: Pedestrian Detection from Low Resolution Public Cameras in the Wild. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. New Delhi, India: 2018. s. 1-5. ISBN: 978-1-5386-3044-0.
  (podiel 13%)
- ([84]) Povoda, L.; Burget, R.; Dutta, M. K.; Sengar, N.: Genetic Optimization of Big Data Sentiment Analysis. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2017, s. 141–144.
  (podiel 88%)
- Mehta, G.; Dutta, M. K.; Burget, R.; Povoda, L. Biometric Data Security Using Fractional Fourier Transform and Chaotic Theory. In *Proceedings of the 39th International Conference on Telecommunication and Signal Processing,*

TSP 2016. Vídeň: 2016. s. 533-537. ISBN: 978-1-5090-1287-9.
(podiel 73%)

- ([83]) Povoda, L.; Burget, R.; Dutta, M. K.: Sentiment analysis based on Support Vector Machine and Big Data. In *Telecommunications and Signal Processing (TSP), 2016 39th International Conference on*, IEEE, 2016, s. 543–545.
(podiel 79%)

- Mašek, J.; Burget, R.; Povoda, L.; Harvánek, M.: Image Search Using Similarity Measures Based on Circular Sectors. In *Computer Science & Information Technology (CS & IT)*. Dubaj, Spojené arabské emiráty: 2015. s. 241-251. ISBN: 978-1-921987-43- 4. ISSN: 2231- 5403.
(podiel 10%)

- ([82]) Povoda, L.; Arora, A.; Singh, S.; Burget, R.; Dutta, M. K.: Emotion Recognition from Helpdesk Messages. In *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2015, s. 310–313.
(podiel 72%)

- Sengar, N.; Dutta, M. K.; Burget, R.; Povoda, L.: Detection of Diabetic Macular Edema in Retinal Images Using a Region Based Method. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*. 2015. s. 412-415. ISBN: 978-1-4799-8498-5.
(podiel 73%)

- Povoda, L.; Burget, R.; Mašek, J.; Dutta, M. K.: Job Shop Scheduling Problem with Heuristic Genetic Programming Operators. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*. Noida, Delhi-NCR, India: 2015. s. 702-707. ISBN: 978-1-4799-5990-7.
(podiel 64%)

- Povoda, L.; Burget, R.; Karásek, J.; Dutta, M. K.; Singh, A.: Genetic Programming Operators for Work- Flow Optimization in Logistic Distribution Centers. In *2013 Sixth International Conference on Contemporary Computing (IC3-2013)*. 2014. s. 151-155. ISBN: 978-1-4799-0190-6.
(podiel 73%)

- Karásek, J.; Burget, R.; Povoda, L. Logistic Warehouse Process Optimization through Genetic Programming Algorithm. In *Advances in Intelligent Systems and Computing, Modern Trends and Techniques in Computer Science*. 285. Springer, 2014. s. 29-40. ISBN: 978-3-319-06739-1.
(podiel 10%)

Vyvinutý software

- Povoda, L.; Burget, R.: NOTAM detektor 1.0; Software pro detekci složitých oblastí v NOTAM zprávách. Ústav telekomunikací, FEKT VUT v Brně.
  URL: http://splab.cz/download/software/software-pro-detekci-slozitych-oblasti-v-notam-zpravach.
  (podiel 80%)
- Harár, P.; Burget, R.; Povoda, L.; Rajnoha, M.: Kex-library; KEX - Software library for easier management and result interpretation of Keras experiments. Kancelář SE5.123 (Harár).
  URL: http://splab.cz/download/software/kex-library.
  (podiel 3%)
- BURGET, R.; POVODA, L.: NOTAM - Honeywell; Data Analytics Tools. Honeywell.
  URL: http://splab.cz/download/software/software-pro-detekci-slozitych-oblasti-v-notam-zpravach.
  (podiel 50%)

Curriculum Vitæ

Lukáš Povoda

Osobné informácie

Dátum narodenia:	18. 1. 1990
Miesto narodenia:	Bojnice
Telefón:	+420 732 252 572
E-mail:	lupo112@gmail.com

Vzdelanie

2014 – 2018	Doktorské štúdium – Teleinformatika Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Technická 3058/10, 616 00 Brno titul: Ph.D. (predpokladaný rok ukončenia: 2018)
2012 – 2014	Magisterské štúdium – Telekomunikační a informační technika Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Technická 3058/10, 616 00 Brno titul: Ing.
2009 – 2012	Bakalárske štúdium – Teleinformatika Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Technická 3058/10, 616 00 Brno titul: Bc.
2005 – 2009	Úplné stredné odborné vzdelanie – Elektrotechnika Stredná odborná škola Handlová, Lipová 8, 972 51 Handlová

Zamestnanie

2013 – súčasnosť	Vedecký pracovník Centra senzorických informačních a komunikačních systémů Vysoké učení technické v Brně, Technická 3058/10, 616 00 Brno
2013 – súčasnosť	Senior PHP Developer a-net group, a.s., Národních hrdinů 3, 190 00 Praha 9

Spolupráca na projektoch

2017 – 2020	Expertní systém pro automatickou analýzu a řízení big data úložišť výrobních společností MPO reg. č. FV20044
2017 – 2019	Hardwarově akcelerovaná identifikace osob z videozáznamů s použitím hlubokého strojového učení MVČR reg. č. VI2VS/554
2016 – 2016	Connected homes - thermostat datamining hospodárska zmluva (750 669 CZK), referenčná osoba Ondřej Lipták - Leader of electronic design group at Honeywell, Brno Hardware Center of Excellence
2016 – 2016	Klasifikace hutních materiálů pomocí spektrometrie laserem indukovaného plazmatu za použití hlubokých neuronových sítí IGA reg. č. FEKT/FSI-J-16-3564
2015 – 2016	Pokročilé meteorologické informace pro letectví TAČR reg. č. TH01010503
2014 – 2015	Výzkum a vývoj technologie pro detekci emocí v nestrukturovaných datech MPO reg. č. FR-TI4/151

Publikačná aktivita

- Publikácie v časopisoch s impakt faktorom: 1 + 1 (v recenznom konaní)
- Publikácie v časopisoch bez impakt faktoru: 2
- Publikácie v konferenčných zborníkoch: 10
- Software: 3
- Príspevky indexované v databáze WoS: 9
- Príspevky indexované v databáze Scopus: 9
- H-index podľa databáze WoS: 2
- H-index podľa databáze Scopus: 3