VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODHAD EMOCÍ Z TEXTU

ESTIMATION OF EMOTIONS FROM A TEXT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ANETA DUFKOVÁ**
AUTHOR

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

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2019/2020

Zadání bakalářské práce

Studentka: **Dufková Aneta**
Program: Informační technologie
Název: **Odhad emocí z textu**
Estimation of Emotions from a Text
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Seznamte se se základy zpracování textu, NLP a strojového učení.
2. Nastudujte postupy pro odhad emocí z textu pomocí strojového učení. Najděte vhodné datové sady. Proveďte rozdělení na trénovací a testovací subsety.
3. Navrhněte a implementujte zvolenou metodu pro odhad emocí z textu. Vyhodnoťte dosaženou úspěšnost.
4. Navrženou metodu zdokonalte.
5. Diskutujte dosažené cíle a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 ze zadání.

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

Vedoucí práce: **Szőke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 1. listopadu 2019

Abstrakt

Tato práce popisuje proces odhadování emocí z textu, při němž je využíváno strojové učení. Proces začíná průzkumem používaných metod, pokračuje výběrem vhodné metody a experimentováním. Využívá několik datových sad, kombinuje je a zkouší různé techniky předzpracování textu. Závěrem je webové rozhraní, které využívá předtrénovaný model a umožňuje detekovat emoce z příspěvků z Twitteru.

Abstract

This thesis describes a process of estimation of emotions from a text using machine learning. The process starts with research of existing methods, continues with choosing a suitable method and experimenting. It uses several datasets, combines them and tests different techniques of text preprocessing. The result is a web interface which uses the pretrained model and allows to estimate emotions from Twitter posts.

Klíčová slova

odhad emocí, analýza sentimentu, dolování z textu, předzpracování textu, zpracování přirozeného jazyka

Keywords

estimation of emotions, sentiment analysis, text mining, text preprocessing, natural language processing

Citace

DUFKOVÁ, Aneta. *Odhad emocí z textu*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Odhad emocí z textu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Igora Szőkeho, Ph.D. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Aneta Dufková
26. května 2020

Poděkování

Poděkování si zaslouží v prvé řadě vedoucí mé práce, Ing. Igor Szőke, Ph.D., protože jsem z každé konzultace s ním odcházela nejen s hlavou plnou myšlenek, ale také s novou chutí pokračovat v práci. Dále bych ráda poděkovala Ing. Ondřeji Novotnému, jenž mi dal cenné rady do začátku a poskytl data pro trénování. Velmi děkuji také Ing. Sebastiánovi Poliakovi za nespočet nápadů a trpělivé zodpovídání všech mých dotazů.

Obsah

Kapitola 1

Úvod

Odhad emocí z textu, jinými slovy analýza sentimentu, je v současné době populárním problémem. Firmy chtějí vědět, co si veřejnost myslí o jejich produktech a službách, dělají se předvolební průzkumy, analyzují se komentáře turistů hodnotících hotely apod.

Místem, kde lidé ventilují své názory, je internet a zejména pak sociální sítě. Právě odsud se dá získat velké množství dat, analyzovat je a dojít k výsledku, co si společnost myslí o daném tématu. Zpočátku by se tato úloha mohla zdát jednoduchá, ale ne z každého textu je jasná jedna konkrétní emoce. Někdy má s jednoznačným určením sentimentu problém i člověk, natož stroj. Když se přidají ještě sarkasmus, ironie a jiné jazykové prostředky, stává se analýza sentimentu komplexním problémem, na jehož řešení se stále pracuje.

V této bakalářské práci se zabývám odhadem emocí z textu pomocí neuronové sítě, zaměřuji se konkrétně na analýzu sentimentu z tweetů, rozeznávání negativní, neutrální a pozitivní emoce. Experimentovala jsem s dvěma typy neuronových sítí, vyzkoušela různé techniky předzpracování textu, podle potřeby měnila a rozšiřovala také získaná trénovací data.

Začala jsem průzkumem technik používaných pro klasifikací sentimentu v kapitole 2, pokračuji kapitolou 3, jež pojednává o zpracování textu. Kapitola 4 se zabývá základy strojového učení.

Následuje implementační část. Prvním krokem je nalezení a výběr vhodných datových sad, o nichž se zmiňuje kapitola 5. Ladění parametrů, zkoušení různých technik zpracování textu a kombinování dat popisuje kapitola 6 věnovaná experimentům. Samotnou implementaci pokrývá kapitola 7.

Od samého začátku, výběrem trénovacích dat a následně i modelu, jsem směřovala ke klasifikování tweetů online. Cílem bylo vytvořit nástroj použitelný pro širokou veřejnost, který by stahoval příspěvky z Twitteru na zadané téma a dokázal z nich určovat, jaký názor má na téma společnost. Na závěr jsem proto natrénovaný klasifikátor implementovala do webové aplikace WhatDoesTwitterThink.com, o níž se pojednává v kapitole 7.

Kapitola 2

Analýza sentimentu

Tato bakalářská práce se zabývá odhadem emocí, tj. analýzou sentimentu, což je významná oblast, která spojuje obor zpracování přirozeného jazyka (dále jako NLP, Natural Language Processing) s technikami strojového učení.

Analýza sentimentu bývá často nazývána také dolováním názorů. Jejím cílem je zjistit postoj, názor, emoci, posudek či přístup autora k nějaké entitě, případně celkovou polaritu dokumentu. Entitou je myšlen produkt, služba, organizace, událost, problém či téma [26]. Může být detekován buď emoční stav autora během psaní příspěvku, nebo efekt, jaký si přeje mít autor příspěvku na čtenáře [28].

2.1 Typy analýzy sentimentu

Analýza sentimentu se využívá pro nejrůznější účely a podle toho se pak implementuje její konkrétní typ. Většinou stačí vyhodnocovat polaritu (pozitivní, negativní nebo neutrální), jindy se ovšem hodnotí více emocí. Podle toho rozeznáváme typy analýz [17]:

- Standardní analýza sentimentu je nejoblíbenější a klasifikuje do tří tříd: pozitivní, negativní, neutrální.
- Jemnější analýza přidává více kategorií, zpravidla jich bývá pět: velmi pozitivní, pozitivní, neutrální, negativní, velmi negativní.
- Detekce konkrétních emocí umí poznat naštvání, štěstí, frustraci apod.
- Analýza založená na aspektech se zaměřuje také na to, o čem je v textu vyjadřován názor. Například z věty „Aplikace má velmi čisté a uživatelsky přívětivé rozhraní.“ dokáže zjistit, že jde o pozitivní hodnocení uživatelského rozhraní.

2.2 Využití analýzy sentimentu

Analýza emocí má celou řadu využití, například:

Reputace firmy

Lidé na internetu sdílí své zkušenosti a doporučení na nejrůznější značky, služby a produkty. Píšou názory na sociální sítě, blogy i do diskuzních fór. Pokud se podaří tyto příspěvky analyzovat, firma si může udělat dobrou představu o tom, jaký názor na ni nebo na její produkty má veřejnost.

Výběr nejlepšího produktu

Analýzu emocí z recenzí a komentářů uživatelů využije i běžný člověk, který si chce něco koupit. Místo toho, aby si musel zdlouhavě pročítat názory ostatních, automatická detekce sentimentu mu prozradí, zda se lidem produkt líbí, nebo ne.

Politika

Analýza sentimentu se využívá i v politice, například před volbami se dělá průzkum, jaké jsou názory na jednotlivé kandidáty.

Výzkum

V mnoha výzkumech se hodí získat názory uživatelů, které prezentují na sociálních sítích, v diskuzích a jinde na webu. Na základě detekce emocí z e-mailů se například zkoumalo, jak se liší emoce jednotlivých pohlaví [26].

2.3 Metody klasifikace sentimentu

Pro klasifikaci sentimentu se používají následující přístupy [43]:

Strojové učení

Pro techniku strojového učení se z textu získají příznaky, vybuduje se klasifikátor a ten se naučí rozpoznávat emoce. Dnes je tato metoda patrně nejpoužívanější, vyžaduje ovšem velké množství anotovaných dat, ze kterých se systém učí. Udávaná přesnost je 60-80 % [43].

Rozpoznávání na bázi slovníku

Tato metoda používá obsáhlý slovník slov, která jsou ohodnocena jako pozitivní či negativní. Podle slov použitých v dané větě či příspěvku se pak vyhodnotí, zda převládají negativní, nebo pozitivní výrazy. Varianta sice nevyžaduje žádná data k trénování, ale zato je zapotřebí obsáhlý slovník. Navíc existuje velká šance, že v datech ke zhodnocení budou i slova, jež nejsou ve slovníku obsažená. Udávaná přesnost činí zhruba 65 %.

Hybridní přístup

Někdy se používá i kombinace výše zmíněných přístupů. Například týmu z HP laboratoří rozpoznávání na bázi slovníku dávalo lepší výsledky (78% přesnost) než na základě strojového učení (68% přesnost) [46]. Po kombinaci obou metod se dostali na přesnost 85 %. Slovník ovšem speciálně obohatili o některé výrazy často používané na Twitteru a přidali ještě několik pravidel, která ošetřují věty obsahující porovnávání („iPhone je lepší než Huawei“) nebo věty, kde jedno negativní slovo mění význam pozitivního. Příkladem druhého zmiňovaného případu je věta „Můj problém pomalu vymizel.“. Slovo „problém“ je běžně negativní, nicméně spojení se slovem „vymizel“ větu dělá pozitivní. Standardní používaný výpočet, který počítá sumu slov, za každé pozitivní přičítá 1 a za každé negativní odčítá 1, by to neodhalil.

Tato práce se zabývá rozpoznáváním emocí z textu za použití strojového učení, v jednom experimentu popsaném v kapitole 6.3 zkouší hybridní přístup.

2.4 Přístup k analýze sentimentu v této práci

Cílem práce je zprovoznit natrénovaný klasifikátor ve webovém prostředí a klasifikovat tweety. Výsledná aplikace má být minimalistická, proto nerozeznává detailní emoce jako frustraci či štěstí. Navíc nedává smysl rozlišovat spoustu emocí, protože většina kategorií by stejně zůstávala prázdná. Vývojářský účet na Twitteru totiž ve verzi zdarma umožňuje stahovat pouze omezené množství tweetů, takže není možné analyzovat velký vzorek dat. Z toho důvodu byla zvolena pouze standardní analýza rozeznávající tři třídy. Analýza založená na aspektech je zbytečná, protože už budou analyzovány tweety pojednávající o konkrétním tématu.

Z metod klasifikace sentimentu byla zvolena technika strojového učení, protože je v dnešní době nejpoužívanější.

Výsledná aplikace je schopná analyzovat libovolná témata, má sloužit široké veřejnosti zejména pro zajímavost a pro zábavu.

Kapitola 3

Zpracování textu

Pro pochopení této práce je nejdříve potřeba vysvětlit některé teoretické pojmy a podívat se na oblasti, jichž se dotýká. Těmito oblastmi jsou zejména zpracování textu a strojové učení, o němž pojednává kapitola 4.

Zpracování textu je podoblastí zpracování přirozeného jazyka (NLP). Tento obor se v posledních letech hodně zkoumá, jelikož jazyk jako dorozumívací prostředek je pro nás velmi důležitý. NLP zároveň propojuje informatiku s lingvistikou, vývoj některých systémů navíc vyžaduje i znalosti z dalších oborů, jako jsou statistika, psychologie či společenské vědy [23].

Při zpracování textu v kontextu této práce se bavíme o textu v elektronické formě, vždy tedy přesně víme, jaké znaky analyzujeme. Elektronický text s sebou ovšem stále nese některé známé problémy, jako jsou například zkratky, slangové výrazy nebo sarkasmus.

3.1 Předzpracování textu

Než se s textem vůbec začne pracovat, je dobré ho převést do formy, která je vhodná pro další zpracování. Vhodné je vymazat věci, jež jsou nepodstatné, změnit tvary slov apod. Níže jsou rozebrané nejčastěji používané techniky (podle [20] a [19]).

Odstranění šumu

Odstranění šumu znamená odstranění znaků, které nemají pro daný úkol význam. Co je třeba odstranit, vždy záleží na konkrétních datech. Pokud je zpracováván HTML text, je vhodné odstranit HTML značky. Při zpracovávání příspěvků ze sociální sítě Twitter je možné odstranit například URL adresy a označení jiných uživatelů. Proč je odstranění šumu tak důležité, je ilustrováno v části o stematizaci.

Převod na malá písmena

Nejjednodušší, ale velmi užitečný typ předzpracování je převod na malá písmena. Je vhodný nejen pro NLP, používá se i v jiných oborech. Například při vyhledávání v databázi je dobré, aby se při dotazu „usa“ zobrazily i záznamy obsahující „USA“ velkými písmeny.

Existují ovšem úlohy, kdy by převedení všech písmen na malá způsobilo ztrátu informace. Například „Kovář“ s velkým K může mít úplně jiný kontext (příjmení) než „kovář“ s malým k (povolání). Vždy tedy záleží na konkrétní úloze a datech, s nimiž se pracuje.

Normalizace textu

Normalizace je převedení slova do jeho standardní formy. Například slova „anooo“ a „áno“ mohou být převedena do podoby „ano“. Podobně slovo „o5“ by se převedlo na „opět“.

Normalizace je důležitá zejména u textů, jako jsou komentáře a příspěvky ze sociálních sítí, uživatelské recenze, textové zprávy a jiné formy textu, u nichž je velká pravděpodobnost překlepů a používání hovorových výrazů.

Bohužel neexistuje žádný univerzální způsob, jak normalizovat text [20]. Vhodné je například použít na texty nástroj na kontrolu pravopisu, ten může pomoci normalizovat aspoň překlepy a pravopisné chyby.

Odstranění stop slov

Stop slova jsou slova, která jsou v jazyce často používaná, ale nenesou velký význam, typicky jde o spojky, předložky apod. Většinou tvoří 20-30 % textu [19]. Příkladem českých stop slov jsou například „ani“, „když“, „nic“, „proto“, „své“.

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your',
'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "i
t's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'tha
t', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'havi
ng', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'o
f', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'abov
e', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'onc
e', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'so
me', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just',
'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'could
n', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn',
"isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Obrázek 3.1: Stop slova, která nenesou velký význam, pro anglický jazyk z knihovny NLTK.

U některých úloh může odstranění stop slov vadit. V jiných případech je také vhodné definovat vlastní stop slova podle obsahu textu, s nímž je pracováno. Vezměme si jako příklad větu „Řekl ti, že nebyla šťastná“. Po odstranění stop slov dle výše uvedeného seznamu by z ní zbylo „řekl šťastná“. Při určování emocí z textu bychom tuto „větu“ vyhodnotili jako pozitivní, původní věta ovšem pozitivní nebyla. U jiných úloh NLP naopak prospěje, když jsou stop slova odstraněná. Například je-li cílem zjistit téma textu, slova jako „ne“, „ano“ nebo „já“ v tom moc nepomohou [44].

Stop slova je možné odstranit, nebo pro zachování informace nahradit jedním obecným slovem. V obou případech se sníží velikost slovníku, což urychlí následné zpracování.

Alternativní možností, kterou umožňuje například knihovna scikit-learn [2], je odstranit slova, která se vyskytují v určitém procentu analyzovaných textů. Tento proces má podobný efekt jako odstranění stop slov.

Stematizace

Stematizace převádí slova na jejich kmen tím, že odstraní morfologické koncovky a případně předpony. Příkladem jsou slova „connection“, „connecting“ a „connected“, která budou všechna přeměněna na stem „connect“. Někdy může ovšem stematizace napáchat i škodu, problémem jsou zejména dva následující případy [12]:

- nesouvisející slova jsou zkrácena na stejný základ (ledový a leda vedou obě na led),
- související slova jsou zkrácena na jiný základ (to se týká slov, která při ohýbání mění kořen, jako je například dítě/děti).

Původní slovo (A)	Po normalizaci (B)	Stematizace A	Stematizace B
...trouble...	trouble	...trouble...	troubl
trouble<	trouble	trouble	troubl
trouble!	trouble	trouble!	troubl
<a>trouble	trouble	<a>trouble	troubl
1.trouble	trouble	1.troubl	troubl

Tabulka 3.1: Rozdílné výsledky stematizace v souvislosti s normalizací. Liší podle toho, zda byla (B), nebo nebyla (A) provedena normalizace. Převzato z [12].

Lematizace

Lematizace je velmi podobná stematizaci, nicméně jejím výsledkem není kmen, nýbrž základní (slovníkový) tvar slova (tj. lemma). Lematizace nefunguje tak, že by pouze odřízla předponu a příponu, je přesnější. Slovo „lepší“ by převedla do podoby slova „dobrý“, protože to je jeho základní forma. Většinou se k tomu používá slovník, kvalita lematizace je tedy závislá na velikosti tohoto slovníku. Když se objeví slova, která se ve slovníku nevyskytují, použije se pro ně algoritmický stemmer. Používá se buď lematizace, nebo stematizace.

3.2 Jak se reprezentuje text

Aby bylo možné s textem pracovat, zkoumat jej a dále zpracovávat, je potřeba jej převést do vhodné reprezentace. Přístupů, které se k tomu používají, existuje několik. Spočívají v tom, že se slova převedou na nějakou číselnou reprezentaci, typicky vektor v mnoharozměrném prostoru. V praxi se jedná o desítky až stovky rozměrů, například níže zmíněný Word2Vec využívá u předtrénovaného modelu 300 rozměrů.

One-hot kódování

Nejjednodušším přístupem k reprezentaci textu je tzv. one-hot kódování. Funguje tak, že vezme celý slovník (všechna slova, která se v textu vyskytují) a naskládá je za sebe. Každá věta (či jiný úsek textu) je pak reprezentována vektorem těchto slov, kde 1 značí, že obsahuje dané slovo, a 0 označuje, že jej neobsahuje [40]. Princip této reprezentace je možné ukázat na dvojici vět:

1. Máma vaří oběd v kuchyni.
2. Máma vaří oběd tam, kde ho vaří vždy.

Nejdříve se stanoví slovník, tedy vezmou se slova: máma, vaří, oběd, v, kuchyni, tam, kde, ho, vždycky. V tomto příkladě neuvažujeme odstranění stop slov. Pak se jen vyjádří, která slova ve větě jsou, a která nikoliv.

Bag Of Words

Na podobném principu funguje tzv. Bag Of Words. Tento model říká, jaká slova se v textu vyskytují a kolikrát, nicméně nezohledňuje, kde je slovo umístěné, a nebere v úvahu, jak si jsou jednotlivá slova podobná. Princip demonstruje tabulka 3.2. V praxi je vhodné počet slov normalizovat celkovým počtem slov, aby byla reprezentace nezávislá na délce textu.

	Máma vaří oběd v kuchyni.		Máma vaří oběd tam, kde ho vaří vždy.	
	One-hot	Bag Of Words	One-hot	Bag Of Words
máma	1	1	1	1
vaří	1	1	1	2
oběd	1	1	1	1
v	1	1	0	0
kuchyni	1	1	0	0
tam	0	0	1	1
kde	0	0	1	1
ho	0	0	1	1
vždy	0	0	1	1

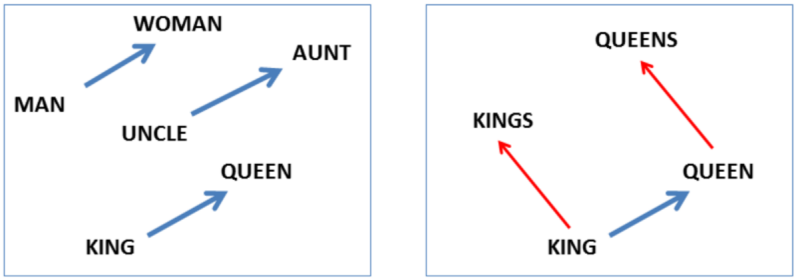
Tabulka 3.2: Tabulka srovnává reprezentaci dvou stejných vět v One-hot kódování a Bag Of Words. One-hot kódování zobrazuje pouze to, zda se dané slovo ve větě vyskytuje, či nikoliv. Bag Of Words přidává informaci o tom, kolikrát se vyskytuje.

BOW je základní a naivní model, i přesto se v NLP využívá často, a to zejména v úlohách klasifikace textu. Jeho síla spočívá v jednoduchosti. Je výpočetně nenáročný a hodí se pro typy úloh, kdy není pozice slov v textu důležitá [47].

One-hot kódování a Bag Of Words jsou základní způsoby vyjádření textu, ovšem vedou na reprezentaci s vysokou dimenzionalitou, proto v této práci používány. Za zmínku stojí ještě TF-IDF (Term Frequency-Inverse Document Frequency) – způsob reprezentace textu, který je schopný vyjádřit četnost slova v dané větě či dokumentu. Pro tuto práci ovšem není relevantní, hodí se spíš pro úkoly, kdy je cílem zjistit téma, o němž text pojednává.

Word2vec

Patrně největší revoluci ve světě zpracování přirozeného jazyka způsobil nástroj známý pod názvem Word2vec, za nímž stojí tým vědců reprezentovaný Tomášem Mikolovem [31]. Reprezentace slov získané tímto nástrojem umožňují provádět se slovy matematické operace.

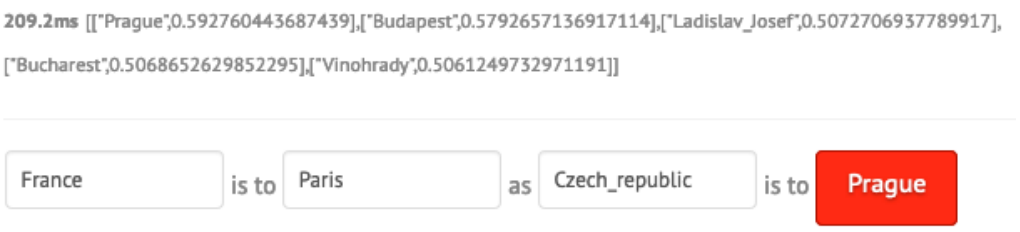


Obrázek 3.2: Levý panel zobrazuje vztahy mezi pohlavím. Pravý panel přidává ještě vztah mezi jednotným a množným číslem. Obrázek převzat z [31].

Dokáže tedy modelovat vztahy, jak zobrazuje následující příklad:

vektor(král) – vektor(muž) + vektor(žena) = vektor velmi blízký k vektor(královna)

Na základě těchto výpočtů je možné například odpovídat na otázky. Odpověď na otázku, jaké je hlavní město České republiky, dostaneme z výpočtu vektor(Paříž) – vektor(Francie) + vektor(Česká republika). Výsledkem bude vektor velmi blízký vektor(Praha).



Obrázek 3.3: Ukázka z aplikace Radima Řehůřka [48], která zobrazuje vztahy slov v reprezentaci Word2vec. Slova, která patří k sobě, se spojují znakem podtržítka _. Radim Řehůřek je původním autorem knihovny Gensim [37], která je v práci použita.

Vznikla další řada nástrojů podobných Word2vec, nejznámějším příkladem je pravděpodobně stanfordský GloVe. Není ovšem tolik známý a rozšířený, proto se jím tato práce nezabývala.

Na Word2vec navazují další populární modely. Jeden z nich vyvinul Tomáš Mikolov se svým týmem, je znám pod jménem FastText. Velkou revoluci v oblasti NLP způsobil i framework BERT z dílny Googlu zveřejněný v roce 2018. Hlavní rozdíl, kterým se BERT liší od Word2vec, je skutečnost, že si poradí s homonymy, tedy slovy s mnohačetným významem [38]. Oba tyto přístupy jsou komplexnější, pokročilejší, ale v této práci nejsou využívány. BERT je poměrně novinkou, která stále není moc rozšířená. FastText je ve srovnání s Word2vec pomalejší, náročnější na velikost a vzhledem k použití ve webové aplikaci je potřeba volit raději úspornější řešení.

Jak převést data na vybranou reprezentaci?

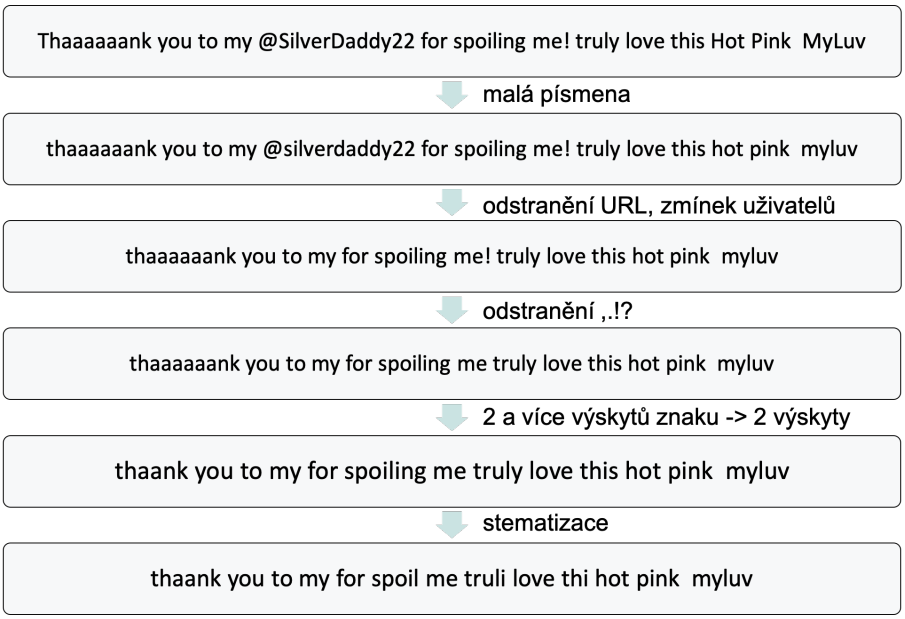
Pro většinu z výše zmíněných přístupů je možné buď využít existující knihovnu, nebo stáhnout již předtrénovaný model. V některých situacích může být vhodnější natrénovat si ho vlastními silami na vlastních datech. Předtrénované modely jsou ale zpravidla trénované na velkém množství dat, které by člověk sám nesehnal. Word2vec i FastText mají předtrénované modely dostupné pro širokou škálu různých jazyků:

- Word2vec pro 12 různých jazyků, mohou být načteny například s pomocí Python knihovny Gensim [37].
- FastText existuje pro 294 různých jazyků včetně češtiny, trénováno bylo na textech z Wikipedie.

3.3 Zpracování textu v této práci

Pro předzpracování textu je využita většina technik zmíněných v kapitole 3.1. Z dat je odstraněn šum (URL adresy, označení jiných uživatelů, písmena „RT“ značící retweet), data jsou převedena na malá písmena. Normalizace se ukázala jako obtížný krok, neexistuje na ni žádný jednoznačný návod a postup. Provádí se aspoň převedení dvou a více výskytů znaku na dva výskyty. Slova jako „thaaaaaanks“ a „thaaanks“ jsou díky tomu převedena na „thaanks“, což je více pozitivní než jen obyčejné „thanks“ s jedním a. Stop slova odstraněna nejsou, protože odstranění stop slov z knihovny `NLTK` vedlo ke zhoršení výsledků. Knihovna totiž mezi stop slovy obsahuje i takové výrazy, jež mají na emoci vliv (např. „not“, „no“, „is

not“. Pokud se odstranila pouze vybraná stop slova, nevedlo to ke zlepšení. Ze dvojice lematizace a stematizace byla zvolena stematizace, protože měla lepší výsledky, jak je ukázáno v experimentu v kapitole 6.2. Kromě toho je stematizace jednodušší a obvykle rychlejší, což hraje roli pro výslednou webovou aplikaci.



Obrázek 3.4: Příklad předzpracování tweetu

Pro reprezentaci textu je využíván model Word2vec předtrénovaný na Google News. Byl zvolen jako rozumný kompromis. Reprezentace pomocí BOW nebo One-hot kódování je příliš jednoduchá, nedokáže zachytit vztahy mezi slovy. BERT ještě není příliš rozšířený a předtrénované modely FastTextu jsou velké. Model Word2vec se slovníkem o velikosti 3 miliony slov má asi 2 GB, FastText se slovníkem 1 milion slov je stejně velký.

Kapitola 4

Strojové učení

Strojové učení (ML, Machine Learning) je oblastí umělé inteligence. Zabývá se algoritmy, které analyzují data, učí se z nich a poté jsou schopné dělat vlastní rozhodnutí [18].

Jednoduše lze celý proces popsat tak, že vstupní data se převedou na matematické vyjádření, algoritmus si je prohlédne a identifikuje v nich vzory. Tyto vzory se použijí k vytvoření modelu, jenž dokáže předpovídat další výsledky.

V kontextu NLP lze strojové učení využít například pro souhrn textu, generování jazyka, odpovídání na otázky, strojový překlad, generování textu k obrázku či detekci spamu.

4.1 Typy strojového učení

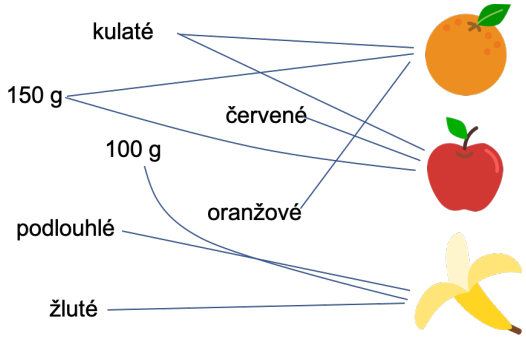
Implementace strojového učení mohou být (podle publikací [45] a [39]) rozděleny do tří rozdílných kategorií.

Učení s učitelem (Supervised learning)

Algoritmus iterativně předpovídá odpovědi k trénovacím datům a je opravován pomocí známých správných odpovědí. Učící fáze pokračuje, dokud se nedosáhne požadované úrovně přesnosti.

Do kategorie učení s učitelem spadají klasifikační problémy a regrese:

- **Klasifikační problém:** Algoritmus dostane vstup a má jej klasifikovat. Příkladem může být kus ovoce, u kterého je třeba rozpoznat, zda jde o banán, jablko či pomeranč. Algoritmus to rozpoznává například podle barvy, hmotnosti a tvaru — těmto vlastnostem říkáme parametry. Pokud je odpověď jenom ano/ne, tedy rozpoznáváme pouze dvě kategorie, mluvíme o binární klasifikaci.
  Detekce sentimentu také spadá pod klasifikační problémy, na základě parametrů rozhodujeme, jaká emoce se v textu projevuje.
- **Regrese:** O regresním problému mluvíme tehdy, když je výstupem číselná proměnná, například věk, cena či hmotnost. Příkladem je algoritmus předpovídající cenu bytu v různých částech Brna na základě rozlohy, umístění, občanské vybavenosti okolí či energetického standardu.



Obrázek 4.1: U klasifikačního problému se algoritmus postupně naučí, podle jakých kritérií může rozpoznat jablko, pomeranč a banán.

Učení bez učitele (Unsupervised learning)

Data používaná k trénování nejsou nijak označena ani klasifikována. Cílem tohoto typu učení je modelovat vzory a vztahy, rozdělit data do skupin. Žádná správná odpověď neexistuje, algoritmy jsou využívány k tomu, aby samy objevovaly a prezentovaly zajímavé struktury.

Do této kategorie spadají problém shlukování a asociační problém:

- **Problém shlukování:** Jedná se o situaci, kdy je cílem objevit vztahy mezi daty a rozdělit je do různých skupin. Naivním příkladem může být shlukování pohlaví podle délky vlasů. Algoritmus hledá podobnosti mezi délkami dvou vlasů, a pokud je rozdíl v délce dostatečně malý, zařadí je do stejné skupiny. Proces pokračuje, dokud nejsou všechny vlasy rozděleny do dvou kategorií.
- **Asociační problém:** V tomto případě algoritmy hledají vztahy mezi daty, pomáhají jim v tom určitá pravidla a příkazy typu if/then. Příkladem pravidla je: „Když zákazník koupí 12 vajec, s pravděpodobností 80 % koupí také mléko.“ Pravidlo přidružení má vždy dvě části – podmínku (známou položku) a její důsledek (položku, jež se hledá).

Zpětnovazebné učení (Reinforcement learning)

Tato metoda učení je založená na zpětné vazbě. Pokud algoritmus dostane obrázek jablka a identifikuje ho jako balon, dostane negativní zpětnou vazbu. V opačném případě obdrží pozitivní vazbu a podle toho se následně učí, jaká odpověď je ta správná.

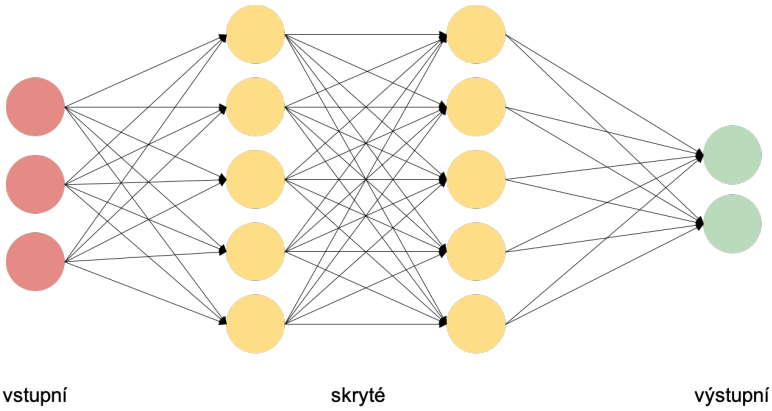
Kombinace obou přístupů (Semi-supervised learning)

Někdy se jako čtvrtá kategorie rozpoznává ještě varianta, která spojuje učení s učitelem s učením bez učitele, bývá nazývána semi-supervised [41]. Jde o případ, kdy se pro učení používají jak neoznačená, tak označená data.

4.2 Neuronová síť

Neuronová síť patří v kontextu strojového učení k metodám učení s učitelem. Jde o výpočetní systém vytvořený z velkého množství jednoduchých propojených prvků nazvaných neurony, které zpracovávají vstupy a přeměňují je na výstupy [15].

Je složená ze tří základních komponent, a to vstupní vrstvy, skrytých (výpočetních) vrstev a výstupní vrstvy. [29]

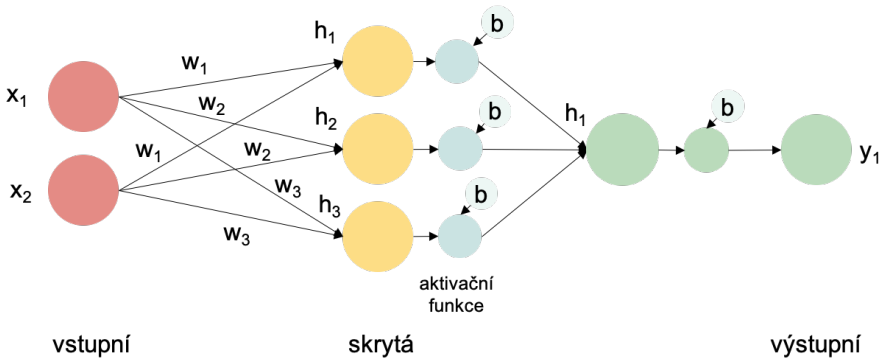


Obrázek 4.2: Model jednoduché neuronové sítě, která má dvě skryté vrstvy.

Učení sítě probíhá dvěma opakujícími se kroky:

1. dopřednou propagací, kdy síť odhaduje výsledek,
2. zpětnou propagací, kdy se minimalizuje chyba mezi skutečným a odhadovaným výsledkem.

Dopředná propagace



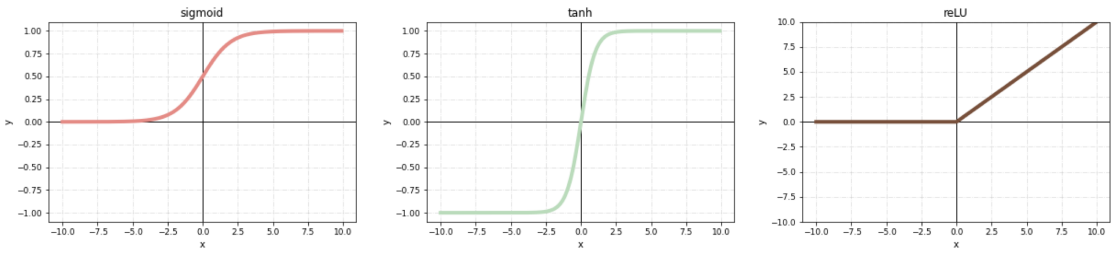
Obrázek 4.3: Do výše uvedeného teoretického modelu jsou při reálném výpočtu přidány ještě váhy jednotlivých hran a aktivační funkce.

Na začátku se náhodně inicializují váhy (w_1, w_2, w_3), zvolí se aktivační funkce a může začít výpočet:

$$h_1 = (x_1 \times w_1) + (x_2 \times w_1)$$
$$h_2 = (x_1 \times w_2) + (x_2 \times w_2)$$
$$h_3 = (x_1 \times w_3) + (x_2 \times w_3)$$

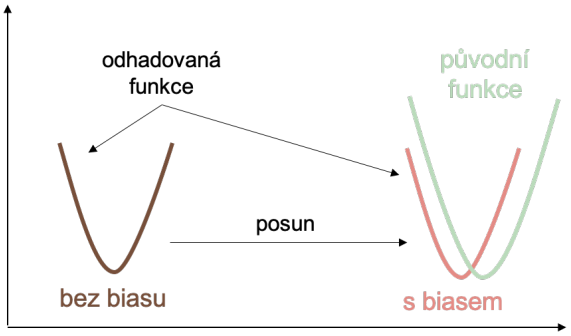
Výstup ze skryté vrstvy prochází přes aktivační funkci. Ta rozhoduje, které neurony budou aktivovány, tedy jaká informace projde do dalších vrstev. Typicky se jako aktivační funkce využívají [30]:

- sigmoida: $f(x) = \frac{1}{1+e^{-x}}$
- hyperbolický tangens (tanh): $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- ReLU (Rectified linear unit): $f(x) = max(0, x)$



Obrázek 4.4: Nejpoužívanější aktivační funkce.

Typicky bývá kromě vstupů do neuronové sítě přidáván ještě bias. Zpravidla bývá nastavován na 1 a má za úkol posunovat funkci.



Obrázek 4.5: Bias se přidává kvůli posunutí hledané funkce. Obrázek inspirován [29].

Zpětná propagace

Jakmile síť dojde k výsledku, ztrátová funkce vyhodnotí, jak moc se výsledek liší od očekávané hodnoty. Tomuto rozdílu se říká ztráta. Cílem učení je minimalizovat ztrátu, během učení by se měla snižovat [30]. Nejznámější ztrátovou funkcí je Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Poté se spočítají parciální derivace této ztráty, aby se zjistilo, jakou měrou se jednotlivé váhy podílí na celkové ztrátě. Derivace jsou vynásobeny malým číslem η nazývaným learning rate. Pokud je toto číslo příliš malé, i po dlouhém trénování bude síť daleko od optimálních výsledků. Je-li moc velké, dojde k závěru příliš brzy, závěr ovšem nebude příliš přesný [32]. Následně jsou upraveny váhy.

Celý proces dopředné a zpětné propagace se opakuje tak dlouho, dokud konverguje.

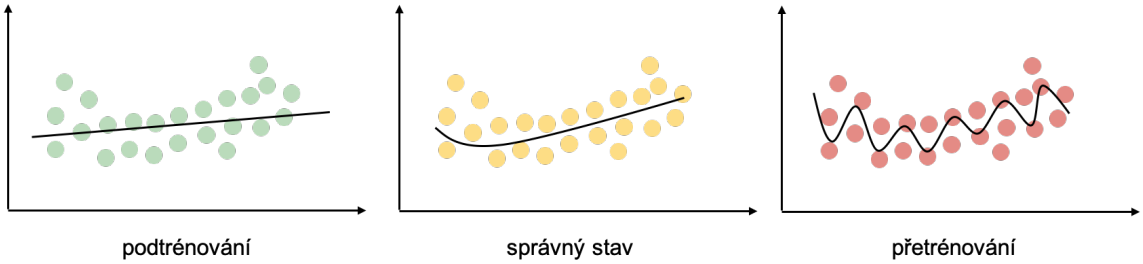
Jak rychle a jak kvalitně se síť naučí, je možné ovlivnit mnoha způsoby. Jedním z nich je dobrý návrh struktury sítě, správně vybrané vrstvy a jejich velikost. Dalším je například správné nastavení hyperparametrů, tedy manuálně zadávaných parametrů. Jde o již zmiňovaný learning rate nebo také aktivační funkci.

Při trénování se můžeme setkat se dvěma hlavními problémy označovanými jako:

1. Přetrénování (overfitting), kdy se model příliš adaptuje na data, z nichž se učí. Na těchto datech pak funguje skvěle, ale na neviděných datech selhává.

2. Podtrénování (underfitting), kdy model nefunguje dobře ani na trénovacích datech, ani na těch, jež ještě neviděl.

V případě podtrénování může pomoci zvětšit model, přidat parametry, trénovat více epoch. V případě přetrénování pomáhá přidat data, použít regularizaci nebo trénovat kratší dobu.



Obrázek 4.6: Při podtrénování není síť dostatečně naučená, naopak při přetrénování je naučená na konkrétní data a neumí dobře generalizovat. Správný stav leží někde mezi těmito dvěma krajními případy.

Rekurentní neuronové sítě (RNN, Recurent Neural Network)

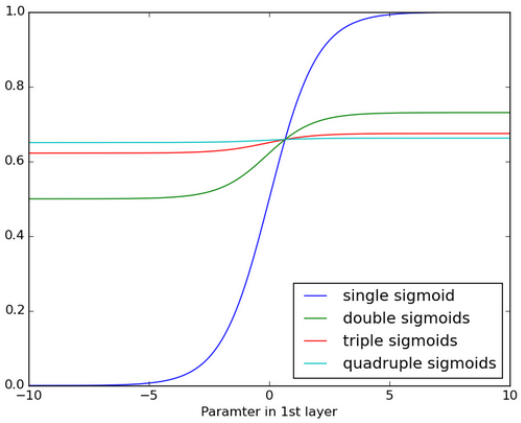
Člověk k přemýšlení využívá veškeré znalosti, které ve svém životě doposud nasbíral. Tradiční neuronové sítě to neumí, ty vždy zahodí, co se doposud naučily, a začínají od začátku. Právě tento problém odstraňují rekurentní neuronové sítě. Jsou to sítě, které obsahují smyčky, pomocí nichž si mohou pamatovat předchozí informace. Mají větší vyjadřovací schopnost než dopředné sítě, dokonce jsou ekvivalentní Turingovu stroji.

Pro jejich trénování se namísto obyčejné zpětné propagace používá zpětná propagace skrz čas (BPTT, Backpropagation Through Time). Chyba se šíří zpětně nejen do vedlejší vrstvy, ale i do předchozího časového kroku. RNN jsou výborně schopné propojit předchozí informaci s tou současnou. Pokud je například třeba vytvořit model, který předpovídá následující slovo podle toho předchozího, RNN poslouží dobře. Nemusí totiž řešit žádný delší kontext, stačí si zapamatovat předchozí slovo.

Jsou ovšem situace, kdy je nutné zapamatovat si delší kontext. Představme si text, který je jako příklad uvedený zde [35]: „Vyrostl jsem ve Francii. Narodila se tam moje maminka. ... Mluvím plynule *francouzsky*.“ Podle předchozího slova je možné odhadnout, že na místě hledaného slova „francouzsky“ bude nějaký název jazyka. Bez širšího kontextu bychom ovšem jen těžko odhadovali, že jde zrovna o francouzštinu. V těchto úlohách obyčejné RNN selhávají, a proto přichází na řadu jejich speciální typ, tzv. **LSTM sítě (Long Short Term Memory)**.

LSTM sítě jsou vytvořené speciálně k tomu, aby si uchovávaly dlouhodobý kontext. Řeší problém mizejícího gradientu, jímž trpí RNN. Jde o to, že při zpětné propagaci má vrstva

neuronů tendenci gradient zmenšovat a po několika vrstvách je už tak malý, že neprobíhá učení. Jednou z možností by bylo používat aktivační funkce, které po několikarém použití nedávají nulu.



Obrázek 4.7: Na obrázku je pomalu vidět mizející hodnota, když se aktivační funkce sigmoida aplikuje víckrát za sebou. Obrázek je převzatý z [34].

Druhým řešením, které využívají LSTM sítě, je použít tzv. brány. Architektura LSTM umožňuje zamezit zapisování do buňky tím, že uzavře bránu, a obsah buňky tedy zůstává po několik cyklů nezměněn. I když je brána otevřená, LSTM nenahrazuje obsah buňky kompletně, bere vážený průměr nové a předchozí hodnoty.

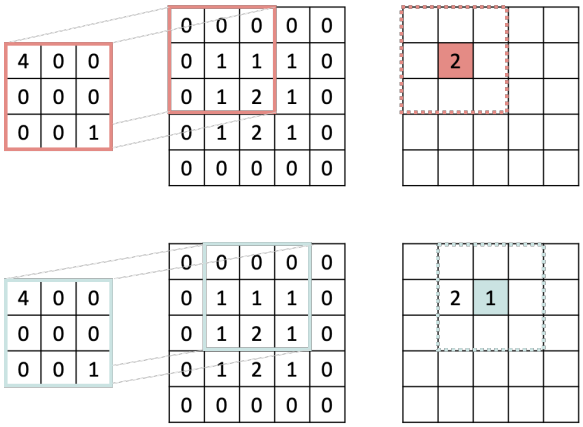
LSTM sítě mají bohaté využití jako například generování textu, rozpoznávání rukopisu, generování rukopisu, vytváření hudby, překládání jazyků. Velké využití nachází při práci s řečí i psaným textem, a proto mají své důležité místo i v této práci.

Konvoluční neuronové sítě (CNN, Recurent Neural Network)

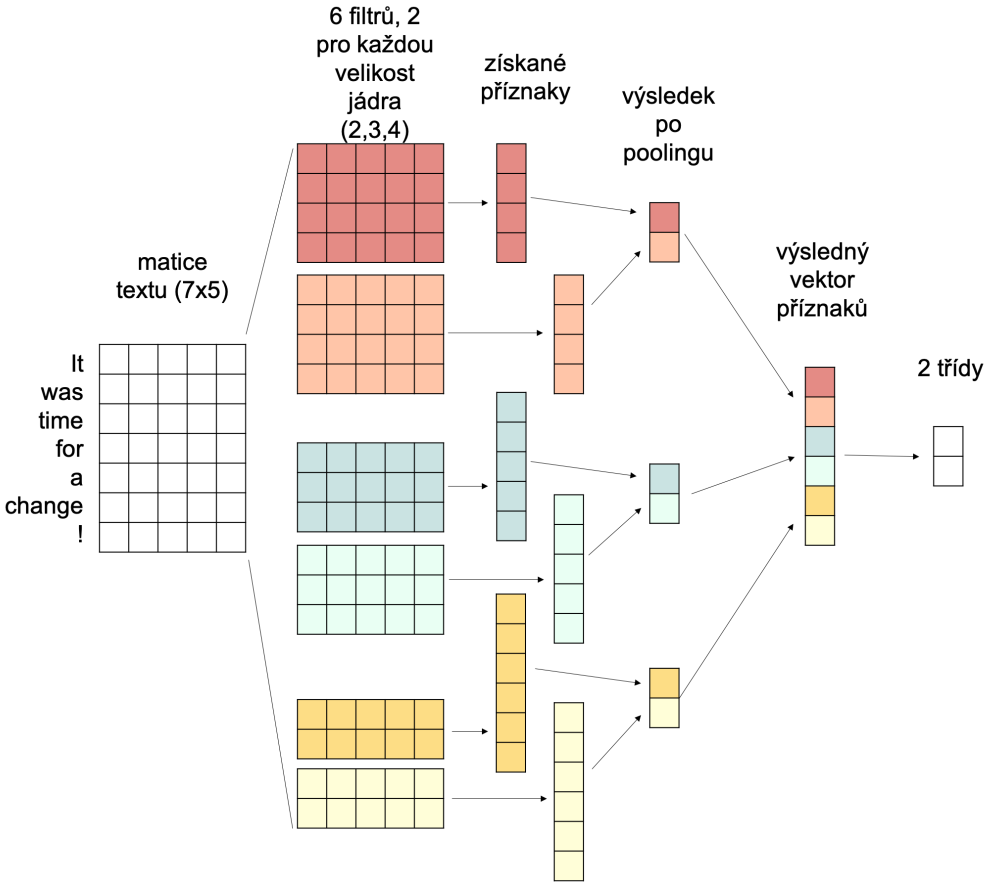
Pro klasifikační úlohy v NLP, jako jsou analýza sentimentu, detekce spamu nebo rozpoznání tématu, jsou vhodné také konvoluční neuronové sítě [14]. Tento typ sítí je specifický svými konvolučními vrstvami. Původně se používaly zejména pro práci s obrázky. Obrázky mají vysokou dimenzionalitu, a jelikož CNN nerealizují plné propojení mezi vrstvami [27], jsou při práci s nimi rychlé.

Konvoluční vrstva na vstup aplikuje filtr, provádí konvoluci. Za konvolučními vrstvami bývají obvykle umístěny také poolingové vrstvy, které zajišťují podvzorkování, tj. snižují dimenzi dat. Postupnou aplikací konvolučních a poolingových vrstev se získá relativně malá reprezentace vstupních dat, na niž se dají aplikovat plně propojené vrstvy.

Při práci s textem je princip podobný jako při práci s obrázky. Věta (či kus textu) se reprezentuje jako matice o velikosti $m \times n$, kde m značí počet slov a n značí délku jejich reprezentace. Slova mohou být reprezentována například pomocí embeddingů. Dále už vše funguje obdobně s tím rozdílem, že jsou obvykle používány filtry, jejichž šířka je stejná jako šířka vstupní matice, pohybují se tedy po jednotlivých slovech.



Obrázek 4.8: Při konvoluci se na vstupní matici aplikuje filtr. Filtrem se pohybuje přes celou matici, jednotlivé prvky, které leží na sobě se násobí a výsledky násobení se mezi sebou sčítají. Na obrázku jsou znázorněné první dva kroky, v prvním kroku probíhá výpočet $4 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 0 + 0 \times 1 + 1 \times 2 = 2$.



Obrázek 4.9: Konvoluční sítě se dají využít také na klasifikování textu. Na text se pohlíží jako na matici.

4.3 Strojové učení v této práci

Rozpoznávání emocí z textu je klasický klasifikační problém, jde tedy o učení s učitelem. Původní myšlenkou bylo implementovat pouze binární klasifikátor, který by rozpoznával pozitivní a negativní tweety, ale v praxi se ukázalo, že u mnoha tweetů nelze jednoznačně identifikovat tyto emoce, a tak je zapotřebí i neutrální třída. Přešlo se tedy na vícetřídní klasifikátor.

První varianta počítala s využitím LSTM sítě podle vzoru v kernelu z Kaggle [33]. Při binární klasifikaci se s ní dařilo dosahovat podobné přesnosti jako ve vzorové implementaci (80 %).

Během vývoje bylo třeba brát v úvahu, že výsledek bude zprovozněn v podobě webové aplikace. Je tedy nutné myslet na velikost i rychlost a LSTM sítě jsou ve srovnání s CNN pomalejší. Záleží na konkrétních parametrech, ovšem LSTM jsou ze své podstaty sekvenční, zatímco v CNN dochází k výpočtům paralelně.

Z aktivačních funkcí se pro klasifikační problémy na výstupu používají sigmoida a softmax. Sigmoida se využívá v případech, kdy může být správných více tříd. U klasifikace emocí je tedy vhodnější softmax počítající s tím, že je pouze jedna správná odpověď – tj. tweet nemůže být zároveň pozitivní a zároveň negativní. V jednom tweetu se sice mohou objevit negativní i pozitivní věty či slova, ale výsledná třída má být vybrána jednoznačně (tweet s negativním i pozitivním obsahem by mohl být označen například jako neutrální).

Jsou využívány klasické techniky pro zmírnění podtrénování (zvětšení modelu, trénování více epoch apod.) a nepřímé ovlivňování ztrátové funkce prostřednictvím nastavování vah tříd a vzorků (viz experiment v kapitole 6.3). Práce s neuronovými sítěmi neexperimentuje příliš do hloubky, jejím cílem je vytvořit jednoduchý klasifikátor použitelný ve webovém prostředí.

Kapitola 5

Výběr vhodné datové sady

Pro určování sentimentu na základě strojového učení je důležité mít k dispozici vhodná data pro trénování. Sehnat velké množství textových dat není v dnešní době problém. Internet je plný textových informací, které nesou nějakou emoci. Snadným zdrojem mohou být příspěvky na sociálních sítích, recenze, komentáře, příspěvky z diskuzí, případně články z blogů.

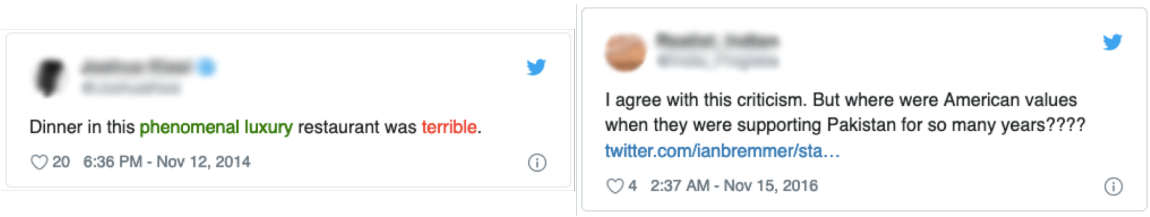
Potíž spočívá v tom, že kvůli použité metodě učení (učení s učitelem) je třeba mít data i správně klasifikovaná. Každá část (věta, jeden příspěvek, odstavec apod.) musí být opatřena informací, zda nese negativní, pozitivní, případně neutrální emoci.

5.1 Přínosy a úskalí tweetů jako zvolené formy dat

Sociální sítě jsou místem, kde lidé velmi často sdílejí své názory, proto jsou příspěvky z nich skvělým materiálem pro analýzu emocí. Krátké texty neboli mikrotexty podobné tweetům jsou v současné době široce rozšířené a v NLP se používají běžně.

Tato forma dat s sebou ovšem nese i nevýhody. Tou hlavní je nejasné vyjadřování. Na Twitteru lidé zcela běžně používají fonetický pravopis („b4“ pro slovo „before“), nejrůznější zkratky („otw“ pro „on the way“), typické jsou i překlepy a emociálně zabarvená slova („goooooood“ pro „good“). Ty úkol znesnadňují, a i když je možné je normalizovat (viz kapitola 3.1 o normalizaci), ošetřit všechny takové případy je složité.

Cílem této práce je vytvořit kromě systému na rozpoznávání emocí také hezké webové rozhraní, které by uživatelům umožnilo analyzovat emoci pro zadané téma. Twitter má povedené API, pomocí nějž se dají tweety pro výslednou aplikaci stahovat, takže také proto byla pro trénování zvolena přímo data z Twitteru.



Obrázek 5.1: Z některých tweetů je i pro člověka složité jednoznačně detekovat emoci. Levý tweet zobrazuje dvě protichůdné emoce v jedné větě. Pravý tweet ukazuje pozitivní emoci v první větě, ale negativní ve druhé.

5.2 Dostupné datové sady

Nasbírání dat a jejich klasifikace by byly příliš časově náročné. Na internetu je k dohledání velké množství připravených datových sad, některé obsahují úryvky z vědeckých prací, další recenze z internetových obchodů nebo názory diváků na filmy. Velká část z nich je zaměřena na konkrétní téma, jako například sady orientované na značku Apple, globální oteplování, ...). Tyto datové sady jsou vhodné pro konkrétní účel, ne však pro tuto práci, která se snaží vytvořit obecný klasifikátor, jenž si poradí se všemi tématy.

Cílem práce je vytvořit systém, který dokáže analyzovat, co si o nějakém tématu myslí veřejnost. Lidé svoje názory často prezentují na sociálních sítích, a proto se přímo nabízelo využít příspěvky z nich. Jako nejvhodnější se jevily příspěvky z Twitteru, které nebývají dlouhé. Původně měly limit 140 znaků, v dnešní době je to 280 znaků[1].

K vyhledávání datových sad byl využit primárně nástroj Google Dataset Search, který vyhledává data z různých zdrojů. Zahrnuje přitom pouze taková data, u nichž je jasný původ, jsou dostupné informace o autorech, o tom, jakým způsobem byla data posbírána, případně anotována. Několik datových sad použitých v práci pochází z platformy Kaggle, dceřinné společnosti Googlu, která sdružuje nadšence do strojového učení z celého světa.

Sentiment140

Jako nejvhodnější byla zpočátku vyhodnocena datová sada z projektu Sentiment140 [21], který se zabývá analýzou sentimentu tweetů. Jejich přístup ke klasifikaci je odlišný od přístupu popisovaného v této práci, nicméně data se jevila jako vyhovující. Sada obsahuje 1,6 milionu tweetů v angličtině. Jelikož jde o velké množství, data nebyla klasifikována ručně. Z Twitteru byly staženy tweety obsahující pozitivního smajlíka „:)“ a ty jsou považovány za pozitivní. Dále byly staženy tweety obsahující negativního smajlíka „:(“, které jsou brány jako negativní. Z obou kategorií je přítomno 800 000 vzorků. Tento přístup samozřejmě nezaručuje 100% správnost, ale pro ověření kvality dat byla provedena manuální kontrola, při které se vyhodnotilo 456 vzorků a pouze 5 z nich se ukázalo jako jednoznačně špatně klasifikované.

Data ze Sentiment140 mají následující strukturu:

Emoce	ID	Datum	Dotaz	Autor	Tweet
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	...
0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	...
4	1978676335	Sun May 31 00:06:06 PDT 2009	NO_QUERY	stevenburciaga	...

Tabulka 5.1: Emoce jsou označeny číselně, 0 značí negativní a 4 pozitivní. Tweety v datové sadě pochází z roku 2009, nechybí ani jejich ID, podle kterých se dají dodnes dohledat.

[1] Délku 280 znaků začal Twitter povolovat v listopadu 2017 – viz „Welcome to a world with 280-character tweets“

Součástí těchto dat je také sada pro testování. Představuje ji 499 ručně klasifikovaných tweetů, v nichž se objevují negativní, pozitivní i neutrální tweety. Tato sada byla během celého vývoje používána jako testovací a v kapitole 6 je označována jako TEST1.

Twitter US Airline Sentiment

V průběhu experimentů se ukázalo, že bude nezbytné doplnit data o vzorky s neutrální emocí. Více o tomto rozhodnutí je popsáno v části zabývající se experimenty.

Většina prozkoumaných datasetů pro analýzu emocí bohužel neutrální emoci neměla. Jednou z výjimek je sada Twitter US Airline Sentiment [5]. Obsahuje tweety z února 2015, v nichž uživatelé označovali aerolinky. Data jsou zpracována podrobně, nechybí ani údaje o tom, jaké je hlavní téma každého tweetu. Pro účel této práce byly však využity pouze text a sentiment.

Coachella 2015 Twitter

Dalším z datasetů, který měl pomoci doplnit neutrální emoci, byl Coachella 2015 [6]. Obsahuje téměř 4000 tweetů vztahujících se k hudebnímu festivalu Coachella. Tweety jsou anotované jako „negative“, „neutral“, „positive“ nebo „cant tell“. Stačilo tedy slovní reprezentaci transformovat na číselnou, přičemž „cant tell“ bylo reprezentováno jako neutrální.

The Emotion in Text dataset

Datová sada The Emotion in The dataset [7] obsahuje 40 000 tweetů s podrobně klasifikovanými emocemi. Jsou rozdělovány ne pouze do tří tříd, ale objevují se zde „naštvání“, „smutek“, „láska“, „nenávist“ apod. Tyto emoce byly převedeny na prostou číselnou reprezentaci tří tříd.

Customer Support on Twitter

Vzhledem k omezenému počtu datasetů, které by obsahovaly kromě pozitivní a negativní emoce také neutrální, byl vyzkoušen další přístup – stáhnout tweety bez anotací a anotovat je pomocí API výše zmíněného Sentiment140.

Jako data byla vybrána sada Customer Support on Twitter [8]. Dat jsou 3 miliony, nachází se v nich smajlíci, což se během experimentování také ukázalo jako důležité (viz kapitola 6.3). Data byla rozdělena na menší části zhruba po 10 000, protože API Sentiment140 neumožňuje zpracovat miliony dat v jednom požadavku.

Klasifikování dat pomocí API Sentiment140 se zdálo jako poměrně úspěšný nápad. Při pozdější ruční kontrole vzorků však vyšlo najevo, že zhruba 10 % je klasifikováno špatně. Ze 100 kontrolovaných bylo 10 chyb.

Ručně anotovaná data

Vzhledem k pochybnostem o kvalitě používaných dat byla pro trénování využita také jedna menší ručně klasifikovaná sada. Ručně se anotovalo 1000 tweetů z datasetu Favorited Tweets [9], který obsahuje oblíbené tweety regionu Střední Ameriky.

Tyto tweety jsou pouze dva roky staré, tudíž by měly obsahovat většinu emotikonů a výrazů používaných v současné době. Na druhou stranu se při ručním anotování, na němž se podíleli dva lidé, ukázalo, že jsou poměrně politicky orientované. Samostatně je tedy příliš použít nelze, lepší je kombinovat je s další sadou.

Emoji sentiment data

Při testování se dále projevilo, že obrovskou roli při určování emocí z textu hrají smajlíci. Z výše zmíněných datasetů se smajlíci nachází ve všech kromě toho ze Sentiment140. V jaké míře, je však otázkou. Protože klasifikátor chyboval ve tweetech se smajlíky, byla do testovacích dat zahrnuta ještě jedna speciální sada.

Emoji sentiment data [25] je lexikon 751 smajlíků s přiřazenými emocemi. Emoce ke smajlíkům byly stanoveny na základě ručního anotování tweetů v 13 různých evropských jazycích. Takto anotovaná data by byla skvělá i pro tuto práci, bohužel je autoři nedali k dispozici, zveřejnili pouze statistiky pro jednotlivé smajlíky.

Emoji	Unicode	Occurences	Negative	Neutral	Positive	Unicode name
😂	0x1f602	14622	3614	4163	6845	FACE WITH TEARS OF JOY
❤	0x2764	8050	355	1334	6361	HEAVY BLACK HEART
😭	0x1f62d	5526	2412	1218	1896	LOUDLY CRYING FACE

Tabulka 5.2: Lexikon neobsahuje jednoznačné určení emoce pro každého smajlíka, vyobrazuje pouze počet výskytů v negativních, neutrálních a pozitivních tweetech.

Z této statistiky pro smajlíky byl vybudován nový dataset, který obsahuje pouze textově reprezentované smajlíky a pro ně emoci. Pro každého smajlíka byla zvolena ta emoce, jež se objevovala v největším počtu tweetů. Tento postup není dokonalý, protože jak ukazují statistiky pro jednotlivé smajlíky, některé z nich jsou používány v podobné míře v pozitivních i negativních tweetech. Síť se však neučí klasifikovat pouze podle smajlíka, bere v úvahu i text, a právě ten by mohl výslednou emoci usměrnit.

Text	Sentiment
emojistart face with tears of joy emojiend	4
emojistart heavy black heart emojiend	4
emojistart loudly crying face emojiend	0

Tabulka 5.3: Pro účely této práce byl vytvořen jednoduchý dataset. Smajlíky jsou v něm reprezentovány unicode jménem a ohraničeny tokeny emojistart a emojiend. Za tuto reprezentaci jsou také nahrazovány v datových sadách použitých pro testování a později v klasifikovaných datech.

5.3 Rozdělení dat na testovací a trénovací

Dataset Sentiment140 již obsahoval 499 ručně anotovaných tweetů určených pro testování, dalších 10 % z celkového počtu bylo využíváno pro validaci. Ostatní datasety byly vyhledávány za účelem obohatit trénovací data, byly proto jako celek používány primárně k trénování.

	Celkem	Negativní	Neutrální	Pozitivní	Poznámky
Sentiment140	1 600 499	800 177	139	800 182	chybí neutrální data, odstranění smajlíci, obsahuje ručně anotovaná testovací data
US Airline	14 639	9 178	3 098	2 363	obsahuje i smajlíky, ale úzce zaměřené
Coachella	3 846	553	928	2 283	úzce zaměřené
Emotion	40 000	7 604	17 097	15 299	nutné převést podrobné emoce do 3 tříd
Customer S.	2 966 649	527 916	1 679 620	759 113	úzce zaměřené, klasifikované přes API Sentiment140
Ruční anotace	1000	539	37	424	spolehlivé anotace, úzce zaměřené (politika)
Emojis	969	108	380	481	čistě smajlíci

Tabulka 5.4: Tabulka shrnuje použité datasety, jejich velikost, rozložení tříd, výhody a nevýhody.

Z trénovacího datasetu je vždy 10 % určeno pro validaci během trénování. Výsledky se testují na ručně anotovaném datasetu Sentiment140 a na vlastnoručně anotovaných datech (viz 5.2). V závěrečných experimentech byla i tato data používána pro trénování.

Bližší informace, na jakých datasetech se trénuje a testuje, jsou popsány v kapitole 6.

Kapitola 6

Experimenty

Důležitou část práce tvoří experimentování s různými daty, velikostí modelu apod. Natrénovat model, který dokáže dobře rozpoznávat data stejné distribuce, jako jsou ta, na nichž se trénoval, není problém. Problém je, aby dokázal generalizovat.

Aby byly experimenty srovnatelné, byly všechny, pro něž to bylo možné, testovány na dvou datasetech. První dataset (v textu dále označován jako „TEST1“) pochází z projektu Sentiment140, obsahuje 499 ručně anotovaných tweetů. V několika prvních pokusech, kdy se pracovalo pouze s negativní a pozitivní třídou, z něj byly odstraněny neutrální vzorky a zbylo 360 vzorků. Druhý dataset (v textu dále označován jako „TEST2“) byl ručně anotován dvěma osobami, povedlo se dát dohromady 1000 tweetů. Je popsán v kapitole 5.2, původně byl vytvořen pro účely tréninku, ale v jiných experimentech se používal i k testování. Jako „TEST0“ je pak v testování zmiňován vzorek dat takových, na nichž se síť učí (tedy validační set). V prvních pokusech není zmiňován, protože má stejnou distribuci dat jako TEST1 a přesnost je srovnatelná.

6.1 Vyhodnocování úspěšnosti

Za účelem zjištění, který experiment dává lepší výsledky, je potřeba stanovit metriku, podle níž se bude porovnávat. Nejjednodušším přístupem je porovnávat procentuální přesnost, tedy kolik vzorků z celkového počtu bylo klasifikováno správně. Tento přístup nedává smysl v případě, kdy je poměr tříd výrazně nevyvážený. Pak by totiž stačilo vytvořit „klasifikátor“, který by klasifikoval vždy třídu, jíž je nejvíce, a tím pádem by měl vysokou přesnost. Jelikož jsou v této práci pro testování používány datasety s poměrně vyváženými třídami, je přesnost vyhovující metrikou.

Pro detailnější zjištění, v čem přesně klasifikátor chybuje, je využívána také metrika precision-recall [2] a matice záměn. Metrika pracuje s pojmy:

- True Positives (T_P) – Vzorky, které byly označeny jako třída A a skutečně patří do třídy A.
- True Negatives (T_N) – Vzorky, které byly označeny jako třída B a skutečně patří do třídy B.
- False Positives (F_P) – Vzorky, které byly označeny jako třída A, ale patří do třídy B.
- False Negatives (F_N) – Vzorky, které byly označeny jako třída B, ale patří do třídy A.

Precision udává poměr T_P vůči T_P a F_P, jak je ukázáno v rovnici 6.1. Recall se počítá jako počet T_P vůči T_P a F_N, jak zobrazuje rovnice 6.2.

$$Precision = \frac{T_P}{T_P + F_P} \tag{6.1}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{6.2}$$

Počet F_P, F_N, T_P i T_N přehledně zobrazuje rovněž matice záměn.

6.2 LSTM

Na počátku se pracovalo s jednoduchým sekvenčním modelem, s embeddingovou vrstvou, která si vytvářela vlastní embeddingy, a LSTM vrstvou podle kernelu z databáze Kaggle [33]. V této fázi se také jednalo pouze o binární klasifikaci, což vycházelo z dostupných dat – využíval se dataset Sentiment140 popsaný v kapitole 5.2, v němž jsou pouze pozitivní a negativní tweety. U této kapitoly není uváděna přesnost pro validační set „TEST0“, protože data z „TEST1“ jsou ze stejné distribuce jako trénovací i validační set. Přesnost TEST0 tedy koreluje s TEST2 a uvádět takové výsledky by bylo redundantní.

Parametry modelu byly odhadovány a zkoušeny experimentálně. Bylo experimentováno s počtem unikátních slov, která jsou parametrem první vrstvy `Embedding`, a s výstupní dimenzí embeddingů, vyšší čísla však nepřinesla lepší výsledky. Ani větší výstupní rozměr `LSTM` vrstvy nezaručil zlepšení, `dropout` hodnoty této vrstvy byly nastaveny na 0.2. Použití ztrátové funkce `categorical_crossentropy` a optimizeru `adam` je pro řešení těchto potíží běžné, stejně jako aktivační funkce `softmax` v poslední vrstvě. Zvětšení modelu o další dvě LSTM vrstvy nepřineslo vylepšení, zvýšilo výpočetní náročnost.

Různý počet trénovacích dat

Z datasetu Sentiment140 je k dispozici 1,6 milionu tweetů, polovina pozitivních a polovina negativních. Trénovat s tímto počtem je však výpočetně náročné, natrénovat každou jednu epochu pak trvá desítky minut, ačkoliv se trénuje na GPU. Z toho důvodu byl proveden experiment, jak velký je rozdíl, když se použijí různé počty dat.

V první fázi bylo využito 500 000 vzorků z každé třídy. Dalším pokusem bylo snížit počet dat, a to tak, že negativních i pozitivních je 50 000. V následujícím pokusu se použilo 80 000 pozitivních a 50 000 negativních. Tato myšlenka vyšla z úvahy, že pozitivních tweetů bývá více než negativních. Tabulka 5.4 sice ukazuje, že ne vždy je to pravda, ale převláda negativních tweetů se projevuje například u datasetu z uživatelské podpory, což je pochopitelné. Podporu lidé kontaktují, když mají problém.

	TEST1	TEST2
baseline (článek Sentiment140)	**82,2 %**	
500tisNEG_500tisPOZ	77,99 %	65,84 %
50tisNEG_50tisPOZ	77,43 %	64,81 %
50tisNEG_80tisPOZ	76,88 %	63,78 %

Tabulka 6.1: Tabulka ilustruje experimenty s počty dat. Lepších výsledků bylo dosaženo, když byly poměry tříd vyrovnané

Předzpracování textu

Výše uvedená zjištění prokázala, že pro další testování bude vhodné pracovat s 50 000 pozitivních a 50 000 negativních tweetů, s vyrovnaným poměrem tříd se dosáhlo lepších výsledků. Testování s tímto počtem není tak výpočetně náročné a bude možné na něm vidět změny. Až se podaří vyladit nejlepší kombinaci parametrů a předzpracování textu, je možné ji pak natrénovat s vyšším počtem dat.

Slovník měl původně velikost 30 000 slov, cílem prvního pokusu z tabulky 6.2 bylo prozkoumat, jak velký vliv to má. Experimentálně tedy byl slovník zmenšen pouze na 20 000 unikátních slov. Ukázalo se, že vliv není příliš vysoký.

Běžným postupem je odstranit stop words. Odstranění anglických stop words, která je možné vzít z knihovny `nltk`, však přineslo mírné zhoršení výsledků.

Další pokus byl projít seznam stop words z knihovny `nltk` ručně a vyřadit z nich ta slova, která by mohla mít nějaký vliv na výsledky. Tento postup sice vedl k mírnému zlepšení, ale jakmile se odstranění stop words začalo kombinovat se stemmingem, výsledky se zhoršily (viz pokus č. 8)

Jiný experiment testoval použití stemmingu místo lemmatizace. Funkce `PortStemmer` z knihovny `nltk` přinesla zlepšení oproti lemmatizaci v řádu několika procent. Zřejmě je v testovacích datech dost slov, která se díky stemmingu namapují na stejný základ, což pro lemmatizaci neplatí, a to ovlivní výsledek.

Kombinace stemming + odstranění stop words je slepou uličkou. Ať už bylo jejich pořadí jakékoliv, výsledky byly vždy špatné.

	TEST1	TEST2
baseline (50tisNEG_50tisPOZ)	**77,43 %**	**64,81 %**
slovnik_20tis_slov	77,15 %	64,22 %
stop_words_NLTK	76, 04 %	66,4 %
stop_words_rucni_vyber	78,55 %	66,27 %
portstemmer	**81,61 %**	**69,06 %**
stemmer_stop_words	55,99 %	52,64 %

Tabulka 6.2: Tabulka udává přesnost pro dva testovací datasety pro výše zmíněné experimenty. Jako nejúspěšnější se jeví pokus, který byl natrénován na vzorku 50 000 pozitivních a 50 000 negativních tweetů, a testoval použití stematizace místo lemmatizace.

Práce byla v začátcích inspirována přístupem projektu Sentiment140, z nějž pochází také trénovací dataset. Podařilo se dosáhnout zhruba stejné přesnosti. V projektu Sentiment140 přesnost za použití metody SVM (Support Vector Machine) dosahovala 82,2 %.

6.3 CNN

Pro NLP se často používají i CNN sítě, které nejsou tak výpočetně náročné jako LSTM. Další pokusy byly tedy prováděny s tímto typem sítě. Novou změnou bylo od začátku také přidání předtrénovaných embeddingů Word2Vec.

Experimenty s modelem

U modelu byly vyzkoušeny nejrůznější hyperparametry, změny ve výsledcích se vždy pohybovaly v rozsahu několika málo procent. Testoval se různý počet konvolučních vrstev, vliv

byl malý. Testovala se taktéž různá velikost jejich kernelu, prakticky bez rozdílu. Parametr obou `Dropout` vrstev byl nakonec nastaven na 0.2, první `Dense` vrstvy na 256. Některé z experimentů jsou popsány v tabulce níže. Snahou je vytvořit klasifikátor, který bude umět generalizovat pro různé příspěvky z Twitteru. Případy, kdy je výrazně lepší výsledek pro jeden testovací dataset, ale ne pro druhý, tedy nejsou zajímavé.

	TEST1	TEST2
baseline (nejlepší výsledek s LSTM)	**81,61 %**	**69,06 %**
2_conv_vrstvy	77,99 %	67 %
9_conv_vrstev	76,19 %	66,86 %
conv_filtr_velikost_100	75,94 %	66,13 %
conv_filtr_velikost_300	79,45 %	65,99 %
conv_filtr_velikost_400	76,94 %	69,95 %
dropout_0.1	77,69 %	65,98 %
dropout_0.2	77,69 %	65,98 %
dropout_0.2_a_dense_256	**77,19** %	**68,48** %
dropout_0.3_a_dense_256	77,19 %	66,13 %

Tabulka 6.3: Tabulka ilustruje některé testované hodnoty hyperparametrů.

Výsledky jsou o něco horší než v případě LSTM, ale celá tato práce je, kvůli použití na webu, o hledání kompromisů mezi dobrým výkonem a rychlostí. Proto je výsledek akceptovatelný.

Přidání neutrálních dat

Původním záměrem bylo neutrální data nepřidávat a manuálně nastavit určitou hranici. V prvním experimentu zdokumentovaném v tabulce 6.4 jsou za neutrální považovány vzorky, kde je $p \in (0.4; 0.5)$, ve druhém kde je $p \in (0.45; 0.55)$. Tento postup však nedával dostatečně uspokojivé výsledky. Experimentálně byla ještě vyzkoušena kombinace se slovníkovou metodou podle lexikonu pozitivních a negativních slov [22]. Pokud síť dávala jednoznačný výsledek (pravděpodobnost > 0.7), bral se jako definitivní. Jestliže si nebyla jistá, bral se ohled na slovníkový přístup (tedy na počet pozitivních a negativních slov ve tweetu).

	TEST1	TEST2
baseline (předchozí pokus s 2 třídami)	**77,19 %**	**68,48 %**
NEU_0.4_0.5	**54,8 %**	**64,2 %**
NEU_0.45_0.55	54,41 %	63,7 %
slovnikovy_pristup	56,02 %	33,14 %

Tabulka 6.4: Trénování na datech bez neutrální třídy se ukázalo jako ne příliš úspěšné.

Později byl slovníkový přístup kombinován ještě s výsledky neuronové sítě, která trénovala na třech třídách (tedy i na neutrálních datech). Výsledek opět o něco vylepšila, ale ne významně. Byl to spíš jen experiment, cílem této práce nebylo postavit klasifikátor založený na slovníku.

Neutrální data by v trénovacím datasetu neměla chybět. Učení se pouze z negativních a pozitivních příkladů nevede k přesné klasifikaci neutrálních dat [24], což dokazuje i tabulka 6.5. Neutrální vzorky navíc pomáhají k lepšímu rozlišování mezi pozitivními a negativ-

ními příklady. Proto další kroky směřovaly k nalezení datasetu obsahujícího také neutrální vzorky.

	Slovníkový přístup				**Práh (0.4;0.5)**			
	prec.	**recall**	**f1-score**	**supp.**	**prec.**	**recall**	**f1-score**	**supp.**
0 (NEG)	0.62	0.74	0.67	147	0.73	0.61	0.67	210
2 (NEU)	**0.22**	**0.33**	**0.27**	**93**	**0.02**	**0.27**	**0.04**	**11**
4 (POZ)	0.77	0.55	0.64	258	0.77	0.51	0.61	277
acc.			0.56	498			0.55	498
macro avg	0.54	0.54	0.53	498		0.47	0.44	498
weighted avg	0.62	0.56	0.58	498		0.55	0.62	498

Tabulka 6.5: Jak u slovníkového přístupu, tak u nastavování prahových hodnot je problém s neutrální třídou (třída s označením 2). Nízká precision značí, že bylo jako neutrální označeno i spoustu vzorků z jiných tříd. Nízký recall zároveň ukazuje na to, jak nízké procento z neutrálních vzorků se podařilo správně klasifikovat.

Kombinace datasetů

Prvním pokusem bylo vzít stávající dataset Sentiment140 a doplnit ho o neutrální data. Neutrální vzorky pocházely ze dvou sad, a to Coachella a Emoji. Obě jsou popsané v kapitole 5. Testovaly se různé poměry tříd.

	TEST0	TEST1	TEST2
baseline (2 třídy a práh)		**54,8 %**	**64,2 %**
28157NEG_18107NEU_37582POZ	75,23 %	51,6 %	53,37 %
8157NEG_18107NEU_67582POZ	75,19 %	55,62 %	69,65 %
78157NEG_18107NEU_87582POZ	**78,2 %**	**57,23 %**	**68,18 %**

Tabulka 6.6: V prvním případě bylo mnoho vzorků chybně klasifikováno jako neutrální, se zvyšováním počtu pozitivní a negativní třídy se výsledky vylepšovaly. Od této tabulky už dává smysl uvádět i TEST0 (tedy validační set), protože trénovací data jsou jiné distribuce než oba dva testovací sety (TEST1 a TEST2).

Ladit poměry tříd v trénovacích datech tak, aby byl výsledek na vybraných testovacích sadách dobrý, nestačí. Klasifikátor musí umět dobře generalizovat a fungovat na reálných tweetech. Místo skládání několika datasetů do sebe byla tedy dalším pokusem kompletní výměna dat. Na řadu přišel dataset Customer Support on Twitter anotovaný s pomocí Sentiment140.

Pro další testy se tedy pracuje s poměrem 1:1:1. Už experimentování s LSTM ukazovalo, že nejlepších výsledků se dosahuje s vyváženým počtem tříd. Je to nejrozumnější přístup, protože výsledný klasifikátor má být použitý na webu, kde budou uživatelé analyzovat vlastní témata. Mohou mít zájem o analýzu sentimentu čistě pozitivních témat (dovolená, úspěch, ...) i čistě negativních (ztráta, smutek, ...).

	TEST0	TEST1	TEST2
baseline (předchozí úspěšný pokus)	**78,2 %**	**57,23 %**	**68,18 %**
50tisNEG_50tisNEU_50tisPOZ	**86,25 %**	**60,64 %**	**47,6 %**
50tisNEG_20tisNEU_50tisPOZ	87,49 %	58,83 %	44,74 %
40tisNEG_20tisNEU_40tisPOZ	91,93 %	56,83 %	40,49 %
30tisNEG_50tisNEU_20tisPOZ	87,49 %	59,84 %	45,71 %

Tabulka 6.7: V prvních dvou pokusech bylo zastoupení tříd zvoleno experimentálně. Ve třetím bylo přizpůsobenu setu TEST1 (rozložení negativní:neutrální:pozitivní je zhruba 2:1:2), ve čtvrtém obdobně setu TEST2. Tento přístup však neměl dobré výsledky.

Pokusy s embeddingy

Ve výchozím stavu se pracuje s 30 000 unikátními slovy a jejich reprezentací pomocí Word2Vec modelu natrénovaného na Google News. Byl vyzkoušen experiment, kdy se používalo 40 000 unikátních slov (v datasetu bylo celkem 43 000 unikátních slov), poté se použil jiný předtrénovaný embedding (předtrénovaný na Wiki news), zkusil se přístup bez předtrénovaného embeddingu a také s povoleným trénováním Word2Vec modelu.

	TEST0	TEST1	TEST2
baseline (předchozí úspěšný pokus)	**86,25 %**	**60,64 %**	**47,6 %**
40tis_unikatnich_slov	87,02 %	60,4 %	41,98 %
trainable=true	**87,26 %**	**58,03 %**	**59,26 %**
bez_predtrenovaneho_embeddingu	91,85 %	61,24 %	43,1 %
jiny_embedding_wiki_news	90,58 %	60,8 %	44,4 %

Tabulka 6.8: Použití embeddingů předtrénovaných na různých datech neudělalo moc velký rozdíl. K vylepšení by nicméně mohlo vést například to, kdyby byly embeddingy natrénované přímo na datech z Twitteru.

V pokusu, kde bylo povoleno trénování předtrénovaného Word2Vec modelu, se výrazně vylepšil výsledek pro TEST2. Je to patrně způsobeno tím, že v něm jsou slova, která nejsou obsažená v předtrénovaném modelu. TEST2 jsou ručně anotovaná data dost orientovaná na politická témata, je tedy možné, že se v nich vyskytují výrazy, které v datasetu TEST1 nejsou. Proto je výsledek lepší pro TEST2 a nikoliv pro TEST1.

Přidání manuálně anotovaných dat

Manuálně anotovaný dataset, který byl v předchozích pokusech využíván pro testování (TEST2), byl tentokrát použit pro trénování. Množství manuálně anotovaných dat je velmi malé, a tak bylo využito váhování vzorků. Váhy byly nastavovány experimentálně, předávány funkci modelu `fit` jako parametr `sample_weight` ve formě pole s hodnotou váhy pro každý vzorek. Při prvním experimentu byla manuálním datům dána váha 0.65 a ostatním 0.35, při druhém manuálním 0.7 a ostatním 0.3. Keras váhy vzorků využívá při zpětné propagaci.

Nahrazení smajlíků

Během různých výše uvedených experimentů bylo samozřejmě ručně zkoumáno, v jakých případech dělá klasifikátor chyby. Pomohla k tomu matice záměn, metrika precision-recall

	TEST0	TEST1
baseline (předchozí úspěšný pokus)	**87,26 %**	**58,03 %**
manualni_0.65_ostatni_0.35	79,18 %	60,28 %
manualni_0.7_ostatni_0.3	79,67 %	55,02 %

Tabulka 6.9: Tabulka ilustruje příklady experimentů s váhami. Příliš vysoká váha manuálně anotovaných dat byla na škodu.

i manuální prozkoumání dat. Na testovacím datasetu z projektu Sentiment140 (označovaný jako TEST1) se ukázalo, že chybuje ve tweetech obsahujících smajlíky. Původní trénovací data Sentiment140 totiž smajlíky neobsahovala, zatímco testovací ano. Reálným příkladem věty z testovacích dat je například: „learning about lambda calculus“. Sama o sobě se zdá neutrální, při změně na „learning about lambda calculus :)“ je ovšem emoce pozitivní.

Proto byla v průběhu testování přidávána data obsahující smajlíky a nezbylo než se s nimi nějakým způsobem vyrovnat. Poslední experiment je tedy zaměřen čistě na smajlíky. V rámci něj byla do trénovacích dat přidána databáze smajlíků (viz kapitola 5.2), tedy smajlík spolu s tím, jakou emoci zobrazuje. Tato databáze smajlíků i všechny smajlíky v textu byly nahrazeny za jejich unicode textovou reprezentaci, token označující začátek smajlíka a token označující konec smajlíka.

Co se týče dalších parametrů, váhy vzorků byly odstraněny, místo toho se využily váhy tříd. Testováním se několikrát ukázalo, že pro nejlepší výsledky je dobrý vyrovnaný počet dat negativních, neutrálních i pozitivních. Tím, že bylo z každého datasetu vybíráno vždy např. 50 000 negativních, 50 000 neutrálních a 50 000 pozitivních, se však část testovacích dat zahazovala. Místo toho se tedy v tomto experimentu použila funkce `compute_class_weight` z knihovny `sklearn`, která dokáže vypočítat váhy tříd tak, aby byly vybalancovány. Bohužel nejde současně použít `sample_weight` a `class_weight`.

Experiment	TEST0	TEST1	TEST2
baseline (předchozí úspěšný pokus)	**87,26 %**	**58,03 %**	**59,26**
dbSmajliku_tridy_vyrovnane	77,26 %	59,84 %	42,92 %

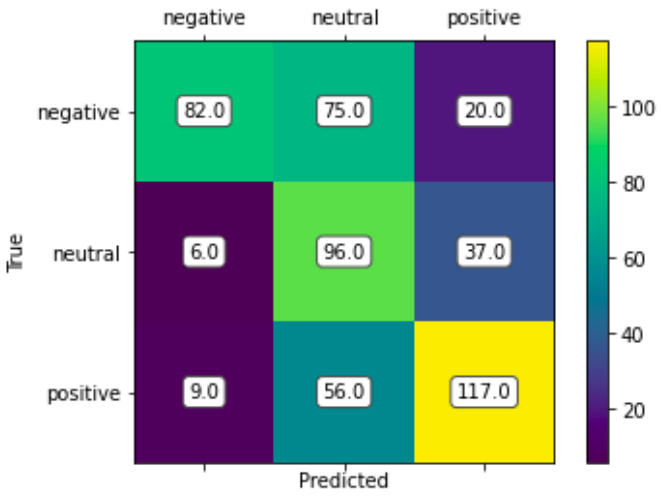
Tabulka 6.10: Výběrem správných dat, nastavením váhy tříd, aby byly všechny 3 třídy v rovnováze, a obohacením o databázi smajlíků se klasifikátor vylepšit nepovedlo.

Závěr experimentů

Na začátku se přesnost pohybovala kolem 80 %. Tehdy se ale jednalo pouze o binární klasifikátor. Jakmile byla přidána třetí neutrální třída, přesnost na tom stejném testovacím datasetu (TEST1) klesla k 50 %. Po výše uvedených experimentech se ji podařilo navýšit maximálně na 60 %.

Důvodem může být například to, že testovací dataset (TEST1) pochází z roku 2009. Tehdy byly vyjadřovací prostředky jiné, neexistovaly takové typy smajlíků, jež se používají dnes, apod.

Samotné toto číslo nemá příliš vypovídající hodnotu, protože není jisté, jak moc odpovídá dataset TEST1 obsahu, který se reálně vyskytuje na Twitteru. Byl však brán jako cíl. Podobně by se mohlo optimalizovat na jiná testovací data.



Obrázek 6.1: Matice záměn ukazuje, že největší problém je v rozeznávání neutrálních dat. Pozitivní či negativní data jsou často chybně rozpoznávána jako neutrální. 96 tweetů bylo správně rozpoznáno jako neutrální, dalších 131 bylo za neutrální označeno chybně.

Krok správným směrem byl přechod na CNN architekturu, urychlil trénování a poběží svižněji i ve webové aplikaci. Velké naděje byly vkládány do nahrazování smajlíků jejich textovými reprezentacemi, úspěch se bohužel nedostavil.

Experimenty ukázaly, jak důležitý je výběr trénovacích dat, největších změn v přesnosti se dosahovalo výměnou a kombinacemi datasetů. Najít data, na kterých by se podařilo natrénovat dobrý klasifikátor tweetů, je složité, protože tweety jsou plné překlepů, hovorových výrazů, slangu a dalších jazykových přípravků. Tato podoba dat ovlivňuje mnoho aspektů – například předtrénovaný embedding natrénovaný na Google News pokrývá asi jenom 60 % slov z datasetu, protože na Google News se zřejmě tolik nevyskytují překlepy a hovorové výrazy jako na Twitteru. Pro lepší verzi klasifikátoru by tedy bylo vhodné se zaměřit zejména na normalizaci dat. To je však náročný úkol, s nímž by částečně mohly pomoci snad jedině nástroje na kontrolu pravopisu. Opravit všechny nesprávné tvary slov, které lidé na Twitteru píšou, je nereálné.

Při ručním prozkoumání dat se také často ukazovalo, že nejen klasifikátor, ale i člověk může mít problém rozhodnout, jaká emoce v tweetu převládá. Je tedy na zváženou, zda má vůbec smysl klasifikovat tweet jako celek a zda je přístup rozdělování do 3 tříd správný.

6.4 Srovnání klasifikátoru

Na závěr bylo provedeno srovnání klasifikátoru s klasifikátorem Sentiment140 a manuálním anotováním.

Pro srovnání bylo použito 100 aktuálních tweetů vyhledaných na dotaz „a“. Díky tomu není vzorek tweetů zaměřený na žádné konkrétní téma. Tweety anotovalo 5 anotátorů. Klasifikátor vyvíjený v rámci této práce se s klasifikátorem Sentiment140 shoduje v 51 ze 100 případů. Zajímavé však je, že pouze v 29 z těchto 51 tweetů měly klasifikátory stejný názor jako lidé, kteří data anotovali.

Celkem v 49 tweetech se klasifikátory neshodly. V 15 případech se s názorem lidských anotátorů shodoval klasifikátor vyvíjený v této práci. V 24 případech se shodoval klasifikátor Sentiment 140. Ve zbylých 10 netrefil emoci podle lidského uvážení ani jeden z klasifikátorů.

Tabulka 6.11 ilustruje některé problematické tweety. Podrobnější výsledky jsou k nahlédnutí v příloze A.

Tweet	Klasifikátor z této práce	Sentiment140	Lidský názor
Secondary school was lowkey a very toxic time in our lives	2	2	0
GOOD MORNING MGA KUMARE I WOKE UP TO THIS. THANK YOU LORD. have a nice day	2	4	4
I made such a stupid mistake before the start of my latest race in ACC have a look	0	2	0
It's time for some Senate Republicans to perp walk @SenatorRomney out of the party as a show of respect to the 63 million voters he insulted.	2	4	0

Tabulka 6.11: Lidský názor vychází z toho, který sentiment byl v anotacích od 5 anotátorů nejčastěji. Ne vždy se jednomyslně shodli.

Kapitola 7

Návrh řešení a implementace

Tato kapitola popisuje použité technologie, implementovaný proces předzpracování textu, strukturu CNN sítě a tok programu. Věnuje se také vývoji výsledné aplikace, jež používá natrénovaný model.

7.1 Použité technologie

Pro vývoj byl použit skriptovací jazyk Python spolu s několika známými knihovnami. Využíval se nástroj Jupyter Notebook [3], který vytváří interaktivní webové rozhraní pro jazyk Python přímo v prohlížeči. Programový kód se díky němu dá spouštět po částech po jednotlivých buňkách, navíc je možné přidávat textové poznámky. Kód je díky tomu přehlednější a strukturovaný. Velkou pomocí byla cloudová služba Googlu s názvem Google Colab [1], která umožňuje využívat výkon GPU. Navíc přímo podporuje Jupyter Notebook.

Nejdůležitější knihovnou v této práci je patrně Keras [16], open-source knihovna psaná v Pythonu určená pro trénování neuronových sítí. Může fungovat na několika frameworcích, zvolen byl TensorFlow [10], který je nejznámější a nejpoužívanější. Pro práci s daty je využívána knihovna Pandas [42], která zvládá nejrůznější operace jako filtrování, selekci dat apod. Balíček NumPy [36] je využíván pro operace s poli a maticemi. Dále je využíván PorterStemmer z knihovny NLTK [13], která slouží pro zpracování přirozeného jazyka, a knihovna Gensim [37] pro načtení předtrénovaných embeddingů. Z modulu scikit-learn [2] je využívána funkce pro výpočet precision-recall metriky.

Výsledná webová aplikace používá HTML, CSS, JavaScript a Flask. Výběr těchto technologií a jejich využití jsou zdůvodněny v kapitole 7.5.

7.2 Předzpracování dat

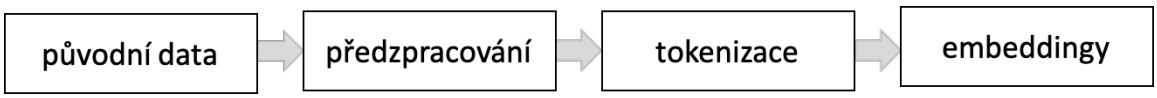
Pro manipulaci s daty se využívá knihovna `Pandas`, pro předzpracování pak také `NLTK` a modul `Preprocessing` z knihovny `Keras`. Data jsou nejdříve převedena na malá písmena, jsou z nich odstraněny nežádoucí informace. Úpravy a odstranění probíhají následovně:

- převod na malá písmena,
- odstranění zmínek uživatelů (@jmenouzivatele)
- odstranění URL adress
- odstranění znaků „RT“ značících retweet,

- odstranění interpunkčních znamének „.“, „,“, „!“, „?“,
- dva a více výskytu znaků je přeměněno na dva výskyty.

Obrázek 7.1: Toto jsou příklady tweetů z datové sady Sentiment140. Dodnes jsou na Twitteru dohledatelné podle ID. Obsahují emociálně zabarvená slova, zkratky a nespisovné výrazy. Dobře ilustrují také šum, který je třeba odstranit – označení jiných uživatelů, zdvojené znaky.

Následně je provedena lematizace (popsána v kapitole 3.1), pomocí tokenizeru je vytvořen slovník všech unikátních obsažených slov a jednotlivé tweety jsou převedeny na sekvence dlouhé `num_words`. Toto číslo je vhodné stanovit pro každý dataset zvlášť podle toho, kolik slov obsahuje. Kolik unikátních slov v datasetu je, se dá zjistit pomocí `word_index`. Z pokusů vyplynulo, že je zbytečné používat úplně všechna slova, některá se vyskytují pouze jednou např. z toho důvodu, že jde o překlepy.

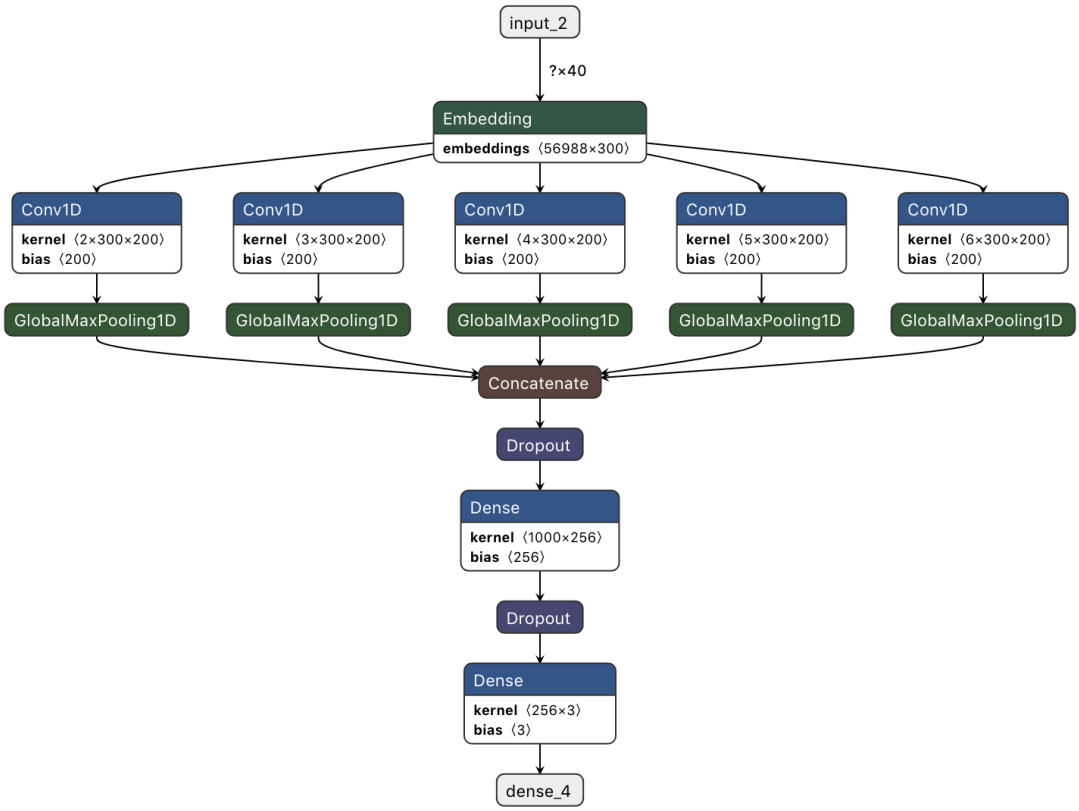


Obrázek 7.2: Data jsou předzpracována, následuje tokenizace a převedení na embeddingy.

7.3 CNN a vícetřídní klasifikátor

LSTM sítě jsou výpočetně hodně náročné, trénování trvá dlouho, což znamená méně proveditelných experimentů. Vzhledem k tomu, že výsledek má být použitelný ve webové aplikaci, je důležitá rovněž rychlost. Z toho důvodu byla původně zamýšlená LSTM síť nahrazena za CNN.

Byl použit funkcionální model po vzoru příkladu klasifikujícího komentáře z filmové databáze IMDB [11], který umožňuje sdílet vrstvy. CNN síť se skládá ze vstupní `Input` vrstvy, jejímž parametrem `shape` je délka každého vstupního vzorku. Tato hodnota byla nastavena na 40, tweety jsou krátké a pravděpodobně nemají více než 40 slov. Následuje vrstva `Embedding`, která využívá předtrénovaný Word2Vec model natrénovaný na Google News [4]. Jádrem je pět konvolučních vrstev připojených k vrstvě `Embedding`, následuje taktéž pět `GlobalMaxPooling1D` vrstev. Počet 5 vychází ze vzorového modelu a experimentů, které ukázaly, že změny počtu konvolučních vrstev nevedou ke zlepšení. Na závěr jsou připojeny vrstvy `Dropout` a `Dense` pro zamezení přetrénování a transformaci výsledku do požadovaného rozměru (3 třídy). Jako parametry modelu byly nastaveny `loss='categorical_crossentropy'` a `optimizer='adam'`. Poslední `Dense` vrstva má aktivační funkcí `softmax`, výsledkem je vektor udávající pravděpodobnost pro každou ze tří tříd. Experimenty s parametry CNN jsou více popsány v kapitole 6.



Obrázek 7.3: Obrázek znázorňuje strukturu CNN sítě.

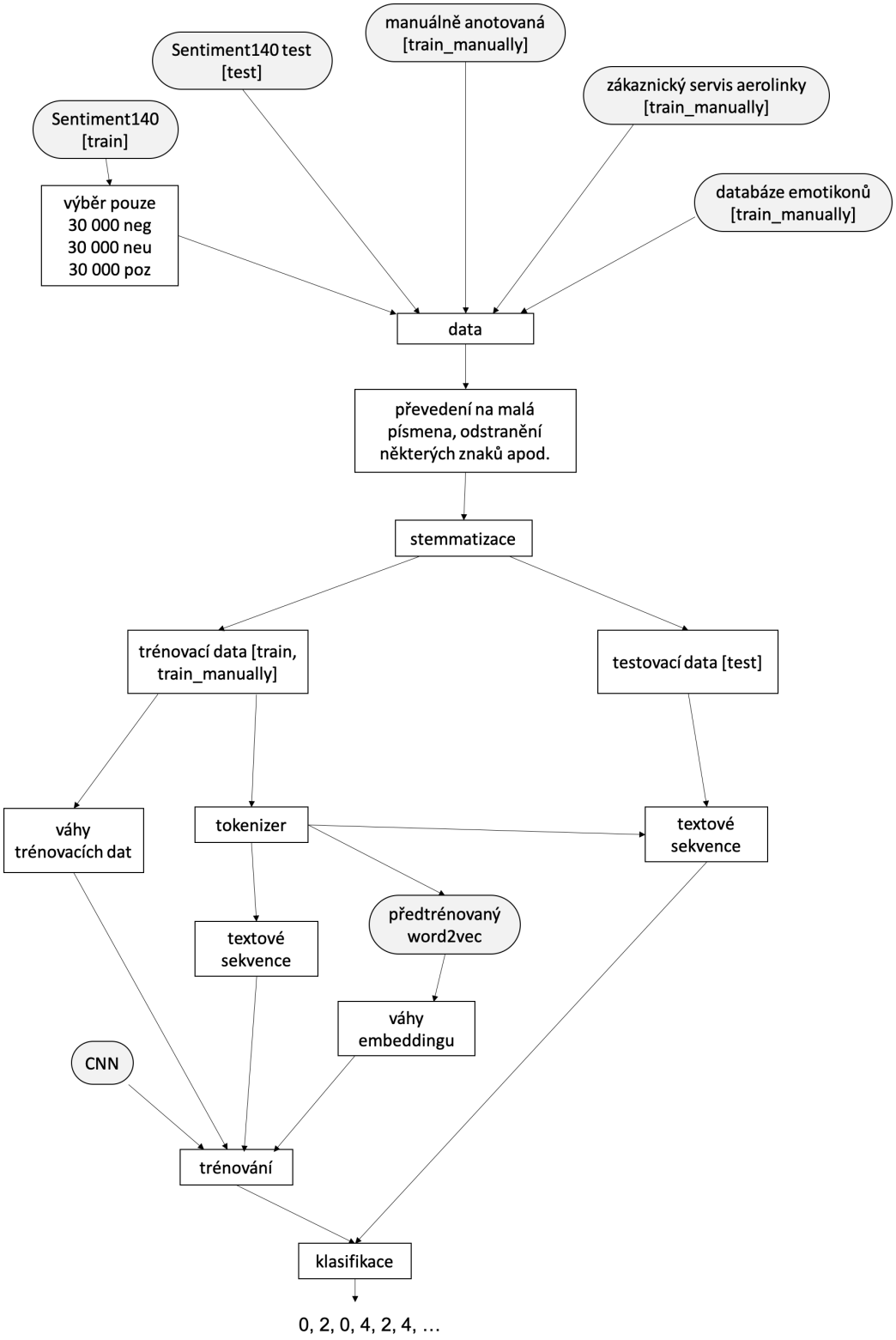
7.4 Výsledný program

Po experimentech popsaných v kapitole 6 je výsledkem program, který nejdříve načte všechna trénovací a testovací data jako strukturu `DataFrame` z knihovny `Pandas`. Datům je přidán jeden sloupec s názvem „split“, podle nějž jsou data rozdělena na trénovací, validační a testovací sadu.

Data jsou předzpracována dle postupu zmíněného v kapitole 7.2, jsou z nich odstraněny nepatřičné informace a je provedena stematizace. Tyto kroky mohly probíhat společně pro všechna data, ale následně je třeba je rozdělit na trénovací a testovací. Po rozdělení jsou nastaveny váhy trénovacích dat.

Dále je načten předtrénovaný embedding, je provedena tokenizace, jsou uloženy embeddingy pro obsažená slova. Pokud pro dané slovo není embedding v předtrénovaném modelu, je nastaven náhodně. Následuje definování struktury sítě a trénování.

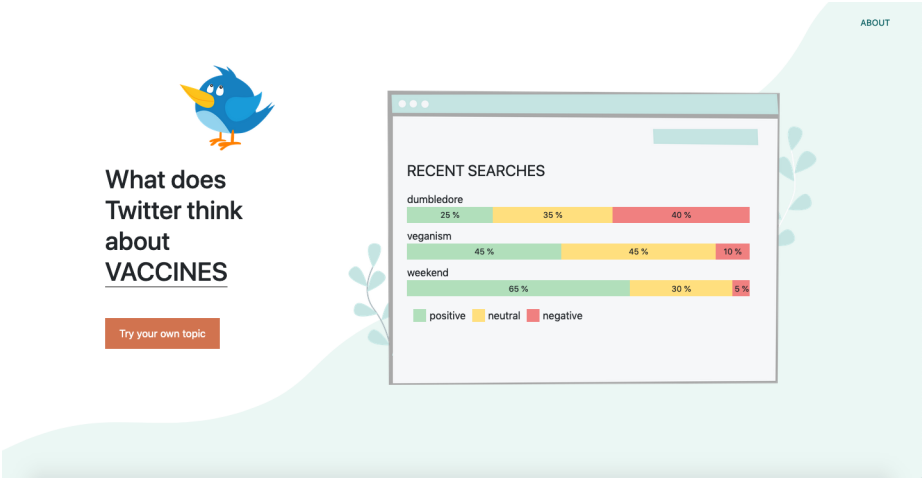
Na závěr je pomocí funkce modelu `predict` klasifikován testovací dataset a je vyhodnocena úspěšnost v podobě přesnosti, nebo podrobněji pomocí matice záměn a metriky precision-recall, která je popsána v kapitole 6.1.



Obrázek 7.4: Diagram zobrazuje tok programu. Vstupy jsou vyznačeny šedým pozadím.

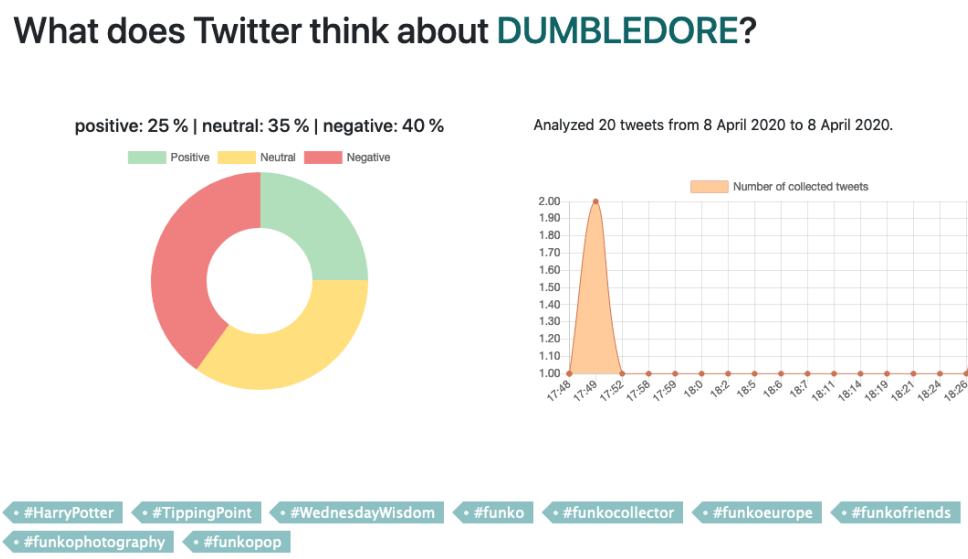
7.5 Webová aplikace

Cílem práce bylo reprezentovat výsledky tak, aby byly využitelné pro širokou veřejnost. Natrénovaný model byl proto použit k vytvoření aplikace WhatDoesTwitterThink.com.



Obrázek 7.5: Aby mohl výsledky práce využít kdokoliv, byl předtrénovaný model implementován do webové aplikace, která dokáže analyzovat data z Twitteru.

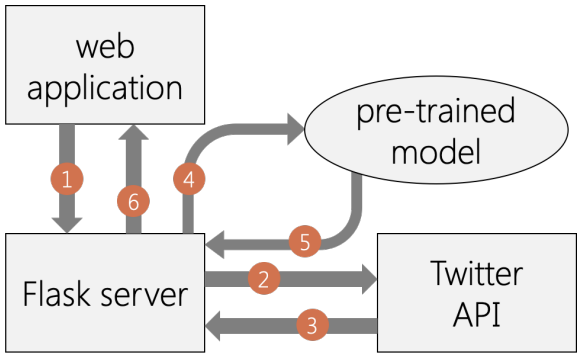
Aplikace umožňuje uživateli zadat libovolné téma, prostřednictvím API Twitteru stáhne tweety v reálném čase, klasifikuje je a zobrazí výsledky ve formě grafů.



Obrázek 7.6: Webová aplikace analyzuje tweety v reálném čase, zobrazuje výsledky ve formě grafů. Na obrázku je vidět graf rozložení tří rozpoznávaných emocí, graf toho, ze kdy pochází analyzované tweety, a hashtagy, jež se vyskytovaly v analyzovaných tweetech.

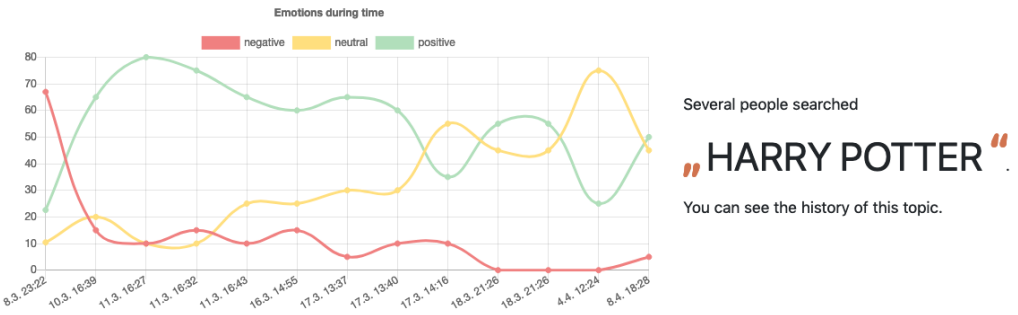
Řešení je postaveno na principu klient-server. Klienta zastupuje aplikace vytvořená pomocí HTML, CSS a JavaScriptu, jeho úkolem je pouze posílat žádosti na server a zpracovávat odpovědi přijaté ve formátu JSON. Pro server byl zvolen Flask, webový framework pro

Python. Původní varianta počítala s využitím knihovny Tensorflow.js, ale možnosti této knihovny jsou omezené a migrování z Pythonu do JavaScriptu je poměrně náročné. Na konvertování natrénovaného modelu existuje funkce, která jej převede z formátu .h5 do .json. Při práci s textem je však potřeba převést ještě tokenizer a s tím už knihovna nepočítá. Je možné vyřešit to manuálně a s tokenizerem pracovat jako se slovníkem, což je datová struktura, s níž si poradí jak Python, tak JavaScript. Přesto se nakonec ukázalo jako hladší řešení využít Flask.



Obrázek 7.7: Když uživatel zadá dotaz, klient pošle dotaz na server. Ten pošle dotaz na Twitter API, aby stáhl tweety. Dále využije předtrénovaný model a tokenizer, aby data klasifikoval a výsledky pošle zpět klientovi ve formátu JSON.

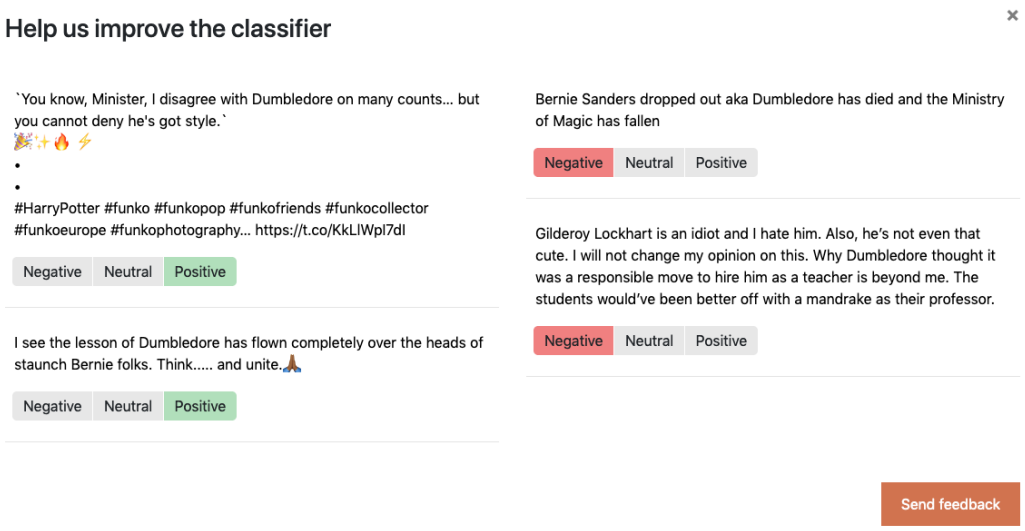
Server dále využívá SQLite – jednoduchou knihovnu psanou v jazyku C implementující relační databázový systém. Díky tomu je schopný ukládat historii vyhledávání a generovat uživatelům grafy pro vývoj emocí v čase.



Obrázek 7.8: Díky databázi je možné u výrazů, které již byly v předchozí době vyhledávány, zobrazovat také vývoj emocí v čase.

Dalším využitím databáze je ukládání zpětné vazby od uživatelů. Během testování a experimentů s modelem i daty se ukázalo, že správná skladba dat hraje velkou roli. Najít odpovídající dataset není snadné. Ty, které jsou dostupné, bývají pár let staré, tím pádem nedostatečně pokrývají současné vyjadřovací možnosti. Přibývají slova i zkratky, používají se smajlíci, kteří před pár lety ještě neexistovali. Z toho důvodu je aplikace využita také pro sběr dat a vytvoření ideálního datasetu. Uživatelé mohou u zobrazených příkladů poskytnout zpětnou vazbu, případně klasifikátor jednoduše opravit.

Do databáze se ukládá ID tweetu, počet negativních, neutrálních a pozitivních ohodnocení. Kvalita dat se zvyšuje, pokud více uživatelů anotuje stejný tweet. Vzhledem k tomu,



Obrázek 7.9: Uživatelé mohou špatně klasifikované tweety jednoduše opravit, jimi zadané údaje se ukládají do databáze pro další využití.

že aplikace stahuje data z Twitteru v reálném čase, se to může stát u méně frekventovaných témat, o nichž se tolik netweetuje. Ve většině situací se však každému uživateli zobrazí originální tweety.

```
cid         name        type        notnull     dflt_value  pk
----------  ----------  ----------  ----------  ----------  ----------
0           id          INTEGER     0                       1
1           topic       TEXT        1                       0
2           negative    REAL        1                       0
3           neutral     REAL        1                       0
4           positive    REAL        1                       0
5           date of se  TEXT        1                       0

cid         name        type        notnull     dflt_value  pk
----------  ----------  ----------  ----------  ----------  ----------
0           id          INTEGER     0                       1
1           tweet_ID    TEXT        1                       0
2           negative    INTEGER     1                       0
3           neutral     INTEGER     1                       0
4           positive    INTEGER     1                       0
```

Obrázek 7.10: Databázi tvoří dvě samostatné tabulky. Do jedné se ukládají veškerá vyhledávání, do druhé špatně klasifikované a opravené tweety.

7.6 Další vývoj a vylepšení

Databáze ručně opravených tweetů by se v budoucnu mohla stát velmi dobrým zdrojem dat pro pozdější trénování a vylepšování klasifikátoru. Zřejmě bude obsahovat přesně ty typy dat, v nichž klasifikátor chybuje, a tak by k vylepšení mohlo stačit přidat tato data ke stávající sadě, případně jim dát vyšší váhu. Aby se však toto povedlo, je potřeba na web přivést dostatek uživatelů.

Webová aplikace byla teprve spuštěna, nicméně Google už si ji stihl zaindexovat. Na webu Serprobot.com bylo ověřeno, že na dotaz „whatdoestwitterthink“ se nachází na 2. pozici ve výsledcích vyhledávání. Na dotaz „what does twitter think“ se nevešla do top 100.

Příčinou pravděpodobně je, že na „whatdoestwitterthink“ není vysoká konkurence. Na variantu s mezerami a dotazy jako „twitter sentiment analysis“ konkurence vysoká je. Mohlo by pomoci využívat SSL zabezpečení, v současné době web běží na nezabezpečeném http, a vyzkoušet další SEO techniky, aby se web dostal k širšímu publiku. Pak už by zbýval jen krůček k využití nově nasbíraných dat a vylepšení klasifikátoru.

Kapitola 8

Závěr

Práce popisuje proces odhadu emocí pomocí neuronové sítě za účelem klasifikace tweetů do tříd negativní, neutrální a pozitivní. Kombinuje tedy dvě populární oblasti, a to strojové učení a zpracování přirozeného jazyka.

Podařilo se najít zajímavá data, zpracovat je, vybrat jejich vhodnou reprezentaci a natrénovat klasifikátor. Práce se po celou dobu vyvíjela, původní koncept byl pozměňován a obohacován. Prvním takovým obohacením bylo přidání neutrální třídy, protože ne všechno je nutně pozitivní, nebo negativní. Dalším významným krokem byla změna typu sítě. Z LSTM neuronové sítě se přešlo k CNN, ze sekvenčního modelu na funkcionální.

S modelem i daty bylo provedeno mnoho pokusů. Některé vycházely z domněnek, jiné z načtených informací, další byly čistě experimentálního charakteru. Všemi těmi experimenty se podařilo dospět k mírnému zlepšení zhruba o 10 % oproti prvnímu pokusu, kdy byl natrénován model se 3 třídami. Klasifikátor pro 3 třídy byl nakonec pro vybraný dataset natrénován tak, že funguje s 60% přesností.

Přesnost na datech z Twitteru může být mírně odlišná například proto, že se vyvíjí jazyk a jazykové prostředky. V dnešní době se používají typy smajlíků, které před pěti lety ještě ani neexistovaly, apod. Během experimentů se ukázalo, že dobrá data jsou pro úspěšný klasifikátor tweetů důležitá, což se odrazilo i na funkčnosti výsledné aplikace.

Natrénovaný model byl použit k vytvoření webové aplikace WhatDoesTwitterThink.com, která dokáže stahovat tweety v reálném čase a klasifikovat je. Nakonec byla do aplikace přidána ještě možnost zpětné vazby. Uživatel může opravit špatně klasifikovaná data, výsledek se ukládá pro pozdější využití. Toto by mohlo pomoci vylepšit přesnost klasifikátoru. Tím, že uživatel opravuje špatně klasifikované tweety, postihuje přesně ten typ dat, s nímž má klasifikátor problémy. Samozřejmě, čím více uživatelů se k aplikaci dostane, tím větší potenciál pro využití získaných dat se nabízí.

Nejdůležitějším cílem do budoucna tedy je dostat web do povědomí co nejvíce uživatelů. Ti mohou klasifikátor opravovat, a tím vytvořit dataset, který povede ke zvýšení přesnosti.

Literatura

[1] *Colaboratory*. Google. Dostupné z: https://research.google.com/colaboratory/faq.html.

[2] *Precision-Recall*. Dostupné z: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html.

[3] *Project Jupyter*. Dostupné z: https://jupyter.org/index.html.

[4] *Word2vec*. Google, Jul 2013. Dostupné z: https://code.google.com/archive/p/word2vec/.

[5] *Twitter US Airline Sentiment dataset*. Oct 2015. Dostupné z: https://www.kaggle.com/crowdflower/twitter-airline-sentiment.

[6] *Coachella 2015 Twitter - dataset by crowdflower*. CrowdFlower, Nov 2016. Dostupné z: https://data.world/crowdflower/coachella-2015-twitter.

[7] *Sentiment Analysis in Text - dataset by crowdflower*. CrowdFlower, Nov 2016. Dostupné z: https://data.world/crowdflower/sentiment-analysis-in-text.

[8] *Customer Support on Twitter*. Kaggle, Dec 2017. Dostupné z: https://www.kaggle.com/thoughtvector/customer-support-on-twitter/home.

[9] *Favorited tweets – dataset by aliz*. data.world, Aug 2018. Dostupné z: https://data.world/a2liz/favorited-tweets.

[10] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Dostupné z: https://www.tensorflow.org/.

[11] Arshad, S. *Sentiment-Analysis*. Github, 2019. Dostupné z: https://github.com/saadarshad102/Sentiment-Analysis-CNN.

[12] Belica, B. M. *Metody sumarizace dokumentů na webu*. Brno, 2013. Diplomová práce. FIT VUT v Brně. Vedoucí práce Ing. Vladimír Bartík, Ph.D.

[13] Bird, S., Klein, E. a Loper, E. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[14] Britz, D. Understanding convolutional neural networks for NLP. *URL: http://www. wildml. com/2015/11/understanding-convolutional-neuralnetworks-for-nlp/(visited on 11/07/2015)*. 2015.

[15] Burger, J. *A Basic Introduction To Neural Networks* [online]. [cit. 2019-12-28]. Dostupné z: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html.

[16] Chollet, F. et al. *Keras* [https://keras.io]. 2015.

[17] Clercq, O. D. *Fine-grained Sentiment Analysis* [online]. Gein: Universiteit Gein, březen 2016 [cit. 2019-12-27]. Dostupné z: https://www.iaria.org/conferences2016/filesHUSO16/OrpheeDeClercq_Keynote_ABSA.pdf.

[18] Copeland, M. *What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning?* [online]. Červenec 2016 [cit. 2019-12-28]. Dostupné z: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

[19] Dr. S. Kannan, V. G. *Preprocessing Techniques for Text Mining.* Madurai: Madurai Kamaraj University, 2014 [cit. 2019-12-26]. Dostupné z: https://www.academia.edu/download/54879053/PreprocessingTechniquesforTextMining.pdf.

[20] Ganesan, K. *All you need to know about text preprocessing for NLP and Machine Learning* [online]. KDnuggets, duben 2019 [cit. 2019-12-26]. Dostupné z: https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html.

[21] Go, A., Bhayani, R. a Huang, L. Twitter sentiment classification using distant supervision. *Processing.* Leden 2009, sv. 150.

[22] Hu, M. a Liu, B. Mining and Summarizing Customer Reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: Association for Computing Machinery, 2004, s. 168–177. KDD '04. DOI: 10.1145/1014052.1014073. ISBN 1581138881. Dostupné z: https://doi.org/10.1145/1014052.1014073.

[23] Jadrníček, B. Z. *Shlukování slov podle významu.* Brno, 2015. Diplomová práce. FIT VUT v Brně. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

[24] Koppel, M. a Schler, J. The importance of neutral examples for learning sentiment. *Computational Intelligence.* Wiley Online Library. 2006, sv. 22, č. 2, s. 100–109.

[25] Kralj Novak, P., Smailović, J., Sluban, B. a Mozetič, I. Sentiment of Emojis. *PLOS ONE.* Public Library of Science. Prosinec 2015, sv. 10, č. 12, s. 1–22. DOI: 10.1371/journal.pone.0144296. Dostupné z: https://doi.org/10.1371/journal.pone.0144296.

[26] Liu, B. *Sentiment Analysis.* 1. vyd. Cambridge University Press, 2015. ISBN 978-1-107-01789-4.

[27] Lopez, M. M. a Kalita, J. Deep Learning applied to NLP. *ArXiv preprint arXiv:1703.03091.* 2017.

[28] Maks, I. a Vossen, P. Sentiment Analysis of Reviews: Should we analyze writer intentions or reader perceptions? In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013.* Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, Září 2013, s. 415–419. Dostupné z: https://www.aclweb.org/anthology/R13-1054.

[29] Malik, M. *Basics of Neural Network* [online]. Březen 2018 [cit. 2019-12-28]. Dostupné z: https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf.

[30] Maslowski, B. P. *Aplikace posilovaného učení při řízení modelu vozidla.* Brno, 2019. Diplomová práce. FIT VUT v Brně. Vedoucí práce Ing. Martin Šůstek.

[31] Mikolov, T., Yih, W.-t. a Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Atlanta, Georgia: Association for Computational Linguistics, červen 2013, s. 746–751. Dostupné z: https://www.aclweb.org/anthology/N13-1090.

[32] Moawad, A. *Neural networks and back-propagation explained in a simple way* [online]. Leden 2018 [cit. 2019-12-28]. Dostupné z: https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e.

[33] Nagy, P. *LSTM Sentiment Analysis: Keras.* Kaggle, May 2018. Dostupné z: https://www.kaggle.com/ngyptr/lstm-sentiment-analysis-keras/execution.

[34] Nicholson, C. *A Beginner's Guide to LSTMs and Recurrent Neural Networks* [online]. [cit. 2019-12-28]. Dostupné z: https://pathmind.com/wiki/lstm.

[35] Olah, C. *Understanding LSTM Network* [online]. Srpen 2015 [cit. 2019-12-28]. Dostupné z: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[36] Oliphant, T. E. *A guide to NumPy.* Trelgol Publishing USA, 2006.

[37] Řehůřek, R. a Sojka, P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.* Valletta, Malta: ELRA, Květen 2010, s. 45–50. http://is.muni.cz/publication/884893/en.

[38] Rizvi, M. S. Z. *Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework* [online]. Září 2019 [cit. 2019-12-28]. Dostupné z: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/.

[39] Samad, A. *What is Machine Learning?* [online]. Červenec 2019 [cit. 2019-12-28]. Dostupné z: https://becominghuman.ai/what-is-machine-learning-d292114cc6ce.

[40] Shaffy, A. *Vector Representations of Text for Machine Learning* [online]. Listopad 2017 [cit. 2020-1-7]. Dostupné z: https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7.

[41] Team, E. S. *What is Machine Learning? A definition* [online]. Březen 2017 [cit. 2019-12-28]. Dostupné z: https://expertsystem.com/machine-learning-definition/.

[42] team, T. pandas development. *Pandas-dev/pandas: Pandas.* Zenodo, únor 2020. DOI: 10.5281/zenodo.3509134. Dostupné z: https://doi.org/10.5281/zenodo.3509134.

[43] Thakkar, H. a Patel, D. R. Approaches for Sentiment Analysis on Twitter: A State-of-Art study. *CoRR.* 2015, abs/1512.01043. Dostupné z: http://arxiv.org/abs/1512.01043.

[44] Vallantin, L. *Why is removing stop words not always a good idea* [online]. Leden 2019 [cit. 2019-12-26]. Dostupné z: https://medium.com/@limavallantin/why-is-removing-stop-words-not-always-a-good-idea-c8d35bd77214.

[45] Zhang, A., Lipton, Z. C., Li, M. a Smola, A. J. *Dive into Deep Learning*. 2020. https://d2l.ai.

[46] Zhang, L., Ghosh, R., Dekhil, M., Hsu, M. a Liu, B. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*. 2011, sv. 89.

[47] Zhou, V. *A Simple Explanation of the Bag-of-Words Model* [online]. Listopad 2019 [cit. 2019-12-28]. Dostupné z: https://victorzhou.com/blog/bag-of-words/.

[48] Řehůřek, R. *Word2vec Tutorial* [online]. Únor 2014 [cit. 2019-12-28]. Dostupné z: https://rare-technologies.com/word2vec-tutorial/#app.

Příloha A

Porovnání klasifikátoru

Tabulka A.1 ukazuje prvních pět tweetů ze srovnání klasifikátoru z této práce s klasifikátorem Sentiment140 a s lidským úsudkem. Podrobné vyhodnocení se nachází na přiloženém paměťovém médiu.

Tweet	K	S140	AC	A1	A2	A3	A4	A5
RT @_E_vonn : Secondary school was lowkey a very toxic time in our lives	2	2	0	0	0	0	0	0
RT @UrstrulyAnil10 : #ZindabadSuperstarMAHESH his a brand @urstrulyMahesh https://t.co/1lyhPLN1k2	2	2	2	2	2	2	2	2
RT @NotLameAsChris : PART 1/2 alright before I get suspended again: here is a video basically highlighting why a bunch of accounts are being suspended and probably won't be allowed back on here https://t.co/InbAmAQX3g	0	2	0	2	0	0	2	0
RT @Dr2NisreenAlwan : Wow powerful and very sad! Healthcare professionals in Belgium turning backs as their PM arrives as a demonstration against the handling of the pandemic. Belgium has one of the higest #Covid19 death rates in the world.	0	0	0	0	2	0	2	0
@ocheboy Drove a CVT for a while , didn't actually give me issues .. just maybe my case was peculiar then . If you had even said the steering lock column actuator issues that comes with the push to start ignition system .. I won't argue .. for that I saw things !	2	2	2	2	2	2	4	2
RT @Iamdevinnsmart : God is so good! Was in a car that flipped over three times..... walked away with a sore back but guess what!?!? I'm Alive	0	0	4	4	4	4	4	4

Tabulka A.1: Klasifikátor z této práce je označen K, Sentiment140 je označen jako S. Tweety hodnotilo celkem 5 lidí (A1 až A5) a poté se vzal názor, který byl nejčastější (AC).

Příloha B

Obsah přiloženého paměťového média

```
/
├── classifier.ipynb ....... Jupyter notebook pro natrénování klasifikátoru
├── comparison_results.txt ..... výsledky porovnání klasifikátoru
├── data/ ..... data potřebná pro trénování
│   ├── data_mix.csv
│   ├── GoogleNews-vectors-negative300.bin.gz
│   ├── manually_annoted.tsv
│   ├── testdata.manual.2009.06.14.csv
│   └── twcs.csv
├── Estimation_of_Emotions_from_a_Text.mp4
├── poster.pdf
├── readme.md
├── screenshots/ ..... screenshoty webové aplikace
├── text_source/ ..... zdrojové soubory textu práce
├── text_tisk.pdf .....text práce v PDF formátu pro tisk
├── text.pdf ..... text práce v PDF formátu
├── web_client/
│   ├── css/ .....CSS styly
│   ├── img/ .....obrázky použité ve webové aplikaci
│   ├── index.html
│   └── js/ .....JavaScriptové soubory využité ve webové aplikaci
└── web_server/
    ├── db_init.py .....skript pro inicializaci databáze
    ├── my_model_small.h5 .....předtrénovaný model
    ├── server.py .....Flask server
    ├── tokenizer_small.pickle .....předtrénovaný tokenizer
    └── tweets.db ..... SQLite3 databáze
```