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

METAVYHLEDÁVÁNÍ RECENZÍ NA ČESKÉM WEBU

METASEARCH FOR REVIEWS ON THE CZECH WEB

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ŠIMON MATYÁŠ**
AUTHOR

VEDOUCÍ PRÁCE **doc. prof. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Matyáš Šimon**
Program: Informační technologie
Název: **Metavyhledávání recenzí na českém webu**
Metasearch for Reviews on the Czech Web
Kategorie: Web

Zadání:

1. Seznamte se s možnostmi pravidelného stahování webového obsahu novinových článků, blogů, uživatelských diskusí, případně i sociálních sítí.
2. Prostudujte algoritmy pro extrakci pojmenovaných entit a aspektově-orientovanou analýzu postojů z textu.
3. Shromážděte datovou sadu pro průběžné vyhodnocování vytvářeného systému.
4. S využitím existujících řešení navrhněte a implementujte systém pro metavyhledávání recenzí filmů, knih, případně výrobků a jejich klasifikaci.
5. Zhodnoťte výsledky a diskutujte další možný vývoj projektu.
6. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle doporučení vedoucího

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

Vedoucí práce: **Smrž Pavel, doc. RNDr., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Práce tvoří systém pro vyhledávání recenzí uživatelem zadaného produktu v pravidelně stahovaných článcích z českých novinkových webů a blogů. U vyhledaných recenzí systém rozpozná produkt kterým se článek zabývá a provede analýzu sentimentu.

Abstract

Thesis forms a system for searching reviews of a user-specified product in regularly downloaded articles from Czech news websites and blogs. For searched reviews, the system recognizes the product that the article deals with and performs a sentiment analysis.

Klíčová slova

vyhledání recenze, analýza sentimentu

Keywords

search reviews, sentiment analysis.

Citace

MATYÁŠ, Šimon. *Metavyhledávání recenzí na českém webu.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. prof. RNDr. Pavel Smrž, Ph.D.

Metavyhledávání recenzí na českém webu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Šimon Matyáš

8. května 2021

Obsah

Kapitola 1

Úvod

Je mnoho novinkových webů a blogů zabývajících se recenzováním různých produktů. Tyto recenze se ovšem často k uživateli vůbec nedostanou, pokud nehledá právě recenze konkrétního produktu. Často uživatelé k ohodnocení produktu využívají krátké neprofesionální recenze od ostatních uživatelů, kteří si daný produkt zakoupili před krátkou dobou a portál, přes který si produkt zakoupili, je požádal o hodnocení.

Navíc pokud uživatel hledá recenze k nějakému produktu, musí proklikat několik různých webů, aby si zjistil podrobné informace ke konkrétnímu produktu. Pokud si například uživatel chce zakoupit nový notebook, musí si nejprve najít konkrétní notebook, který ho zajímá a poté si vyhledat články ke konkrétnímu notebooku.

Českých portálů píšících rozsáhlejší recenze je mnoho, ovšem většinou jsou zaměřeny pouze na jeden produkt typ produktu (*svetandroida.cz*, *auto-mania.cz*). Práce se tedy hlavně zaměřuje na vyhledávání článků obsahující rozsáhlejší recenze, ne na krátké uživatelské recenze jako je tomu třeba na webech *heureka.cz* a *zbozi.cz*, tyto servery mají u některých produktů dostupné i rozsáhlé recenze, které ovšem musí k produktu doplnit zaměstnanci ručně. Práci mohou využít tedy i pracovníci těchto portálů k vyhledání recenzí. Pro americká média existuje portál *metacritic.com*, který shromažďuje recenze o filmech, knihách, muzice a počítačových hrách, pro češtinu nástroje pro vytvoření takového systému dosud nebyly dostupné.

Cíl práce je tedy vytvořit systém, který vyhledá články obsahujících recenze na zadanou skupinu produktů, jako jsou například *telefon*, *monitor*, *auto* a je schopen vyhledat i věci, jako jsou třeba *film*, *kniha* a *počítačová hra*. Dále pro vyhledané recenze určí, jestli článek hodnotí produkt pozitivně či negativně.

Při zadání produktu jsou pomocí wikidat vyhledány alternativní názvy produktu a firmy, které daný produkt vyrábí. Podle výrobců, alternativních názvů a slov často se vyskytujících v recenzích je u článkům přiřazena hodnota určující pravděpodobnost, že se jedná o článek hodnotící hledaný produkt. Po ohodnocení všech článků jsou poté seřazeny podle přiřazené hodnoty, je u nich ohodnocen sentiment a jsou vypsány na standardní výstup.

Práce využívá projekt *FeedsCrawler*, který pravidelně každý den stahuje články z českých novinkových webů a blogů.

Kapitola 2 obsahuje teorii témat využitých v práci, jako jsou způsoby, jakými jsou stahovány články z portálů, jakým způsobem jsou zpracovány, je zde také něco o tom, co jsou to wikidata, rozpoznávání typu článku, názvu produktu, kterým se článek zabývá a jakým způsobem probíhá analýza sentimentu. V kapitole 3 jsou popsány data použité k vyhledání recenzí, tvorba datasetu pro testování práce, postupy použité pro rozpoznání

recenzí, relevantního textu v článcích a produktu kterým se recenze zabývá, jakým způsobem jsou využita wikidata, upravené podmínky vyhledávání pro speciální produkty, popis konfiguračního souboru a trénování modelu pro určování sentimentu. Kapitola 4 se zabývá rozborem skriptů a problémy kterými jsem se při tvorbě zabýval. V Kapitole 5 je jak jsem práci testoval a jak vypadají výsledky práce. Kapitola 6 obsahuje zhodnocení výsledků celé práce.

Kapitola 2

Rozbor řešené problematiky

Tato kapitola se zabývá teoretickým rozborem jednotlivých částí systému. V následujících sekcích je rozebrán projekt *FeedsCrawler* využitý pro stahování a zpracování vstupních dat pro tuto práci, informace o tom co jsou to Wikidata a jakým způsobem jsou v nich vyhledávány znalosti, podle čeho se dá poznat, že článek obsahuje recenzi, rozpoznání jakým produktem se článek zabývá a nakonec postupy zhodnocení, jestli autor článku hodnotí produkt negativně či pozitivně.

2.1 Stahování dat

Zdrojová data poskytuje projekt *FeedsCrawler*, který pomocí technologií RSS a ATOM vyhledává a následně stahuje obsahy článků z českých webů. Články jsou staženy v jazyce HTML a pro další práci je potřeba je dále zpracovat, aby bylo možné vyhledání relevantního textu a určeni slovních druhů u jednotlivých slov[24]. Tento projekt stahuje i texty ze sociálních sítí a diskusí, tato práce se zaměřuje na část *cs_media*, která se zabývá na českými deníky a novinkovými portály.

RSS

Rich Site Summary text ve formátu XML, který obsahuje data o článku, umožňuje uživatelům odebírat nové články z několika webů najednou bez toho, aby uživatel musel každý z portálů navštívit a podívat se, jestli není dostupný nějaký nový článek. Vývoj technologie RSS skončil v roce 2003[20][11].

ATOM

Atom Syndication Format nástupce formátu RSS, doplňuje jeho nedostatky a hlavně je standardem IETF, ale jeho podstata je stejná, taky ukládá metadata o článcích ve formátu XML a umožňuje tak odběr nových článků z webu[7][32].

Zpracování HTML

Po stažení článku je třeba stažená data patřičně zpracovat, aby s nimi mohly pracovat nástroje pro určení slovních druhů. Je třeba identifikovat kódování, jaké HTML dokument používá, odstranit z dat přebytky jako jsou třeba reklamy a rozdělit dokument na logické celky vhodné pro další zpracování[24].

Určování slovních druhů

V poslední části zpracování zdrojových dat dochází k rozboru jednotlivých slov textu pomocí českého nástroje *MorphoDita*[1]. V této části jsou určeny základní tvary slov a slovní druhy [24].

2.2 Wikidata

Volně dostupná, kolaborativní a vícejazyčná znalostní báze. Wikidata[2] používá nespočet projektů, Wikipedia je například používá pro úpravu článku ve všech jazycích najednou[38].

Položky

Všechny položky ve Wikidatech mají štítky, krátký popisek, alternativní názvy a jsou označeny unikátním identifikátorem začínajícím písmenem Q a číslem[38].

Vlastnosti

Vlastnosti lze chápat jako kategorie dat. Vlastnosti spojené s hodnotou se nazývají výroky. Každá vlastnost má na wikidatech svou vlastní stránku a může se spojovat s položkami, čímž dochází k vytvoření propojené datové struktury. Vlastnosti ve wikidatech mají identifikátor začínající písmenem P a číslem[38].

Výroky

Výroky popisují dané položky, skládají se z vlastnosti a hodnoty[38].

Například položka *Vincent van Gogh*(Q5582) má ve svých vlastnostech *occupation*(P106) uvedenou hodnotu *painter*(Q1028181). Jako další vlastnosti jsou u této konkretní položky uvedeny i věci jako *work location*(P937) nebo třeba *writing language*(P6886). Vyhledané hodnoty mají také svoje vlastnosti. Na základě těchto výroků je možné si dohledat různé informace jako například, kteří malíři píšící francouzským jazykem pracovali někdy během svého života ve městě s nižší populací než je 1 000 000 obyvatel.

SPARQL

SPARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage je sémantický dotazovací jazyk pro data uchovaná ve formátu RDF[26]. Je uznáván jako klíčová technologie sémantického webu. *SPARQL* umožňuje uživatelům vyhledávání v datech dodržujících RDF specifikaci pomocí dotazů[39][8].

RDF

Resource **D**escription **F**ramework[3] (systém popisu zdrojů) je rodina specifikací vypracovaných organizací World Wide Web Consortium (W3C). Používá se jako obecná metoda pro modelování informací v různých syntaxích. Jedná se o Standardizovaný formát, který umožňuje vyjadřovat popisné informace o zdrojích[22][8].

[1] https://ufal.mff.cuni.cz/morphodita
[2] https://www.wikidata.org/wiki/Wikidata:Introduction/cs
[3] https://www.w3.org/RDF/

2.3 Rozpoznávaní recenze

Určení, jestli se jedná o článek obsahující recenzi může být v některých případech těžké i pro člověka. Článek který má v hlavičce uvedeno například *RECENZE: Samsung Galaxy S20* lze považovat za recenzi i bez čtení zbytku obsahu, ovšem článek který má v názvu *Nový iPhone 12* může obsahovat pouze informace o tom, že společnost Apple představila nový telefon, který bude k získání až za několik měsíců a o recenzi se tudíž nejedná.

Na první pohled se tedy často dá rozlišit, které články by recenzi obsahovat mohli a které ne už z hlavičky článku.

Navíc nestačí rozlišit, jen jestli se jedná o recenzi. Ještě je potřeba zjistit, jestli se článek zabývá hledaným produktem, například v případě hledání článků na téma *telefon* se budou ve správném článku vyskytovat slova jako *mobil*, *telefon* nebo třeba výrobce *Samsung*[19][15].

Metoda klíčových slov

Jedna z nejvyužívanějších metod pro tvorbu vyhledávačů je vyhledávání pomocí klíčových slov. Vyhledávače využívající tuto metodu potřebují k práci databázi webových stránek a slova, pomocí kterých budou v dané databázi vyhledávat[30]. Když uživatel zadá klíčová slova, vyhledávač prohledá databázi webových stránek a podle výskytu hledaných slov určí relevanci hledané stránky, poté uživateli poskytne nalezené výsledky seřazené podle relevance. Tuto metodu například využívá známy vyhledávač *Google*[4][1].

Relevantní text

Pro hledání v obsahu je ještě potřeba vyhledat pouze relevantní text, aby například nebyl článek vyhodnocen špatně, protože je někde na straně v sekci *Určitě si přečtěte* odkaz na recenzi.

Rozpoznání vhodného textu je možné několika způsoby. Pomocí extrakce z HTML kódu článku lze spolehlivě rozlišit, jestli se jedná o relevantní text či nikoli na základě tříd a značek kódu, ze kterého se stránka skládá. Problém tohoto způsobu je ten fakt, že každý portál používá jinou strukturu stránky a bylo by tedy potřeba vytvořit specifický kód pro vyhledávání na každý zpracovávaný web[21].

Další možný způsob rozlišení relevantního obsahu je pomocí délky textu jednotlivých odstavců. Popis vedlejších článku je obvykle tvořen krátkým popisem a titulkem, který odkazuje na danou stránku. Menu webu většinou také obsahuje odkazy a krátké texty, takže tímto způsobem není problém tuto sekci odstranit. Tento způsob využívá i nástroj *justext*[5] využívaný v této práci. Tento nástroj vhodný text vyhledává na základě délky textu, hustoty odkazů v odstavci, ale navíc pro detekci používá i polohu odstavce v článku, počet slov, které nenesou žádnou informací a další způsoby pro vyhledávání relevantního textu[28][31][19].

2.4 Rozpoznání názvu produktu

Je třeba také v článku detekovat, o jaký produkt se jedná, podobně jako u rozpoznání recenze, se tohle dá zjistit už z hlavičky článku. Často je název produktu uveden právě

[4] https://www.google.com/
[5] http://corpus.tools/wiki/Justext

v titulku článku. Ve speciálních případech například u filmů a knih někdy v hlavičce článku nejde jednoduše rozpoznat, co je názvem filmu a co už ne[23].

Rozpoznávání pojmenovaných entit

Úkol zabývající se vyhledáním pojmenovaných položek v nestrukturovaném textu. Součástí tohoto úkolu je většinou ještě rozřazení vyhledané položky do předem definovaných kategorií, jako jsou například *Jména osob*, *lokace* nebo *organizace*[41][34].

Postupy rozpoznání entit

Systémy zabývající se tímto úkolem používají různé způsoby rozpoznání entit[41].

Jazykové gramatické techniky - Ručně vytvořené gramatické systémy, většinou poskytují vyšší přesnost, ale také tvorba takového systému vyžaduje víc práce.

Statistické modely - Modely tvořené pomocí strojového učení. Tvorba takového modelu vyžaduje velké množství dat s ručně pojmenovanými produkty.

2.5 Určení sentimentu

Analýza sentimentu se snaží určit názor autora, který recenzi napsal. Vstupem je tedy čistý text recenze, který je vyhodnocen jako pozitivní či negativní pomocí modelu vytrénovaného na textech recenzí, kterým byl sentiment ručně přiřazen.

Určení sentimentu může být v některých případech těžké i pro člověka. Jednoduchou větu *Film se mi velice líbil*, není problém ohodnotit, obsahuje slovo *líbil*, takže se dá považovat za pozitivně hodnotící text. U vět jako je třeba *Film nelze hodnotit špatně*, které sice obsahují negativní slova, ale jedná se o větu pozitivní, je určení sentimentu už obtížnější. Pro člověka toto není problém určit, ale počítač na to potřebuje nástroje složitější než jednoduché vyhledání pozitivních či negativních slov ve větě[4]. Tento typ úloh je řešen pomocí následujících metod[6].

Klasifikace pomocí slovníků

Slovníkové metody určují sentiment pomocí předem připravených slovníků. Tyto slovníky obsahují slova s již přiřazeným sentimentem. Slova ve slovnících jsou roztříděna na pozitivní, negativní a neutrální[6][18][35].

Klasifikace pomocí strojového učení

Pro určování sentimentu existuje mnoho volně dostupných bezplatných nástrojů využívající strojové učení, techniky zpracování přirozeného jazyka a statistiky pro určování analýzy ve velkých kolekcích textů, novinkových webech i sociálních sítích[12]. Analýzu sentimentu není omezena pouze na zpracování textu, ale jde provést i u obrázků a videí.

Modely využívající strojového učení potřebují nějaký zdroj dat, na kterých se naučí, jak data ohodnotit a určit, jestli se jedná o pozitivní či negativní text. Učení modelu jsou dva typy, které jsou rozvedeny v následujících odstavcích[18][12][6].

Učení s učitelem

Trénovací sada spočívá v datech, které mají manuálně přiřazený požadovaný výsledek. V případě hodnocení recenzí je tedy třeba text nejprve ručně ohodnotit, jestli se jedná o pozi-

tivní či negativní sentiment a dvojici textu a požadovaného výsledku vložit do trénovací sady která poté bude využita k naučení hodnotícího modelu jak vypadá text pozitivní či negativní[2].

Učení bez učitele

Učí se z neoznačených dat a snaží se na základě podobnosti dat vytvořit shluky. Příklad využití tohoto způsobu v oblasti vyhledávání článků by bylo například přiřazení kategorie článku na základě výskytu slov v obsahu a podobnosti s články se stejnými slovy. Články by tak byly rozřazeny do kategorií, jako je například *Sport*, *Technologie* či *Móda* [2][14].

Modely strojového učení

Model ve strojovém učení označuje soubor vytvořený trénováním pomocí nějaké metody[3].

Transformer

Model představený v roce 2017, který váží vliv různých částí vstupních dat. Jeho primární využití je v oblasti zpracování přirozeného jazyka. Tento model se krátce po svém představení stal oblíbeným kvůli možnosti paralelního zpracování a tím urychlení tréninku. Umožnil tak trénovat na větších datových sadách než starší modely. To vedlo k vývoji předcvičených systémů, jako jsou BERT a GPT, které byly školeny na velkých obecných datových souborech, jako jsou Wikipedia Corpus a Common Crawl, a mohou být doladěny pro specifický úkol[37].

Metody strojového učení

Metody jsou ve strojovém učení postupy, které pomocí zadaných vstupních dat vytvoří model strojového učení[3].

BERT

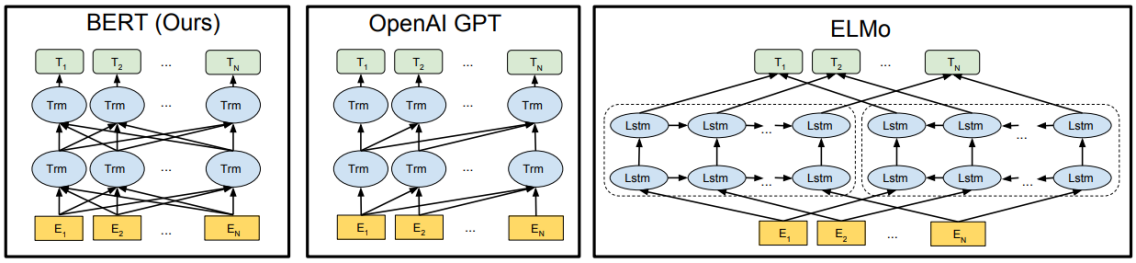
Bidirectional Encoder Representations from Transformers je metoda pro zpracování přirozeného textu vyvíjena společností *Google AI*. Když byla v roce 2018 publikována, způsobila v oblasti strojového učení rozruch svými přesnými výsledky v široké škále úkolů zabývající se zpracováním přirozeného jazyka[17]. Tato metoda se snaží naučit kontextové vztahy mezi slovy v textu a pomocí získaných informací vytvořit model schopný pochopit kontext věty. Využívá modelu Transformer[37] pro zpracování vět a odhaduje tak kontext podle slov následujících i předchozích. Model vytvořen metodou BERT je trénován maskováním 15ti % tokenů s cílem snažit se uhodnout zamaskovaný token. Dále se snaží předpovědět následující větu[9][36][10]. Porovnání s ostatními metodami je vyobrazeno na obrázku 2.1.

OpenAI GPT

Generative **P**re-trained **T**ransformer je před trénovaný jazykový model založený na transformátorech, nasledovaný doladěním pro specifický úkol[29][10]. Model je trénován pouze pro odhad následujících slov ve větě, pracuje tedy s textem pouze *zleva doprava*.

ELMo

Embeddings from **L**anguage **Mo**del (Vkládání z jazykového modelu) je metoda využívající obousměrný jazykový model, který je předem natrénován na velkých textových korpusech, aby se naučil význam slov a jejich lingvistický kontext[27][10].



Obrázek 2.1: Rozdíly v architekturách před trénovaných modelů. **BERT** používá oboustranný Transformer, **OpenAI GPT** používá pouze transformer *zleva doprava*. **ELMo** používá zřetězení nezávisle trénovaných LSTM (Long short-term memory) *zleva doprava* a *zprava doleva* pro generování funkcí pro následné úkoly. Z těchto tří architektur je pouze reprezentace BERT v levém i pravém kontextu ve všech vrstvách. Architektury BERT a OpenAI GPT mají doladovací přístup, zatímco ELMo má přístup založený na funkcích. Převzato z [9].

Aspektově orientovaná analýza

Další způsob určení polarity článku je možný na základě aspektů. Například věta *Baterie vydrží dlouho, ale Hlasitost je nízká.* obsahuje dva aspekty, konkrétně *Výdrž Baterie* a *Hlasitost* produktu. Z textu je třeba rozpoznat tyto aspekty, k nim přiřadit slova je hodnotící a určit, jestli je aspekt ohodnocen pozitivně či negativně[25].

Aspektově orientovaná analýza neprovede pouze rozlišení recenzí na pozitivní či negativní, ale navíc může sentiment roztřídit do kategorií, které jsou hodnoceny. To lze využít například u zjištění, co se konkrétně zákazníkům nelíbí při hodnocení nějakého zařízení, je tak možné vyhledat rozdíl jestli se recenzentovi nelíbí hlučnost zařízení, ale líbí se mu třeba vzhled[18].

Vyhledání aspektů

Aspekty jdou získat pomocí modelu vytrénovaného strojovým učením. Tento způsob je vhodný pro vyhledávání informací stejného typu, například při zpracování recenzí hodnotící restauraci se budou často vyskytovat aspekty podobného typu jako například *Jídlo* nebo třeba *Personál*. Systém se naučí, jak tyto aspekty rozpoznat a poté může v textu identifikovat, že se jedná o hodnocení jídla, když recenzent napíše *Nic mi nechutnalo*. Nevýhoda tohoto způsobu je, že potřebuje rozsáhlou datovou sadu, na které by se naučil aspekty rozpoznávat[40].

Další způsob získání aspektu je pomocí gramatického zpracování textu, zejména určením stavby věty a závislostí jednotlivých slov ve větě. Tento způsob má výhodu v tom, že není potřeba dopředu určit, jaké aspekty nás zajímají a systém sám vyhledá, jaké aspekty jsou ve větě obsaženy a jaké slova je hodnotí[5].

Větná stavba

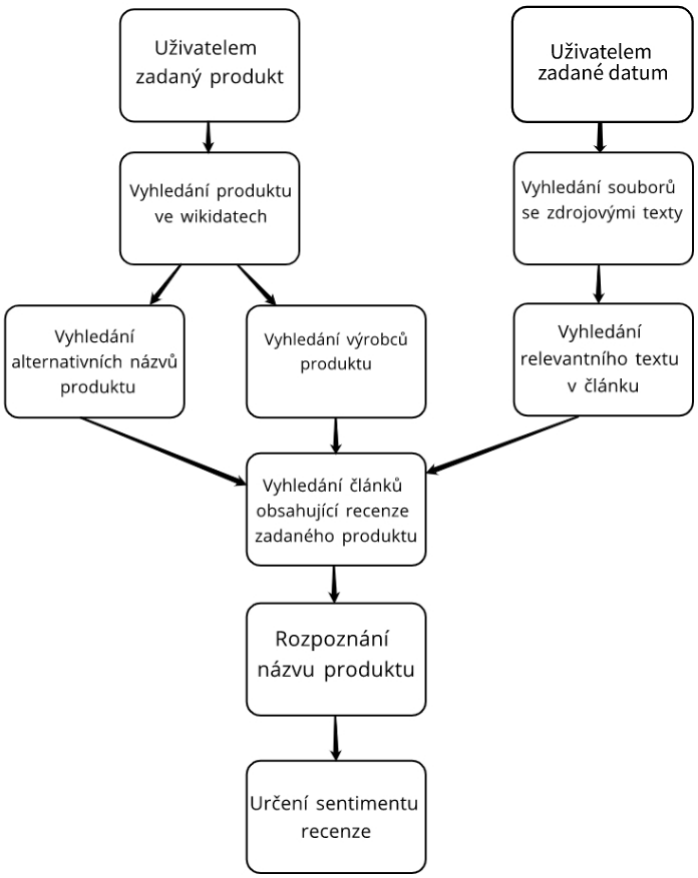
Pro určení aspektů pomocí větné stavby je třeba větu rozložit na jednotlivá slova, určit u nich gramatické informace, jako jsou slovní druhy a závislosti ve větě. Pro tento úkol je v práci využit český nástroj *UDPipe*[6]. Tento nástroj provede rozdělení věty na jednotlivá slova, přiřadí slovní druhy a určí závislosti slov ve větě.

[6] https://ufal.mff.cuni.cz/udpipe/2

Kapitola 3

Návrh řešení

V této kapitole jsou popsány jednotlivé části celé práce 3.1, je zde uvedeno s jakými daty která část pracuje a jejich bližším rozborem. Využití znalostní báze Wikidat v práci, jakým způsobem je v textu vyhledán relevantní text, tvorba datové sady, na které byla později práce testována. Postup, jakým způsobem jsou články hodnoceny, jestli se jedná o recenzi či nikoli, rozpoznání názvu produktu. Jaké jsou v práci speciální případy a co je u nich jiného, popis konfiguračního souboru a postup, jakým byl určován sentiment článků.



Obrázek 3.1: Návrhu systému

3.1 Dostupná data

Práce používá data zpracované projektem *FeedsCrawler*, které jsou uloženy na serveru `minerva3` od výzkumné skupiny `KNoT`. Přesná poloha dat je:

`/mnt/data-2/feeds_crawling/big_brother/workplace/4-tagged/cs_media/`

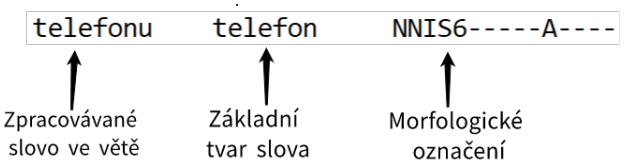
Skripty této práce budou hledat vstupní data v tomto adresáři, je tedy třeba spustit práci z tohoto serveru. Pro využití dat z jiného adresáře je třeba upravit údaj v konfiguračním souboru udávající polohu zdrojových dat.

Soubory vstupních dat

Soubory jsou rozděleny po dnech, takže v každém souboru jsou obsaženy všechny články vydané v jeden den. Jednotlivé články souboru jsou rozděleny specifickými značkami. Obsah každého článku je dále pomocí dalších značek rozdělen na menší části, jako titulek, odstavce a jednotlivé nadpisy. V prvním řádků každého článku je obsažena URL adresa článku, titulek a unikátní identifikátor. Podrobnější informace k využitým značkám jsou uvedeny v práci zabývající se tvorbou projektu *FeedsCrawler*[24].

Zpracovaný text

Na následujícím obrázku je ukázán příklad jednoho řádku ze zpracovaných dokumentů. Jsou zde vyznačeny pouze informace důležité pro tuto práci, hodnoty jsou od sebe odděleny tabulátory. Na obrázku 3.2 je zobrazen příklad jednoho řádku ze vstupního souboru.



Obrázek 3.2: Ukázka jednoho řádku ze zpracovávaných dat

Morfologické označení

Tuto hodnotu přiřadí každému slovu nástroj *MorphoDita*, který používá označování podle Českého Morfologického systému od Jana Hajiče [16]. Toto označení se skládá z 15ti pozic odpovídající části řeči, pohlaví, číslu, velikosti písmen atd[1]. Pro tuto práci je nejdůležitější značka `NN`, která indikuje, že se jedná o podstatné jméno.

Výběr dnů ke zpracování

Zpracování všech dnů od konce roku 2019, kdy projekt *FeedsCrawler* začal stahovat články, by bylo časově náročné. Uživatel má možnost vybrat si dny, ve kterých se budou recenze vyhledávat. Výběr dnů ke zpracování je proveden při spuštění skriptu, uživatel zadá datum prvního a posledního dne a soubory nalezené mezi těmito daty jsou určeny ke zpracování. Pokud je počáteční i konečné datum stejné je zpracován pouze jeden den.

[1] https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/html/ch02.html

3.2 Použití wikidat

Pro vyhledávání celé skupiny produktů je třeba najít alternativní názvy zadaného produktu, které se mohou v hledaném článku vyskytnout, navíc jsou vyhledány firmy, které se daným produktem zabývají, protože jsou většinou také v článcích zabývajících se hledaným produktem zmíněny.

K vyhledávání těchto informací jsou použity Wikidata 2.2, ve kterých práce vyhledává pomocí jazyku *SPARQL*.

Alternativní názvy produktu

Podle zadaného produktu je ve Wikidatech podle názvu nalezena správná položka a jsou staženy její alternativní názvy v Českém a Anglickém jazyce. Angličtina je stažena kvůli tomu, že se některé názvy moderních výrobků používají i v česky psaném textu, například se často používá *smartphone* na místo *chytrý telefon*.

Výrobci produktu

Z názvu zadaného produktu je získán identifikátor výrobku. Poté jsou vyhledány všechny položky, které mají vlastnost `product or material produced`(P1056) a jedna z hodnot této vlastnosti je právě ID produktu. U položek, které odpovídají tomuto výroku, je vybráno jednoslovné označení z oficiálního či alternativních názvů, pokud je dostupné.

Jednoslovný název je lepší kvůli tomu, že v článcích není většinou uveden celý oficiální název firmy jako třeba u *Volvo Cars* je většinou uváděno pouze *Volvo*.

3.3 Vyhledání relevantního textu

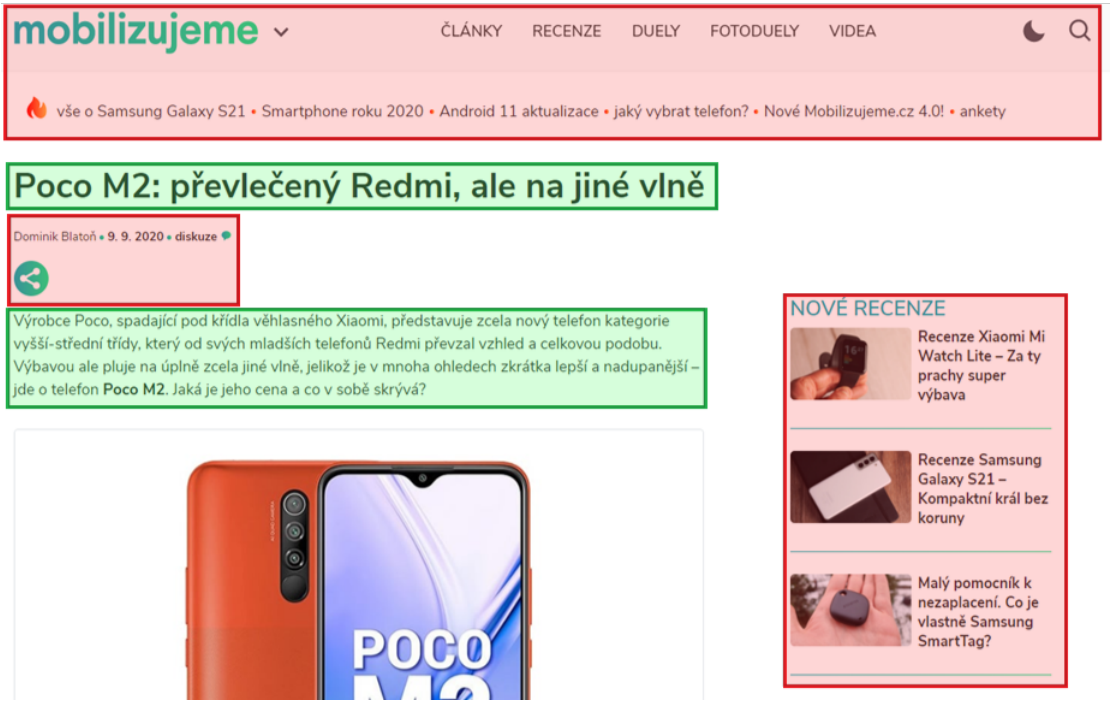
Pro správné rozpoznání recenze je třeba vybrat pouze nadpisy a obsah článku3.3 kvůli tomu, že se v textu často vyskytují odkazy na podobné články nebo je někde na stránce sekce *Mohlo by vás zajímat*, která obsahuje mnoho dalších odkazů. Práce se tedy snaží vybírat pouze relevantní text. Odstranění reklam proběhne již při zpracování vstupních dat, dále je tedy potřeba odstranit menu webu, odkazy na podobné články a nepotřebné informace jako například jméno autora článku.

Relevantní text se nachází v hlavičce článku, nadpisech, podnadpisech a samotných odstavcích. Vyhledání hlavičky a nadpisů není problém, protože jsou ve zpracovaných datech označeny specifickými značkami pro hlavičku, nadpisy a podnadpisy.

Pro vyhledání správných odstavců je použitý počet vět v odstavci, ze kterých se text skládá. Věty v odstavcích jsou oddělené značkami, není tedy problém zjistit jejich počet. Sekce se souvisejícími články někdy mají krátký popisek obsahu, většinou tyto krátké popisky mají jednu dlouhou či dvě kratší věty.

3.4 Tvorba datasetu pro testování

Pro tvorbu datasetu jsem manuálně prošel články ze dnů 10.09.2020 - 16.09.2020 ve kterých je dohromady stažených 18516 článků. Zaměřoval jsem se na vyhledávání článků obsahující recenze. Celkově jsem tak našel 80 článků, které jsem označil jako recenze. Vyhledané recenze jsem rozřadil do několika kategorií z nich nejčastěji byly recenze telefonů a automobilů.



Obrázek 3.3: Červeně je vyznačen nerelevantní text, který je v článku přebytečný a je třeba ho odstranit, Zeleně je označen obsah článku důležitý pro tuto práci. Část obrázku převzata z portálu *mobilizujeme.cz*

Kategorie a jejich počty

Články rozřazené do kategorií a seřazené podle četnosti.

- telefon: 25
- auto: 22
- notebook: 7
- hardware: 6
- sluchátka: 5
- film: 5
- videohra: 4
- kniha:4

3.5 Rozpoznání recenze

Při tvorbě datasetu jsem si všímal, podle čeho recenze identifikuji a zjistil, že ve většině případů jde rozpoznat, jestli se jedná o recenzi nějakého produktu už z adresy url článku. Článek s adresou *https://mobilenet.cz/clanky/recenze-motorola-moto-g-5g-plus-stredni-trida-s-e-vsim-vsudy-41676* lze označit za recenzi i bez otevření článku. Stejně jako u url adresy

https://www.sport.cz/ostatni/basketbal/clanek/1663829-hrac-nba-mel-navstevu-na-pokoji-za-trest-byl-vyloucen.html#rss jde vidět, že se o recenzi jednat nebude. Pak jsou ještě články, u kterých z adresy nejde zjistit, o čem se v něm píše, protože v URL neuvádějí část hlavičky, ale pouze unikátní identifikátor jako například *http://www.halonoviny.cz/articles/view/54232217*

I přes to, že se dá recenze často identifikovat už z url adresy, jsou zpracovávány celé články. Při zpracování skript prochází jednotlivé odstavce a v případě, že je zpracovávaný text označen za relevantní, jsou zpracována jednotlivá slova odstavce. V případě, že je slovo označeno jako podstatné jméno je porovnán jeho základní tvar se seznamem alternativních názvů produktů, výrobců a slov označující recenze. Pokud je základní tvar slova v některém seznamu z hledaných slov, je článku přičteno skóre které znázorňuje podobnost s textem obsahující recenzi hledaného produktu.

Skóre článků

Hodnota, která je článku přičtena při nalezení slova v některém ze seznamů, záleží na několika aspektech, jako ve kterém ze seznamů bylo slovo nalezeno a taky, ve kterém odstavci. Pokud je například slovo *Recenze* uvedeno v hlavičce článku, je přičtena vyšší hodnota, než když je nalezeno jméno produktu v obsahu článku. Přesné případy a hodnoty jsou uvedeny v sekci 4.3.

Výsledné články

Po projití všech odstavců v článku je podle skóre vyhodnoceno, jestli článek bude přidán do výsledků. Do výsledků je přidán článek, který v textu či hlavičce obsahuje alespoň jedno slovo ze seznamu alternativních názvů, jedno slovo ze seznamu definující recenze a skóre vyšší či rovno hodnotě 30.

Když je článek označen jako recenze, proběhne rozpoznání názvu produktu. Po prohledání všech vstupních souborů jsou výsledné články seřazeny podle skóre a je u nich vyhodnocen sentiment.

Rozpoznání názvu produktu

Název je rozpoznán na základně jednoduchých gramatických pravidel. Vyhledávání názvu pomocí strojového učení by pro tuto práci vyžadovalo velkou datovou sadu, aby se model naučil rozpoznávat produkty všech typů.

Protože bývá název produktu, kterým se článek zabývá uveden v hlavičce článku, použil jsem pro identifikaci názvu produktu jednoduchou extrakci slov z titulku. Jsou vybrány slova začínající velkými písmeny a číslovky, navíc jsou odstraněny slova, podle kterých bylo určeno, že se jedná o recenzi. Vybraná slova jsou poté porovnána s url adresou a jako název označena pouze ty, která se vyskytují v url adrese i titulku článku.

Název produktu je poté ověřen vyhledáním produktu na portálu *heureka.cz* a porovnáním prvních výsledků vyhledávání.

3.6 Speciální případy

Pro některé typy produktu byly přidány speciální podmínky rozpoznání názvu. Tyto speciální případy jsou hlavně pro produkty, kde rozpoznání na základě velkých písmen není možné, jako například u knih, filmů a počítačových her. Tyto případy mají předem určené

weby, na kterých hledat jaký produkt. Tímto způsobem je tedy možné pro každý portál předem určit způsob vyhledání názvu, protože si weby drží stejný formát HTML pro určitý typ produktu, lze hledaný název vyhledat pomocí HTML kódu.

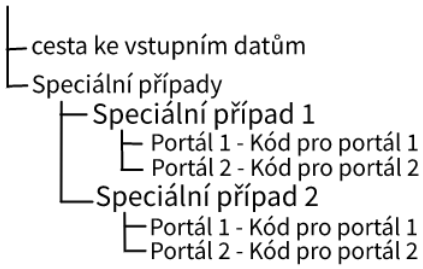
Speciální případy s weby, na kterých se mají tyto produkty hledat a kódem pro každý web jsou uloženy v konfiguračním souboru. Poté, co jsou vyhledány alternativní názvy hledaného produktu je prohledán tento soubor, jestli jeden z alternativních názvů není speciálním případem, pokud ano je s produktem zacházeno, jako se speciálním to znamená, že recenze budou vyhledávány pouze na portálech uvedených u daného speciálního případu a pro hledání nazvu produktu bude využit předurčený kód pro každý z webů.

3.7 Konfigurační soubor

Tento malý textový soubor poskytuje hlavnímu skriptu informace důležité pro chod systému. Obsahuje cestu ke vstupním datům, se kterými skript pracuje a podmínky zpracování speciálních případů.

Konfigurační soubor je napsán formou slovníku jazyka *Python*, je tedy potřeba dodržovat přesný formát zápisu.

Každý případ, který by měl mít speciální podmínky hledání názvu produktu, musí mít uvedené klíčové slovo určující, že se jedná o speciální případ. Vypsané portály, na kterých se budou články vyhledávat a kód, pomocí kterého se v článku najde název produktu. Rozložení tohoto souboru je vyobrazeno na obrázku 3.4.



Obrázek 3.4: Rozložení konfiguračního souboru

Zápis kódu

Výsledek kódu pro hledání názvu je třeba zapsat do proměnné `ArticleProduct` a celý kód vložit do trojitých uvozovek. Pro vyhledávání v HTML lze použít proměnnou `page_soup`, která obsahuje HTML stránky zpracované nástrojem *BeautifulSoup*.

3.8 Určení sentimentu

U článků je ohodnocen sentiment autora recenze, hlavně jestli recenzent hodnotí produkt pozitivně či negativně. Primárně je sentiment určován na základě aspektů. V textu zpracovávaného článku jsou vyhledány aspekty produktu a jejich hodnocení, poté je určeno, jak jsou jednotlivé aspekty ohodnoceny. Pokud se v textu nezdaří vyhledat žádné vhodné

slova na základě kterých by šel aspekt ohodnotit, je provedena analýza sentimentu pomocí modelu BERT společně s knihovnou *Transformers* od společnosti *Huggin Face*[2].

Aspektová analýza

Pro vyhledání aspektů jsem zvolil způsob určování na základě větné stavby. Vyhledávání na základě strojového učení není v tomto případě vhodné, protože práce musí zpracovávat různé produkty a pro naučení modelu, který by spolehlivě vyhledával aspekty ve všech produktech, by bylo potřeba vytvořit velkou datovou sadu.

Sentiment je určován až poté, co je článek vyhodnocen jako recenze. Aby byly aspekty vyhledávány pouze v relevantním textu, je obsah zpracovávané stránky stažen a zpracován nástrojem *justext*, který vyhledá text vhodný ke zpracování. Tento text je poté zpracován knihovnou *udpipe*. Tato knihovna obsah článku rozdělí na jednotlivé věty a určí větnou stavbu 3.5. Jednotlivá slova mají přiřazeny slovní druhy a vztahy mezi slovy. Ze slov je odstraněna interpunkce a slova, která nenesou žádné informace. Dále jsou na základě slovních druhů a vztahů mezi slovy jsou vyhledány aspekty a jejich hodnocení.

Sentiment slov hodnotící aspekty je určen pomocí lexikonů negativních a pozitivních slov. Po prohledání celého textu je porovnán počet pozitivně a negativně ohodnocených aspektů, na základě vyššího počtu je pro celý článek určen sentiment.

Využité lexikony

Aspektová analýza využívá pro určení sentimentu tři lexikony, první dva se zabývají určením, jestli je vyhledaný aspekt ohodnocen pozitivně či negativně. Poslední lexikon obsahuje takzvané *Stopslova*, což jsou slova, která ve větě neobsahují žádné informace a nejsou tak pro určení aspektu či hodnocení důležitá, typicky se jedná o spojka či předložky, v českém jazyce jsou to slova například *aby*, *ale*, *být*.

Lexikon pro rozlišení negativních a pozitivních slov byl poskytnut společností *Kaggle*[3], která se zabývá tvorbou volně dostupných datových sad pro využití při strojovém učení.

Slovník obsahující stopslova byl poskytnut webem *countwordsfree.com*[4], který se zabývá počtem slov v textech určitého jazyka a tvoří tak datové sady stopslov pro různé jazyky.

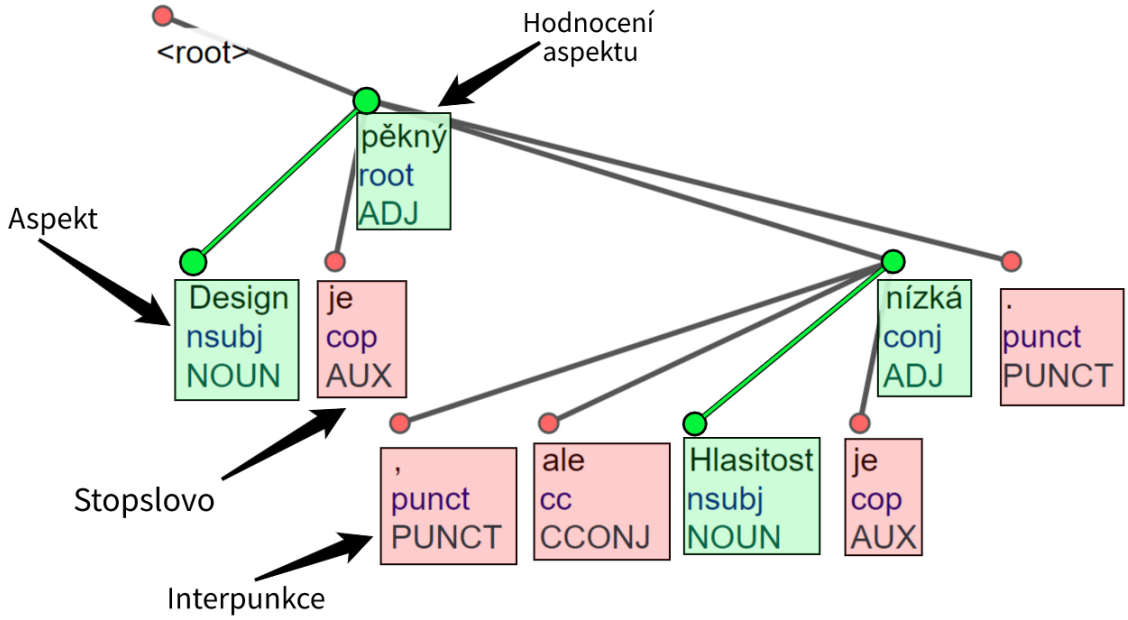
Určení sentimentu pomocí modelu BERT

Pro určení sentimentu byl použit model BERT společně s knihovnou *Transformers* od společnosti *Huggin Face*. Tento způsob jsem zvolil, protože má dostupný mnoho předtrénovaných modelů včetně několika modelů podporující český jazyk.

Zpracování vstupního textu

Modely určující sentiment pracují pouze s čísly. Před tím, než bude u článku určen sentiment. Je nejprve potřeba vstupní text zpracovat do číselné podoby, se kterou umí modely pracovat. Pro převod do číselné podoby lze využít modely společnosti *Huggin Face*. Konkrétně jsem použil model *DeepPavlov/bert-base-bg-cs-pl-ru-cased* určený pro zpracování slovanských jazyků, tento model byl trénován na článcích Wikipedie pro Bulharský, Polský, Český a Ruský jazyk a také na Ruských novinkových portálech.

[2] https://huggingface.co/
[3] https://www.kaggle.com/
[4] https://countwordsfree.com/



Obrázek 3.5: Ukázka zpracování věty nástrojem *udpipe*. Zeleně jsou vyznačeny aspekty a jejich hodnocení. Červeně jsou označeny části věty odstraněné z rozpoznání aspektů. Část obrázku převzata z online ukázky nástroje *udpipe*

S využitím tohoto modelu je vstupní text převeden na seznam čísel, se kterým dále pracuje model vytrénovaný na určování sentimentu.

Ohodnocení sentimentu

Sentiment je určen pomocí modelu natrénovaného na článcích, u kterých bylo ručně určeno, jestli se jedná o pozitivně či negativně hodnotící recenzi. Vyhodnocení sentimentu počítá i s aspektem podnadpisů, kterým dává vyšší váhu na výsledek ohodnocení článku.

Skript obsahuje funkci, která ohodnotí článek ze zadané URL adresy pomocí vytrénovaného modelu, nejprve je stažen obsah stránky, který je následně zpracován nástrojem *justext*. Tento nástroj vyhledá relevantní text vhodný k ohodnocení, odstraní související články a nepotřebné informace, takže ze stránky získáme pouze text vhodný k ohodnocení a podnadpisy článku. Text je poté rozdělen podle odstavců a u každého odstavce a podnadpisu je určen sentiment. Po ohodnocení všech vhodných textů je jejich výsledek zprůměrován a podle průměru je určeno, jestli se jedná o pozitivní či negativní článek.

Trénování modelu

Model byl trénován na článcích vyhledaných při tvorbě datasetu pro testování. U těchto článků bylo poté manuálně určeno, jestli recenze obsahuje pozitivní či negativní hodnocení. Dále byl článek ručně rozdělen podle odstavců a u každého odstavce určen sentiment na základě celého článku. Takže pokud recenze hodnotila produkt pozitivně, všechny odstavce v textu byly označeny za pozitivní.

Model se zaměřuje na profesionálně napsané recenze, takže se dá předpokládat, že články nebudou obsahovat pravopisné chyby, se kterými by mohly být problémy při hodnocení recenzí psaných uživateli.

Kapitola 4

Realizace

Práce je psaná ve skriptovacím jazyce Python 3.6.9 a počítá s tím, že je spuštěna na serveru minerva3 výzkumné skupiny KNOT. Skripty musí být spuštěny z tohoto serveru, protože jsou zde dostupná zpracovaná data projektu *FeedsCrawler*. Na serveru minerva3 lze skripty spustit z domovském adresáře projektu `/mnt/minerva1/nlp/projects/sentiment10`

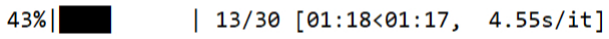
V této kapitole jsou podrobně popsány funkce jednotlivých skriptů, a jak fungují. Práce využívá následující skripty a soubory:

- `brandFinder.py` - ve wikidatech vyhledá alternativní názvy pro zadaný produkt a výrobce zadaného produktu.
- `findReviewArticles.py` - hlavní skript, vyhledá články mezi dvěma daty obsahující recenze zadaného produktu.
- `productFinder.py` - potvrzení, že zadaný produkt je platný.
- `SentimentAnalyzerAspect.py` - analýza jestli je vyhledaná recenze pozitivní nebo negativní.
- `udpipeParse.py` - vyhledání aspektů a jejich hodnocení pomocí větné stavby.
- `morphoditaTagger.py` - doplnění textu o slovní druhy a základní tvary slov pro přesnější vyhledávání recenzí.
- `config.txt` - konfigurační soubor.

Při spuštění hlavního skriptu `findReviewArticles.py` jsou nejprve vyhledány alternativní názvy zadaného produktu a firmy, které se výrobou daného produktu zabývají. Proběhne čtení konfiguračního souboru, vyhledání vstupních souborů mezi zadanými daty a paralelní vyhledání recenzí ve vstupních souborech. Jakmile jsou vyhledány recenze je u nich určen sentiment a jsou vypsány na standardní výstup a seřazeny podle pravděpodobnosti, že se opravdu jedná o článek zabývající se recenzí daného produktu. Články jsou seřazeny sestupně, takže na posledním řádku je článek nejvíce se podobající recenzi.

Průběh jednotlivých kroků, které skripty vykonávají, je podrobně popsán v následujících sekcích.

V průběhu skriptu je postup aktuálního zpracování ukazován na chybovém výstupu pomocí knihovny *tqdm*. Výpis postupu 4.1 poskytuje informace o procentuálním postupu práce s grafickým znázorněním, počet již zpracovaných souborů, délka zpracování, předpokládaný čas do dokončení práce a průměrná délka zpracování jednoho souboru.



Obrázek 4.1: Ukázka výpisu na chybovém výstupu v průběhu zpracování.

4.1 Použití wikidat

Wikidata jsou využita pro vyhledávání alternativních názvů produktu a firem či výrobců, kteří se daným produktem zabývají. Pro vyhledávání je použit dotazovací jazyk *SPARQL* a pro jeho používání v jazyce Python práce používá modul `SPARQLWrapper`, dále je pro vyhledávání využita API Wikidat. Výsledky hledání ve Wikidatech jsou vráceny ve formátu `json`. Tato část práce probíhá ve skriptu `brandFinder.py`.

Názvy produktu

Vyhledání názvů probíhá ve funkci `findProductNames(product)`, vstupní parametr je název produktu, pro který se mají vyhledat alternativní názvy. Zadaný řetězec je poté vyhledán ve Wikidatech voláním API wikidat na adrese *https://www.wikidata.org/w/api.php* s parametry:

```
params = {
    'action': 'wbsearchentities',
    'format': 'json',
    'language': 'cs',
    'search': query
}
```

poslední parametr `search` má jako hodnotu proměnnou `query`, která obsahuje řetězec `product` vložený jako vstupní parametr funkce.

Tento příkaz ve Wikidatech vyhledá hledaný řetězec. Skript si dále uloží identifikátor první vyhledané položky do proměnné `WikidataID`.

Pomocí identifikátoru je vyhledána konkrétní položka a do seznamu `result` vloženo označení položky v českém a anglickém jazyce a také všechny alternativní názvy v těchto jazycích. Nakonec je ještě tento seznam prohledán cyklem, který vyhledává slova obsahující diakritiku, pokud je takové slovo nalezeno, je do seznamu přidáno ještě jednou bez diakritiky.

Po dokončení těchto kroků funkce vrací seznam `result` obsahující označení a alternativní názvy zadaného produktu v českém a anglickém jazyce.

Názvy firem a výrobců

Vyhledání názvů firem a výrobců je záležitost funkce `findBrands(product)` tato funkce, stejně jako u vyhledání alternativních názvů nejprve vyhledá položku zadanou vstupním parametrem ve Wikidatech pomocí API wikidat pro získání identifikátoru položky, který je opět uložen do proměnné `WikidataID`.

Dále jsou pomocí následujícího dotazu v jazyce SPARQL vyhledány položky zabývající se výrobou hledaného produktu. Dotazy SPARQL lze otestovat online[1].

[1] https://query.wikidata.org/

```
SELECT ?item ?itemLabel(GROUP_CONCAT(DISTINCT(?altLabel);separator=",")
AS ?altLabel_list)
WHERE {
  ?item wdt:P1056 wd:"""+WikidataID+""" .
  OPTIONAL {?item skos:altLabel ?altLabel .FILTER(lang(?altLabel)="en")}
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" .}
}
GROUP BY ?item ?itemLabel
```

Tento dotaz vyhledá všechny položky, které mají vlastnost *product or material produced* (P1056) a v této vlastnosti uvedenou položku s identifikátorem uvedeným v proměnné `WikidataID`. Z položek, které odpovídají tomuto výrazu, je uloženo označení položky a její alternativní názvy. Výsledek je ve formátu `json`.

item	itemLabel	altLabel_list
Q6686	Renault	Groupe Renault, Renault S.A., Renault SA
Q6742	Peugeot	Peugeot Automobiles
Q6746	Citroën	Automobiles Citroën S.A., Citroen
Q29637	Škoda Auto	Skoda, Automobilový koncern ŠKODA a.s., ŠKODA

Tabulka 4.1: Část výsledku vyhledání výrobců pro položku "automobil"(Q1420)

V případě, že je označení výrobce víceslovné, jsou prohledány alternativní názvy, jestli se v nich nenachází název jednoslovný, který je pro vyhledávání v článcích lepší, protože jsou články rozděleny na jednotlivá slova a víceslovný název by tak nebyl detekován.

Vyhledané názvy jsou vloženy do seznamu `brandList`, který je po prohledání všech výrobců vrácen jako výsledek funkce.

4.2 Výběr souborů ke zpracování

Soubory ke zpracování jsou vybrány na základě uživatelem zadaných vstupních argumentů a konfiguračního souboru. Nejprve je vytvořen seznam všech souborů adresáře zadaného parametrem `dataSource` v konfiguračním souboru. Poté jsou data zadané uživatelem zpracovány knihovnou `dateTime` a v seznamu všech souborů vyhledány dny nacházející se v intervalu dvou dat, které zadal uživatel.

Formát dat

Datum je třeba zadat ve specifickém formátu, aby s ním knihovna `dateTime` byla schopna dále pracovat. Vstupní soubory jsou zadané ve formátu **[YYYY]-[mm]-[dd]**, zvolil jsem tedy tento formát i pro argumenty při spuštění skriptu. Datum je třeba zadat včetně pomlček mezi hodnotami, příklad správně zadaného data: `2021-3-20` = 20. března 2021

4.3 Určovaní recenzí

Vyhledání recenzí probíhá ve funkci `readReviews(link)`, v této funkci je vytvořena proměnná `reviewWordsListPart`, která obsahuje řetězce, podle kterých se určuje, jestli článek obsahuje recenzi nebo ne. Řetězce použité pro identifikaci recenzí:

```
reviewWordsListPart = ['recenze','hodnoceni','test']
```

Pro vyhledání vhodných slov pro rozlišení recenze od ostatních článků jsem prošel recenze z datové sady a všechny články rozdělil na slova a vytvořil slovník jejich četnosti. Abych se vyhnul slovům, která se vyskytují v článcích, které recenze neobsahují jsem prošel i několik článků bez recenzí a sledoval, která slova se často vyskytují ve všech článcích.

Tato funkce vyžaduje parametr `link`, který obsahuje odkaz na zpracovávaný datový soubor, ve kterém se mají recenze vyhledávat. Jednotlivé články tohoto souboru jsou rozděleny podle řetězce **<doc**, který se nachází vždycky na začátku nového článku.

Cyklus postupně prochází všechny články ve zpracovávaném souboru, z prvního řádku článku určí URL adresu zpracovávané stránky. Z URL adresy si skript zjistí několik informací, proběhne kontrola, jestli se jedná o českou webovou stránku s doménou *.cz*, protože zdrojové soubory obsahují i stránky slovenské se kterými práce nepracuje. Dále je z URL zjištěno, jestli se nejedná pouze o odkaz na nový komentář, toto je zjištěno z konce url adresy, která v případě, že se jedná o komentář, končí řetězcem a identifikátorem komentáře, například: *#comment-360411*, práce s odkazy na komentáře nepracuje, aby nedošlo k duplicitnímu zpracování stejného článku. Na konci zpracování prvního řádku proběhne ještě kontrola, jestli se nejedná o vyhledávání speciálního produktu, a pokud ano tak, jestli je URL adresa v seznamu adres pro právě zpracovávaný produkt.

Po přečtení prvního řádku proběhne zpracování zbytku článku. Skript prochází soubor řádek po řádku a kontroluje jednotlivé slova a značky. V průběhu si skript ukládá informace o aktuálně zpracovávaném odstavci a až narazí na značku `<\p>`, která udává konec odstavce je celý odstavec zpracován podle značek nalezených při jeho zpracování. Značky, které skript vyhledává při zpracování odstavce:

- `<head>` značka indikující, že se jedná o hlavičku dokumentu, nastaví proměnnou `ArticleHeadFlag` na hodnotu `True`.
- `<s>` značení konce věty. Když skript narazí na tuto značku, zvedne počítadlo počtu vět `ParagraphTagsS` ve zpracovávaném odstavci.
- `<h>`: značka indikující, že se jedná o nadpis či podnadpis v textu, nastaví hodnotu proměnné `ParHeading` na `True`.
- `<\p>`: značka konce odstavce. Jakmile skript narazí na tuto značku začne jeho zpracování.

Než je odstavec zpracován je u něj provedena kontrola, jestli se jedná o relevantní text, toto je provedeno následující podmínkou.

```
if(ParagraphTagsS>2 or ArticleHeadFlag or ParHeading):
```

Odstavec je tedy zpracován, pokud obsahuje více než 2 věty nebo se jedná o hlavičku článku či podnadpis. V případě, že je text vyhodnocen jako relevantní jsou v něm vyhledána klíčová slova a podle toho ohodnocen článek.

Zpracování odstavce

Poté, co je odstavec vyhodnocen jako vhodný ke zpracování jsou nastaveny váhy výskytů slov určitého typu. Například pokud je v textu slovo ze seznamu názvů produktu, je přičtena výsledku článku hodnota určena váhou slov z právě tohoto seznamu.

Tato váha je určená podle toho, jestli se jedná o titulek článku či nikoli. V případě, že se jedná o titulek jsou nastaveny tyto hodnoty na:

- `BrandWeight`=10 - hodnota určující váhu slov ze seznamu firem či výrobců produktu.
- `ProductWeight`=20 - hodnotu určující váhu slov ze seznamu názvů.
- `ReviewWeight`=30 - hodnota určující váhu slov ze seznamu indikující recenze.

V opačném případné jsou tyto hodnoty nastaveny na

- `BrandWeight`=0.5
- `ProductWeight`=10
- `ReviewWeight`=0

Hodnota `ReviewWeight` je nastavená na 0 při vyhledávání v textu článku, protože často bývají v článku odkazy na recenze podobných produktů, tudíž by článek mohl být vyhodnocen jako recenze pouze na základě toho, že se někde v článku takový odkaz nachází. Hodnota `BrandWeight` je nastavena na 0.5, aby nebyly detekovány články, ve kterých je sice zmíněn výrobce hledaného produktu, ale článek se tímto produktem nezabývá, většina hledání tedy proběhne již v hlavičce článku a zbytek textu slouží pouze ke zjištění, jestli se jedná o článek zabývající se hledaným produktem.

Každý zpracovávaný řádek je rozdělen podle znaku `\t`, čímž se z řádku vytvoří seznam informací o slově, který je vložen do proměnné `lineList`. U každého slova je provedena kontrola, jestli se jedná o podstatné jméno, toto je zjištěno z hodnoty uložené na pozici `lineList[2]`, která obsahuje morfologické informace o zpracovávaném slově. Tyto informace jsou prezentovány formou řetězce, o podstatné jméno se jedná, pokud řetězec obsahuje podřetězec *NN*. V případě, že se jedná o podstatné jméno, je základní tvar zpracovávaného slova, který je uložen na pozici `lineList[2]` porovnán se seznamy obsahující hledaná slova. Pro přesnější porovnání je základní tvar slova převeden na malá písmena při vyhledávání názvu produktu a recenze.

Označení recenze

Článek je označen za recenzi, pokud obsahuje alespoň jeden výskyt názvu produktu, jedno slovo ze seznamu recenzí a jeho výsledná hodnota je vyšší či rovna hodnotě 30. Tuto hodnotu jsem zvolil tak, aby u každého produktu, který bude uživatel hledat, byl vyhledán alespoň nějaký článek. Maximální výsledná hodnota článků se výrazně liší podle produkt, který uživatel vyhledává.

Poté, co je článek vyhodnocen proběhne určení názvu produktu a výsledek je přidán do slovníku `resultDict`. Článek je vložen do slovníku pod klíčem jeho URL adresy a jako hodnoty jsou do slovníku vloženy:

- `WordResult` - Výsledné skóre článku
- `IsBrand` - počet výskytů názvu výrobce v textu článku
- `IsProduct` - počet výskytů názvu produktu v textu článku
- `IsReview` - počet výskytů slov recenzí v textu článku
- `ArticleProduct` - název produktu kterým se článek zabývá
- `urlName` - název portálu na kterém se článek nachází

Poté, co jsou zpracovány všechny články v souboru vrátí funkce slovník `resultDict`.

4.4 Určování názvu produktu

Určování názvu produktu probíhá jako součást funkce pro vyhledávání recenzí, název je určen až poté, co je článek označen za recenzi. Pokud se nejedná o speciální produkt, je název článku určen na základě velkých písmen v titulku článku a url adresy.

Skript rozdělí titulek článku na slova podle bílých znaků a vyhledá slova začínající velkým písmenem a číslovky, tyto slova a číslovky jsou poté porovnány s URL adresou stránky. Pokud se vybraná slova v této adrese nachází, jsou vloženy do proměnné `ArticleProduct` reprezentující název produktu.

Po vyhledání názvu skript `productFinder` provede ověření, jestli se jedná o platný název produktu. Ověření probíhá ve funkci `WebScrape_product(product)`. Vstupní parametr *product*, který obsahuje řetězec s vyhledaným názvem produktu je převeden do formátu vhodného pro vyhledávání převedením mezer na znak *+*, poté proběhne hledání na webu *heureka.cz*. Toto hledání je provedeno pomocí webscrapingu url adresy vytvořené spojením *https://www.heureka.cz/?h%5Bfraze%5D=* a vyhledávaného řetězce. Poté, co je stažen HTML kód z portálu *heureka.cz* jsou prohledány titulky položek. Většinou pokud je zadán neplatný produkt tak vyhledávání nenajde žádnou položku na základě zadaného řetězce a produkt je tudíž označen za neplatný. V případě, že hledání vrátí několik položek jsou porovnány titulky prvních dvou položek s vyhledávaným řetězcem a na základě podobnosti je produkt vyhodnocen jako platný či neplatný. Porovnání řetězců probíhá pomocí třídy *SequenceMatcher* z Pythonovského modulu *difflib*. Tato třída porovnává řetězce podle nejdelší souvislé shodné posloupnosti a poté pomocí metody *ratio()* vrací hodnotu v intervalu 0.0-1.0 kde vyšší číslo reprezentuje vyšší podobnost. Název je označen jako platný, pokud dosahuje podobnosti alespoň 0.3.

Speciální produkty

Pro produkty, jejichž název nejde určit na základě velkých písmen, je třeba použít jiný způsob určení názvu. Název těchto produktů je vyhledán pomocí webscrapingu zpracovávaného článku. Protože je každý portál jiný a používá vlastní způsob zápisu HTML nelze vytvořit skript, který by vyhledal název na všech webech, je proto, tedy potřeba používat specifický kód pro každý portál. To znamená, že tyto produkty budou vyhledávány pouze na portálech, které mají určený kód specifikován. Speciální produkty, weby a kód pro určení názvu produktu jsou definovány v konfiguračním souboru.

U speciálních produktů je nejprve stažen obsah ve formě HTML a poté pomocí knihovny `BeautifulSoup` je obsah zpracován a vložen do proměnné `page_soup`. V konfiguračním souboru je podle URL aktuálně zpracovávaného článku vyhledán kód, který z proměnné `page_soup` vyhledá název produktu a vloží jej do proměnné `ArticleProduct`, se kterou skript dále pracuje stejně jako u ostatních produktů.

4.5 Paralelní zpracování

Paralelní zpracování umožňuje spustit skript jako více procesů. Paralelizace je v této práci využita pro zpracování několika souborů najednou funkcí `readReviews(link)`. Soubory určené ke zpracování jsou uloženy v seznamu `CorrectDateFilesList`, ze kterého jednotlivé procesy vybírají soubory které ještě nebyly zpracovány, dokud není zpracován celý seznam. Po dokončení práce na souboru jsou výstupní hodnoty funkce `readReviews(link)` přidány

do seznamu `resList`. Poté, co jsou zpracovány všechny soubory jsou položky seznamu `resList` převedeny na slovník `resultDict` pro další zpracování.

Počet procesů

Počet, kolik procesů se má spustit najednou je specifikován jako vstupní argument při spuštění skriptu. Paralelní zpracování výrazně sníží délku zpracování všech souborů, ale je taky náročnější na paměť počítače, na osobním počítači který má 16 GB paměti, jsem mohl práci spustit maximálně na dvou procesech.

Způsob paralelního zpracování

Paralelní spuštění je implementováno tak, že je zpracováváno více souborů najednou, to se při testování ukázalo jako nejvlivnější na zrychlení zpracování. Tento způsob implementace při spuštění práce na zpracování pouze jednoho souboru nemá žádný vliv na délku zpracování.

Testoval jsem i zpracování jednoho souboru více procesy, ale to na snížení délky zpracování nemělo moc vliv.

4.6 Určování sentimentu

Určením sentimentu se zabývá skript `SentimentAnalyzerAspect.py`, který je volán z hlavního skriptu funkcí `Sentiment_from_url(url)`.Tato funkce vrátí řetězec *pos*,*neg* či *??* na základě ohodnocení sentimentu textu článku. Nejprve se skript pokusí vyhodnotit článek na základě aspektové analýzy, v případě že se nepovede pomocí tohoto způsobu určit hodnocení článku, je využita analýza sentimentu pomocí strojového učení.

Aspektová analýza

Nejprve je stažen obsah článku z URL adresy zadané jako vstupní parametr při spuštění funkce, obsah toho článku je poté zpracován nástrojem *justext* pro vyhledání relevantního textu. Text vybraný tímto nástrojem je prohledáván po odstavcích a každý odstavec je zpracován skriptem `udpipeParse.py`, který v zadaném textu vyhledá aspekty a jejich hodnocení. Aspekty ze zpracovávaného odstavce jsou poté ohodnoceny na základě lexikonů pozitivních či negativních slov. Po porovnání počtu pozitivně a negativně hodnocených aspektů je článku přiřazen sentiment podle vyšší hodnoty, pokud se v článku nepodaří určit sentiment na základě aspektů je volána funkce `Sentiment_from_Articletext(ArticleText)`, která z již staženého článku provede analýzu sentimentu pomocí strojového učení.

Vyhledání aspektů a jejich hodnocení

Tato část práce probíhá ve skriptu `udpipeParse.py`, k tomuto skriptu je přistoupeno pomocí funkce `AspectAnalyse(txt)`. Tato funkce vyhledá ve vstupním textu aspekty a jaká slova je hodnotí, jako vstupní parametr je očekávána věta či odstavec textu. Pro zpracování textu jsem využil nástroj *udpipe* v kombinaci s modelem `UD Czech PDT`. Nejprve je zadaný odstavec zpracován tímto nástrojem, který text rozloží na jednotlivé věty, určí základní tvary slov, jejich slovní druhy slov a jakou závislost na sobě ve větě mají[5]. Tvorba tohoto skriptu byla inspirována článkem od Rohana Goela[13].

Vstupní text je dále zpracováván po jednotlivých větách, do skriptu je vložen ze souboru `stop_words_czech.txt` lexikon stopslov, při zpracování věty jsou v základních tvarech slov vyhledávány stopslova a ta jsou z věty odstraněna. Je tak vytvořena nová věta, která stopslova neobsahuje. Nová věta je opět zpracována nástrojem *udpipe*, aby se vytvořili nové větné závislosti.

Na základě závislostí jsou vytvořeny dvojice slov, které na sobě závisí a jejich vztah je popsán pojmem definovaným ve využitém modelu.

Ve zpracovávané větě jsou vyhledány slova označené jako podstatné či přídavné jména, na základě těchto slov je vytvořen seznam vlastností zpracovávané věty a průběžně je tvořen seznam vlastností celého odstavce.

Poté je seznam vlastností porovnán se seznamem dvojic, jsou vyhledávány položky, které se nacházejí v seznamu dvojic a zároveň v seznamu vlastností, pokud je taková položka nalezena, je nahlédnuto na vztah dvojic, pokud je vztah dvou slov obsažen v následujícím seznamu, obsahuje dvojice hodnocení některé z vlastností.

["nsubj", "acl:relcl", "obj", "advmod", "amod", "neg", "xcomp", "compound", "orphan", "acl", "conj", "appos"]

Tento seznam vztahů byl vytvořen podle článku Subhabrata Mukherjee a Pushpaka Bhattacharyya zaměřeného na aspektovou analýzu recenzí produktů[33], popis jednotlivých zkratek je v dokumentaci použitého modelu[2].

Po prohledání všech vět ve zpracovávaném odstavci jsou vyhledány ve vlastnostech podstatná jména a ty označeny jako aspekty, na výstup funkce je poté vložen seznam Aspektů a ke každému aspektu přidán seznam slov, které tento aspekt hodnotí.

Ohodnocení na základě aspektů

Výsledek hodnocení článku je určen proměnnou `rating`. Aspekty a jejich hodnocení jsou vyhledávány zvlášť v každém odstavci zpracovávaného článku, po vyhledání aspektů je nahlédnuto na všechna slova hodnotící tyto aspekty. Pokud je hodnotící slovo nalezeno v lexikonu pozitivních slov, je inkrementována hodnota proměnné `rating`, pokud je toto slovo nalezeno v negativním lexikonu, je tato hodnota dekrementována. Po prohledání všech odstavců je pomocí proměnné `rating` určen sentiment článku, pokud je její hodnota pozitivní je článku přiřazen pozitivní sentiment, v opačném případě je přiřazen negativní sentiment, pokud je hodnota této proměnné rovna 0, je tento článek zpracován pomocí strojového učení.

Analýza sentimentu pomocí strojového učení

Pro hodnocení sentimentu skript používá knihovny **PyTorch**, **TensorFlow 2.0** a **Transformers** pro vytvoření a použití modelu pro vyhodnocení sentimentu pomocí strojového učení.

Tento skript byl vytvořen podle článku od *Venelina Valkova* zaměřující se na analýzu sentimentu recenzí aplikací na webu *play.google.com*[3][36].

Nejlepší před trénovaný model, který jsem pro zpracování českého textu našel, byl

DeepPavlov/bert-base-bg-cs-pl-ru-cased

[2] https://universaldependencies.org/treebanks/cs_pdt/index.html#ud-czech-pdt
[3] https://play.google.com/store

Tento model má problémy s diakritikou, ale po odstranění diakritiky funguje docela dobře.

Na následujícím příkladu lze vidět průběh převedení české věty *Auto je podle nás v redakci velice povedené vozidlo* do tvaru vhodného pro určení sentimentu pomocí před trénovaného modelu.

```
Sentence: Auto je podle nas v~redakci velice povedene vozidlo.
  Tokens: ['Auto', 'je', 'podle', 'nas', 'v', 'redakci', 'velice', 'povede', '##ne', 'vozidlo','.']
Token IDs: [23265, 2053, 3874, 12947, 190, 88651, 16210, 101961, 2136, 60058, 119]
```

Některé slova byly rozděleny na dva tokeny, druhý token začíná značkou `##` která naznačuje, že je součásti tokenu předchozího. Rozdělení nastane v případě, že model, který větu zpracovává určité slovo nerozpozná a rozdělí ho na slova, které poznává.

Funkce `Sentiment_from_url(url)` očekává jako vstupní parametr url adresu článku, u kterého má být ohodnocen sentiment. Nejprve je stažen HTML kód zadané stránky, který je následně zpracován nástrojem *justext* s parametrem pro vyhledávání českého textu. A text je poté postupně po odstavcích třídou `SentimentClassifier` zpracováván a každý odstavec či podnadpis je pomocí vytrénovaného modelu ohodnocen. Výsledky ohodnocení se sčítají, poté jsou zprůměrovány a na základě průměru je vyhodnocen sentiment článku. Návratové řetězce funkce:

- 'pos' - Sentiment článku byl vyhodnocen jako pozitivní
- 'neg' - Sentiment článku byl vyhodnocen jako negativní
- '??' - Sentiment článku se nepodařilo určit, tato situace nastane většinou v případě že se obsah stránky nepodaří stáhnout nebo nástroj *justext* nenajde na stránce žádný text vhodný ke zpracování.

Trénování modelu

Trénování modelu probíhalo ve skriptu `training.py`. Tento skript obsahuje několik tříd a funkcí popsaných níže.

Tvorba Datasetu

Nejprve jsem vytvořil dataset pro trénování. Dataset se skládá z článků označených jako pozitivní či negativní recenze, tyto články jsou poté rozděleny na odstavce a z nich vytvořen soubor `trainingReviews.csv`, který je využit pro trénování modelu. Tento soubor obsahuje na každém řádku dvě hodnoty, text a sentiment a při jeho tvorbě je z článků odstraněna diakritika.

Dataset se skládá celkem z 172 odstavců s přiřazeným sentimentem, z toho je 99 označeno jako pozitivní a 73 jako negativní. Pro trénování jsou tyto odstavce rozřazeny do tří kategorií:

- Trénovací data - Data, na kterých se model učí.
- Validační data - Data použitá k ověření přesnosti modelu při jeho učení.
- Testovací data - Data použitá pro otestování modelu po dokončení učení.

Dataset byl rozdělen na 155 testovacích odstavců, 8 validačních a 9 testovacích.

Průběh trénování

Model Bert vyžaduje, aby počet tokenů ve všech zpracovávaných textech byl stejný, je proto zvolena konstanta `MAX_LEN`, která udává počet, na který se bude počet tokenů zarovnávat, pokud je text delší, k vyhodnocení sentimentu je použito pouze prvních X tokenů kde X je hodnota `MAX_LEN` a zbytek textu odstřižen, v opačném případě kde je počet tokenů nižší než maximum je zpracovávaný text doplněn prázdnými tokeny s hodnotou 0, aby byla splněna požadovaná délka. Po projití odstavců v datasetu jsem zvolil hodnotu `MAX_LEN`=200.

Model BERT používá několik speciálních tokenů, které je potřeba do textu doplnit, aby šlo text zpracovat, dále je potřeba text upravit, aby šel zpracovat pomocí knihovny `PyTorch`. O přidávání těchto tokenů a zpracování textu se stará tokenizer, který metodou `encode_plus` tokeny automaticky doplní a text zpracuje. Tato metoda je pro správné zakódování spouštěna s následujícími parametry:

```
encoding = tokenizer.encode_plus(
  review, − zpracovavany text
  add_special_tokens=True, − pridani specialnich tokenu
  max_length=MAX_LEN, − nastaveni maximalni delky textu
  return_token_type_ids=False, − vypne vraceni identifikatoru typu tokenu
  padding='max_length', − doplneni kratsiho textu prazdnymi tokeny na delku MAX_LEN
  truncation=True, − odstrizeni prebytecneho textu pokud je delsi nez hodnota MAX_LEN
  return_attention_mask=True, − vraceni masky oznacujici ktere tokeny nejsou prazdne
  return_tensors='pt', − konverze vysledku do formy vhodne pro knihovnu PyTorch
)
```

Toto zpracování probíhá v třídě `ArticleReviewDataset(dataset)`. Výstupem této třídy je slovník obsahující seznam tokenů zpracovaného textu, maska pozornosti s informací, které tokeny, jsou nulové a které ne a nakonec tenzory knihovny PyTorch.

Protože by zpracování celého datasetu najednou bylo náročné na paměť počítače, je dataset rozdělen na několik menších dávek, o to se stará funkce

```
create_data_loader(df,tokenizer,max_len,batch_size)
```

Tato funkce podle parametru `batch_size` datovou sadu rozdělí na menší dávky.

Pro trénování modelu je použito několik dalších funkcí a tříd, které stručně popsány níže:

- `train_epoch` - funkce pro trénování modelu, projde trénovací data jednoho epochu a vrací počet správných výsledku a chybovost modelu.
- `eval_model` - ohodnocení přesnosti aktuálního stavu trénovaného modelu.

Více informací k těmto třídám a funkcím je dostupné v článku zabývajícím se tvorbou toho skriptu[36].

Kapitola 5

Vyhodnocení

V této kapitole jsou rozvedeny výsledky jednotlivých částí, které při práci probíhají a jak byla práce testována. Dále popisuje problémy, které nastávaly při vývoji a nastávají v aktuální verzi systému.

5.1 Délka zpracování zdrojových dat

V prvních verzích práce bylo součástí systému i určování slovních druhů u zdrojových dat pomocí nástroje *MorphoDita*, to mělo vliv na délku zpracování článků z jednoho dne. Vyhledání článků z jednoho dne tak trvalo průměrně asi 8 minut. Aktuální verze pracuje již se zpracovanými vstupními soubory, které mají určené slovní druhy a vyhledávání recenzí je tímto výrazně zrychleno, vyhledání článků jednoho souboru tak trvá průměrně 30 sec.

Možnost paralelního spuštění dále zrychluje průběh celé práce. Při spuštění práce na vyhledání recenzí mezi daty 01.01.2021 - 19.04.2021 a paralelně spuštěno jako 4 procesy na serveru **minerva3**. Celkem je zpracováno 94 souborů, zpracování celkem trvá 45 minut a 40 sekund, z toho je 21 minut a 29 sekund stráveno prohledáváním souborů a 24 minut a 11 sekund ohodnocováním sentimentu vyhledaných článků. Kdyby bylo třeba vstupní data před vyhledáváním zpracovat trvalo by vyhledání recenzí v těchto dnech 12 hodin a 32 minut.

Celkově tak využitím předem zpracovaných souborů došlo ke snížení časové náročnosti práce o **93.9%**.

Produkt	Délka zpracování	Hledání článků	Určení sentimentu	Nalezených článků
telefon	00:45:40	00:21:29	0:24:11	638
film	00:10:36	00:06:33	00:04:03	222
auto	00:55:56	00:30:54	00:25:02	632
kniha	00:08:13	00:05:14	00:02:59	94
sluchátka	00:20:29	00:09:55	00:10:34	184

Tabulka 5.1: Délka trvání vyhledání a zpracování článků pro různé produkty ve dnech 01.01.2021 - 19.04.2021 paralelně spuštěno na 4 procesech, server minerva3

Délka vyhledání pro knihy a filmy je nižší, protože se jedná o speciální produkty, které mají předem určené portály, na kterých se mají recenze vyhledávat, není tedy potřeba prohledávat všechny články, jako je tomu u ostatních produktů.

5.2 Rozpoznání recenze

Na výstup jsou vypsány články, které se i nejméně podobají recenzi a seřazeny podle skóre přiřazeného při vyhledávání recenze. Články jsou na výstupu seřazeny sestupně, takže články s nejvyšším skóre jsou na nejnižším řádku. Tento způsob jsem zvolil, abych se vyhnul případům kde práce žádný výsledek, nevrátí pokud uživatel vyhledává produkt, který moc článků nerecenzuje. To ovšem i znamená, že pokud uživatel vyhledává produkt, který je recenzován často bude vypsáno mnoho článků a kvůli ohodnocování sentimentu bude dokončení práce déle trvat.

Při zpracování souborů mezi dny 01.01.2021-19.04.2021 které celkem obsahují 94 dnů, je při vyhledávání článků recenzující telefony nalezeno 638 článků. Protože jsou výsledky seřazeny podle relevance, přesnost správně vyhledaných článků postupně klesá podle toho z kolika článků přesnost počítáme. Na tomto vzorku je posledních 30 článků označeno se 100% přesností, 100 článků je označeno s přesností 79% a začínají se vyskytovat články týkající se telefonů, ale nejedná se o recenze, jde například o články píšící o nově představených telefonech, tipech na prodloužení životnost baterie telefonu.

Velikost vzorku	Správně určených článků	Přesnost
30	30	100%
100	79	79%
200	127	63,5%
638	302	47,3%

Tabulka 5.2: Ukázka přesnosti rozpoznávání recenzí pro telefony podle počtu vzorku

Pro zvýšení přesnosti celkového vyhledání a zrychlení práce by bylo možné vzít například jen 30% nejlepších výsledků a zbytek vyhledaných článků ignorovat. Při využití tohoto způsobu by ovšem mohlo dojít ke ztracení několika správně označených recenzí.

Hledaný produkt	Přesnost	Celkem vyhledaných článků
telefon	79%	638
sluchátka	54%	184
film	67%	222
kniha	44%	94
auto	85%	632

Tabulka 5.3: Ukázka přesnosti rozpoznávání recenzí různé produkty na posledních 100 článků při zpracování dnů 01.01.2021-19.04.2021

Testování vyhledávání recenzí

Při tvorbě vyhledávače recenzí jsem výsledky porovnával s datovou sadou vytvořenou manuálním vyhledáváním. Skript jsem testoval hlavně na produktech *telefon* a *auto*, pro které jsem v této sadě našel nejvíce článků. Aktuální verze práce najde všechny články v datové sadě, pro některé produkty najde i články, které jsem při manuálním vyhledávání přehlédl.

5.3 Vyhledání relevantního textu

Vyhledávání relevantního textu pomocí počtu vět většinou správně vyhledá obsah článku, někdy je jako relevantní označen i text na konci článku obsahující informace o portálu jako provozovatel serveru, adresu sídla provozovatele či informace o vydavatelství, tento text ovšem většinou neobsahuje žádná klíčová slova, které by ovlivnili výsledek hledání.

Lepší způsob implementace vyhledání relevantního textu by bylo propojení s nástrojem *justext*, který spolehlivě vyhledává relevantní obsah článků. Tento nástroj ovšem k práci potřebuje zdrojový neupravený HTML kód stránky, který by bylo potřeba získat, buď stažením celé stránky pomocí adresy URL, nebo stažené HTML najít v nezpracovaných souborech projektu *FeedsCrawler*. Nezpracované soubory jsou ovšem uloženy v jiném pořadí než data, ve kterých jsou vyhledávány články. Navíc by po určení relevantního textu tímto nástroj bylo potřeba určit slovní druhy u jednotlivých slov a zpracování jednoho článku by tak mohlo být časově velice náročné.

5.4 Rozpoznání názvu produktu

Rozpoznávání názvů produktů na základě velkých písmen funguje poměrně spolehlivě, vyhledávaný produkt ovšem musí být vypsán v titulku i v url adrese článku, což může vést k případům, kde článek obsahující recenzi hledaného produktu není vypsán kvůli špatnému titulku či nevhodné url adrese. U speciálních produktů vyhledávání názvů pomocí HTML kódu funguje jako další kontrola, jestli se jedná o článek zabývající se daným produktem protože portály většinou drží stejný formát pro jeden typ produktu, takže pokud je například při hledání filmů vyhledán článek obsahující recenzi filmové kamery, není nakonec vypsán, protože články recenzující tento typ produktu používají jiný formát než u filmů.

U speciálních produktů je problém, že je třeba nejdříve definovat weby, na kterých se mají hledat tyto produkty hledat a napsat kód pro vyhledávání názvu. V Aktuální verzi jsou tyto podmínky definovány pro produkty *film*, *kniha* a *videohra*.

Na následujících příkladech jsou ukázány výsledky hledání názvu produktu z titulku článku:

Název Produktu	Titulek článku
Sony Xperia 1 II	Neortodoxní špička pro vybrané zákazníky. Test Sony Xperia 1 II
Motorola Moto G 5 G Plus	RECENZE Motorola Moto G 5G Plus Střední třída se vším všudy
Microsoft Surface Duo	Microsoft Surface Duo v prvních recenzích: není důvod k oslavám
Peugeot 3008 GT Hybrid 4 2020	Test hybridu Peugeot 3008 GT Hybrid4 (2020)

Tabulka 5.4: Výsledky hledání názvu produktu z titulku článku

5.5 Použití wikidat

Vyhledávání alternativních názvů, výrobců a samotného produktu je omezeno znalostmi wikidat, to znamená že může nastat případ, při kterém bude systém vyhledávat produkt jiný než uživatel zamýšlel. Například pokud uživatele zajímají články recenzující počítačové

hry a zadá produkt *hra* není na wikidatech nalezena položka *videohra* která uživatele zajímá ale je nalezena položka *hra*, která vrátí výstup, který uživatele nezajímá.

S vyhledáním správné položky ve wikidatech může nastat problém i v případě, že je zadán produkt v množném čísle, například pokud uživatele zajímají recenze automobilů a zadá produkt *auta* není ve wikidatech nalezena položka *automobil* kterou by uživatel očekával, ale je nalezena položka *Auta* označující řeku v Bělorusku.

Problém se znalostmi wikidat nastává ještě u vyhledání výrobců produktů, například u výrobců telefonu není nalezena firma *Apple*, která v produktech které vyrábí, nemá uvedenou přímo položku *telefon*, ale má uvedeny konkrétní modely telefonů.

5.6 Určení sentimentu

V článku jsou vyhledány všechny aspekty zmíněny v článku, to znamená, že jsou vyhledány i aspekty, které se úplně netýkají hodnoceného produktu, to může ovlivnit hodnocení článku. Další problém který při určování nastává je se základními tvary slov, v některých případech základní tvar změní sentiment, který je pro dané slovo určen, s tímto problémem jsem se setkal například u slova *neortodoxní*, kterému po zpracování byl přiřazen základní tvar *ortodoxní*, což zapříčinilo změně sentimentu, který byl článku přiřazen.

Výsledky určování sentimentu pomocí strojového učení záleží na produktu, kterým se článek zabývá, to je způsobeno malým rozsahem datové sady. Při tvorbě datové sady nastal často problém s manuálním přiřazením sentimentu produktu, datová sada tedy byla tvořena články, které byly číselně ohodnoceny a podle této hodnoty bylo možno určit, jestli článek hodnotí produkt pozitivně či negativně. U některých produktů, které v datové sadě nejsou moc prezentovány, záleží na sentimentu článků, které byly k danému produktu vyhledány, například pro produkt *film* většina nalezených článků hodnotila produkt negativně, proto při vyhledání článků recenzující filmy, pro které je třeba určit sentiment pomocí strojového učení je většina označena jako negativní.

Název Produktu	Přesnost
telefon	85.7%
film	46.15%
auto	89.1%
kniha	61.40%
sluchátka	80.65%

Tabulka 5.5: Přesnost určení sentimentu u různých produktů

Přesnost určení sentimentu jsem testoval na článcích, které obsahují číselné ohodnocení produktu, takže se podle této hodnoty dá zjistit, jestli autor hodnotí produkt pozitivně či negativně. Určení sentimentu u technických produktů je přesnější z toho důvodu, že tyto produkty často mívají vysoké hodnocení, například průměrné hodnocení automobilů v článcích, ze kterých jsem přesnost určoval bylo 86%. Hodnocení filmů je zase často blíže hodnotě 50% takže se z této hodnoty těžko určuje, jestli je produkt hodnocen pozitivně či negativně.

5.7 Výstup

Po dokončení práce jsou na standardní výstup vypsány články označené jako recenze sestupně seřazené podle ohodnocení článku. Většinou je pár posledních článků správně označeno za recenze a pak se začnou mezi výsledky vyskytovat i chybně označené články, ve kterých je zmíněn hledaný produkt, ale nejedná se o recenze. Přesné výsledky k těmto hodnotám jsou v tabulce 5.2. Chybně vyhledané články se mohou objevit z několika důvodů, například při hledání produktu *telefon* jsou někdy vyhledány články o testech na COVID-19, protože slovo *test* je jeden z řetězců, podle kterého se určuje, jestli se jedná o recenzi a dále je v textu zmíněno třeba telefonní číslo na doktora. Často jsou ale vyhledány i články, které s produktem souvisí, ale nejedná se přímo o recenzi, například představení nového modelu.

Výstup je vypsán na standardní výstup ve formátu:
[název produktu] - [sentiment] - [url adresa článku]

Kapitola 6

Závěr

Práce má za úkol vyhledat články obsahující recenze zadaného produktu na českých webech, rozpoznat o jakém produktu se v textu píše a určit sentiment obsahu článku. Tento úkol práce splňuje.

I přesto, že systém správně vyhledá články obsahující recenze, ve výsledcích se vyskytují i špatně zvolené články, je tedy třeba v některých případech ručně vybírat, které články jsou recenze a které ne. Výsledky jsou seřazeny podle relevantnosti.

Vyhledání názvu produktu funguje u článků, které mají daný produkt uveden v titulku i url adrese článku, i když to tak články většinou mají, může nastat případ, kde článek obsahující recenzi produktu není přidán na výpis právě kvůli absenci názvu v titulku či url adrese.

Pro určování názvů některých produktů je potřeba nejprve vytvořit unikátní kód pro jejich vyhledání pomocí HTML a definovat portály, na kterých má být tento produkt hledán.

Aktuálně je práce nahraná pouze na serverech společnosti KNoT, kde má přístup ke zdrojovým datům, ve kterých jsou vyhledávány články. To znamená, že systém mohou použít pouze uživatelé, kteří mají k tomuto serveru přístup, v budoucnu by mohlo být vytvořeno webové rozhraní aby práci mohli využít i uživatelé, kteří k tomuto serveru přístup nemají.

Sentiment je určován s průměrnou přesností 72,6%. Články jsou rozlišeny pouze na pozitivní či negativní a není jim přiřazen žádný stupeň hodnocení, takže tento údaj je pro uživatele spíše orientační.

Práci mohou využít pracovníci portálů zaměřující se na recenze jako například *heureka.cz* na vyhledávání článků obsahující recenze a pak tyto články doplnit k produktům na portálu, pro který pracují.

Využití uživatelem, který si chce zakoupit produkt a hledá recenze produktu, který ho zajímá by délka zpracování možná mohla odradit od použití práce. Tento problém by se dalo vyřešit propojením práce s projektem *FeedsCrawler* a recenze vyhledávat hned po stažení článků a vytvořit tak soubory pro často hledané produkty, tímto způsobem by uživatel zajímající se o recenze produktů měl výsledky hned, pokud by vyhledával recenze často hledaných produktů.

Literatura

[1] Bast, H., Björn, B. a Haussmann, E. Semantic search on text and knowledge bases. *Foundations and Trends in Information Retrieval*. 1. vyd. Now Publishers Inc. Hanover, MA, USA. 2016, sv. 10, 2-3, s. 119–271. ISSN 1554-0669, 1554-0677.

[2] Bílý, D. *Rozpoznávání postojů z filmových recenzí* [online]. Brno, CZ, 2020. [cit. 2021-04-11]. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/22452/.

[3] Brownlee, J. *Difference Between Algorithm and Model in Machine Learning* [online]. Srpen 2020 [cit. 2021-05-1]. Dostupné z: https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning/.

[4] Cambria, E., Das, D., Bandyopadhyay, S. a Feraco, A. *A practical guide to sentiment analysis*. 1. vyd. New York, NY: Springer, 2017. ISBN 9783319553924.

[5] Chockalingam, N. *Simple and Effective Feature Based Sentiment Analysis on Product Reviews using Domain Specific Sentiment Scores*. Leden 2018 [cit. 2021-05-1].

[6] Cihlářová, D. *Zpracování uživatelských recenzí* [online]. Brno, CZ, 2019. [cit. 2021-04-28]. Master's thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: https://www.fit.vut.cz/study/thesis/22145/.

[7] Cimprich, P. Atom 1.0: formát. *Root.cz* [online]. 1. vyd. Internet Info, s.r.o. Leden 2006, č. 1, [cit. 2021-04-11]. Dostupné z: https://www.root.cz/clanky/atom-1-0-format/.

[8] Cyganiak, R. A relational algebra for SPARQL. *Digital Media Systems Laboratory HP Laboratories Bristol. HPL-2005-170*. 1. vyd. 2005, sv. 35, č. 1, s. 9.

[9] Devlin, J., Chang, M.-W., Lee, K. a Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2. vyd. Google AI, Říjen 2018.

[10] Ghati, G. Comparison between BERT, GPT-2 and ELMo. *Medium* [online]. 1. vyd. Medium. May 2020, č. 1, [cit. 2021-04-22]. Dostupné z: https://medium.com/@gauravghati/comparison-between-bert-gpt-2-and-elmo-9ad140cd1cda.

[11] Glotzbach, R., Mohler, J. a Radwan, J. E. RSS as a course information delivery method. In: Association for Computing Machinery. *SIGGRAPH '07*. San Diego, California: ACM Press, Srpen 2007, s. 16. ISBN 9781450318303.

[12] Godbole, N., Srinivasaiah, M. a Skiena, S. Large-Scale Sentiment Analysis for News and Blogs. *Icwsm*. 1. vyd. New York NY, USA: [b.n.]. 2007, sv. 7, č. 21, s. 219–222.

[13] Goel, R. Implementing Aspect Based Sentiment Analysis using Python. *Medium* [online]. 1. vyd. Analytics Vidhya. Prosinec 2020, č. 1, [cit. 2021-04-30]. Dostupné z: https://medium.com/analytics-vidhya/aspect-based-sentiment-analysis-a-practical-approach-8f51029bbc4a.

[14] Goel, V. *Applying Machine Learning to classify an unsupervised text document* [online]. Towards Data Science, listopad 2018 [cit. 2021-05-06]. Dostupné z: https://towardsdatascience.com/applying-machine-learning-to-classify-an-unsupervised-text-document-e7bb6265f52.

[15] Gordon, M. a Pathak, P. Finding information on the World Wide Web: the retrieval effectiveness of search engines. *Information processing & management*. 2. vyd. Elsevier. Březen 1999, sv. 35, č. 1, s. 141–180. ISSN 03064573.

[16] Hajič, J. *Disambiguation of Rich Inflection: Computational Morphology of Czech*. 1. vyd. Karolinum, 2004. ISBN 9788024602820. Dostupné z: https://books.google.cz/books?id=sB63AAAACAAJ.

[17] Horev, R. BERT Explained: State of the art language model for NLP. *Medium* [online]. 1. vyd. Towards Data Science. Listopad 2018, č. 1, [cit. 2021-04-22]. Dostupné z: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[18] Klocok, A. *Analýza recenzí výrobků* [online]. Brno, CZ, 2020. [cit. 2021-04-11]. Master's thesis. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/22451/.

[19] Kohlschütter, C., Fankhauser, P. a Nejdl, W. Boilerplate detection using shallow text features. In: Leibniz Universität Hannover. *Proceedings of the third ACM international conference on Web search and data mining*. New York, New York, USA: ACM Press, 2010, s. 441–450. ISBN 9781605588896.

[20] Krčmář, P. Vše podstatné o RSS. *Root.cz* [online]. 1. vyd. Internet Info, s.r.o. Září 2006, č. 1, [cit. 2021-04-11]. Dostupné z: https://www.root.cz/clanky/vse-podstatne-o-rss/.

[21] Laender, A. H., Ribeiro Neto, B. A., Da Silva, A. S. a Teixeira, J. S. A brief survey of web data extraction tools. *ACM Sigmod Record*. 1. vyd. ACM New York, NY, USA. Červen 2002, sv. 31, č. 2, s. 84–93. ISSN 0163-5808.

[22] Lassila, O., Swick, R. R. et al. Resource description framework (RDF) model and syntax specification. 1. vyd. Citeseer. 1998, č. 1.

[23] Mansouri, A., Affendey, L. S. a Mamat, A. Named entity recognition approaches. *International Journal of Computer Science and Network Security*. 1. vyd. Citeseer. 2008, sv. 8, č. 2, s. 339–344.

[24] Matějka, J. *Rozšíření systému pro získávání, zpracování a analýzu rozsáhlých kolekcí textů z webu* [online]. Brno, CZ, 2018. [cit. 2021-04-13]. Bachelor's thesis. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/20846/.

[25] Pavlů, J. *Analýza obsahu sociálních sítí týkající se českých mobilních operátorů* [online]. Brno, CZ, 2018. [cit. 2021-04-30]. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: https://www.fit.vut.cz/study/thesis/20346/.

[26] Pérez, J., Arenas, M. a Gutierrez, C. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)*. 3. vyd. ACM New York, NY, USA. Srpen 2009, sv. 34, č. 1, s. 1–45. ISSN 0362-5915, 1557-4644.

[27] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C. et al. *Deep contextualized word representations*. 2. vyd. Allen Institute for Artificial Intelligence,Paul G. Allen School of Computer Science & Engineering, University of Washington, Únor 2018.

[28] POMIKÁLEK, J. *Removing Boilerplate and Duplicate Content from Web Corpora* [online]. Brno, CZ, 2011. [cit. 2021-04-17]. Disertační práce. Masarykova univerzita, Fakulta informatiky, Brno. Vedoucí práce Pala, K. Dostupné z: https://is.muni.cz/th/o6om2/.

[29] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. et al. Language models are unsupervised multitask learners. *OpenAI blog*. 1. vyd. San Francisco, California, United States: [b.n.]. 2019, sv. 1, č. 8, s. 9.

[30] Rasolofo, Y. a Savoy, J. Term proximity scoring for keyword-based retrieval systems. In: Springer. *European Conference on Information Retrieval*. 2003, s. 207–218. ISBN 3540012745.

[31] Robinson, S. The Ongoing search for efficient web search algorithms. *SIAM News*. 1. vyd. 2004, sv. 37, č. 9, s. 4.

[32] Sayre, R. Atom: The standard in syndication. *IEEE Internet computing*. 4. vyd. IEEE. Červenec 2005, sv. 9, č. 4, s. 71–78. ISSN 1089-7801, 1941-0131.

[33] Subhabrata Mukherjee, P. B. *Feature Specific Sentiment Analysis for Product Reviews*. 1. vyd. Bombay: Dept. of Computer Science and Engineering, IIT, 2012.

[34] SÜSS, M. *Rozpoznání pojmenovaných entit v textu* [online]. Brno, CZ, 2019. [cit. 2021-04-25]. Diplomová práca. Mendelova univerzita v Brně, Faculty of Business and Economics. Vedoucí práce Ing. František Dařena, P. doc. Dostupné z: https://theses.cz/id/rkb3wp/.

[35] Taboada, M., Brooke, J., Tofiloski, M., Voll, K. a Stede, M. Lexicon-based methods for sentiment analysis. *Computational linguistics*. 1. vyd. MIT Press. Červen 2011, sv. 37, č. 2, s. 267–307. ISSN 0891-2017, 1530-9312.

[36] Valkov, V. Sentiment Analysis with BERT and Transformers by Hugging Face using PyTorch and Python. *Curiousily* [online]. 1. vyd. Duben 2020, č. 1, [cit.

2021-04-23]. Dostupné z: https://curiousily.com/posts/sentiment-analysis-with-bert-and-hugging-face-using-pytorch-and-python/.

[37] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need*. 5. vyd. Google Brain, Červen 2017.

[38] Vrandečić, D. a Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*. 10. vyd. ACM New York, NY, USA. Září 2014, sv. 57, č. 1, s. 78–85. ISSN 0001-0782, 1557-7317.

[39] W3C. *SPARQL 1.1 Query Language* [online]. W3C, Březen 2013 [cit. 2021-04-15]. Dostupné z: https://www.w3.org/TR/sparql11-query/.

[40] Wang, B. a Liu, M. *Deep learning for aspect-based sentiment analysis*. Stanford University, 2015.

[41] Winter, L. *Algoritmy pro rozpoznávání pojmenovaných entit* [online]. Brno, CZ, 2017. [cit. 2021-04-31]. Master's thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: http://hdl.handle.net/11012/67989.

Příloha A

Spuštění skriptu

Virtuální prostředí

Před spuštěním skriptu je nejprve potřeba vstoupit do virtuálního prostředí, které obsahuje knihovny potřebné ke spuštění skriptu. Pro přístup do virtuálního prostředí je potřeba knihovna **virtualenv**.

Virtuální prostředí je v adresáři **sent10** a lze spustit příkazem:

```
source sent10/bin/activate
```

Vstupní argumenty

Při spuštění skriptu je nutné zadat několik následujících vstupních argumentů:

```
python3 findReviewArticles.py [reviewProduct] [startDate] [endDate] [parProc]
```

- `reviewProduct` - Hledaný produkt pro který budou vyhledány recenze.
- `startDate` - počáteční datum hledaných recenzí ve formátu `YYYY-MM-DD`
- `endDate` - konečné datum hledaných recenzí ve stejném formátu.
- `parProc` - počet procesů pro paralelní zpracování.

Příloha B

Obsah paměťového média

- doc - textová část práce a její zdrojové kódy
- models - modely, slovníky a datové sady využité používané skripty
- result - ukázky výsledků práce
- Scripts - zdrojové kódy skriptů
- sent10 - virtuální prostředí
- training - skript a datová sada využitá k trénování modelu k určení sentimentu
- plakat.pdf - plakát
- readme.txt - textový soubor popisující práci, obsahuje návod spuštění skriptů

Příloha C

Plakát