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

ANALÝZA RECENZÍ VÝROBKŮ

ANALYSIS OF PRODUCT REVIEWS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. ANDREJ KLOCOK**
AUTHOR

VEDOUCÍ PRÁCE — **Doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

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

Zadání diplomové práce

Student: **Klocok Andrej, Bc.**
Program: Informační technologie Obor: Informační systémy
Název: **Analýza recenzí výrobků**
Analysis of Product Reviews
Kategorie: Algoritmy a datové struktury

Zadání:

1. Prostudujte metody identifikace aspektů v uživatelských recenzích a analýzy postojů na základě strojového učení.
2. Navrhněte a implementujte systém, který dokáže pravidelně získávat, indexovat a analyzovat data stahovaná ze srovnávačů zboží.
3. Vytvořte systém pro automatickou klasifikaci shromažďovaných dat, analýzu trendů a vizualizaci výsledků.
4. Demonstrujte vytvořený systém na vhodně zvolených příkladech a statistikách spokojenosti.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle dohody s vedoucím

Při obhajobě semestrální části projektu je požadováno:

- funkční prototyp řešení

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 2019
Datum odevzdání: 3. června 2020
Datum schválení: 1. listopadu 2019

Abstrakt

Zákazníci internetových obchodov generujú obrovské množstvo informácii o službách a produktoch pomocou recenzií, ktoré sú dôležitým zdrojom spätnej väzby. Táto diplomová práca sa zaoberá vytvorením systému pre analýzu recenzií výrobkov a obchodov v českom jazyku. Popisuje doterajšie spôsoby analýzy sentimentu a nadväzuje na aktuálne riešenia. Výsledný systém implementuje automatické sťahovanie dát a ich indexáciu, následne analýzu sentimentu spolu so sumarizáciou textu v podobe zhlukovania podobných viet na základe vektorovej reprezentácie textu. Súčasťou je aj grafické užívateľské rozhranie vo forme webovej stránky. Počas semestra bol vytvorený dataset recenzií s celkovým počtom prevyšujúci šesť miliónov recenzií spolu s rozhraním na jednoduchý export dát.

Abstract

Online store customers generate vast amounts of product and service information through reviews, which are an important source of feedback. This thesis deals with the creation of a system for the analysis of product and shop reviews in the czech language. It describes the current methods of sentiment analysis and builds on current solutions. The resulting system implements automatic data download and their indexing, subsequently sentiment analysis together with text summary in the form of clustering of similar sentences based on vector representation of the text. A graphical user interface in the form of a web page is also included. A review dataset with a total of more than six million reviews was created during the semester along with an interface for easy data export.

Kľúčové slová

spracovanie prirodzeného jazyka, recenzia, analýza sentimentu, klasifikácia, extrakcia dát z webu, strojové učenie, reprezentácia textu

Keywords

natural language processing, review, sentiment analysis, clasification, web scrapping, machine learning, text representation

Citácia

KLOCOK, Andrej. *Analýza recenzí výrobků.* Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

Analýza recenzí výrobků

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána docenta Pavla Smrža. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Andrej Klocok
2. júna 2020

Poďakovanie

Ďakujem vedúcemu mojej diplomovej práce docentovi Pavlovi Smržovi za odbornú pomoc, rady a vedenie pri vypracovaní tejto diplomovej práce.

Obsah

Kapitola 1

Úvod

Od nástupu Web 2.0 sa zvyšuje užívateľmi generovaný obsah a jeho využitie. V dnešnej dobe je čoraz populárnejší termín *sociálne médiá*. Ľudia vykazujú potrebu zdielať svoje postoje, nálady a emócie prostredníctvom rôznych platforiem.

S týmto súvisí aj rozmach internetových obchodov, kde si človek môže kúpiť rôzne výrobky z pohodlia domova. Zákazníci následne prispievajú k rozvoju sociálnych médií prostredníctvom recenzií výrobkov, služieb, ktoré uvádzajú na portály internetových obchodov. Zákazníci radi zdieľajú svoje skúsenosti a názory na jednotlivé produkty a služby. Obsah recenzií je bohatý na subjektívne názory zákazníkov, pričom využitie takýchto znalostí je výhodné jednak pre obchodníka, ktorému sa poskytne spätná väzba k svojim službám, ale aj pre samotného zákazníka, ktorý je nerozhodný a túži si vybrať vždy ten najlepší tovar.

Táto práca sa zaoberá vytvorením systému pre sumarizáciu postojov recenzentov k výrobkom alebo obchodom prostredníctvom jednotlivých recenzií, ktoré sa získavajú zo zrovnávačov produktov. Systém využíva niekoľko modelov domén produktov k analýze sentimentu, pričom dokáže určiť chybné položky recenzií alebo výsledné skóre recenzie.

V rámci sumarizácii postojov systém presahuje analýzu sentimentu a dokáže analyzovať kľúčové atribúty produktu alebo kategórie produktov vykonaním poloautomatického zhlukovania podobných viet, založenom na vektorovej reprezentácii textu. Výsledkom zhlukovania sú aj takzvané príznačné slová pre danú kategóriu alebo produkt, ktorého recenzie sú predmetom analýzy.

Dôraz sa kladie na kvalitu datasetu recenzií, systém využíva filtrovací model na odstránenie irelevantných recenzií, prípadne na upozornenie na nesprávne hodnotenie recenzie.

Ciele tejto práce môžu byť využité priamo zrovnávačom produktov alebo webovým obchodom, ktorý by výstupy analýz mohol použiť priamo pri zobrazovaní produktu s cieľom zvýšenia informovanosti. Obdobné využitie systému môže byť pre dátových analytikov, napríklad pri uvedení nového produktu na trh sa obchodník môže poučiť zo skúseností ľudí s podobným produktom.

Práca je členená do šiestich kapitol. V kapitole 2 je uvedený teoretický pohľad, potrebný pre pochopenie problematiky. Zaoberá sa prehľadávaním webu, indexovaním, analýzou sentimentu a následne uvedením doterajších a aj súčasných prístupov strojového učenia v spracovaní prirodzeného jazyka. Návrhom výsledného systému analýz sa venuje kapitola 3. Nasleduje kapitola 4, v ktorej je popísaná implementácia takéhoto systému. Kapitola 5 popisuje jednotlivé experimenty zamerané na porovnanie datasetu a prevádzané experimenty nad dátami. Záverečná kapitola 6 zhodnocuje ciele systému a dosiahnuté výsledky.

Kapitola 2

Teoretický rozbor

Táto kapitola obsahuje základné teoretické znalosti, z ktorých sa vychádza pri návrhu systému a následne pri jeho realizácii. Táto práca sa zaoberá vytvorením systému pre sumarizáciu obsahu recenzií. Jednotlivé podkapitoly sa viažu k samotnému návrhu systému.

Prvým krokom je samotný spôsob prehľadávania internetového obsahu s cieľom stiahnutia samotných recenzii. Následne je potrebné tieto recenzie reprezentovať ako samostatné dokumenty s cieľom rýchleho vyhľadávania a indexácie. Pre tieto dokumenty je podstatný hlavne postoj autora recenzie k jednotlivým aspektom produktu. Ďalej sú v rýchlosti uvedené základné klasifikačné metódy. Záverečné podkapitoly sú venované algoritmom strojového učenia, ktoré sa používajú pri reprezentácii textu do vektorového priestora.

2.1 Prehľadávanie webu

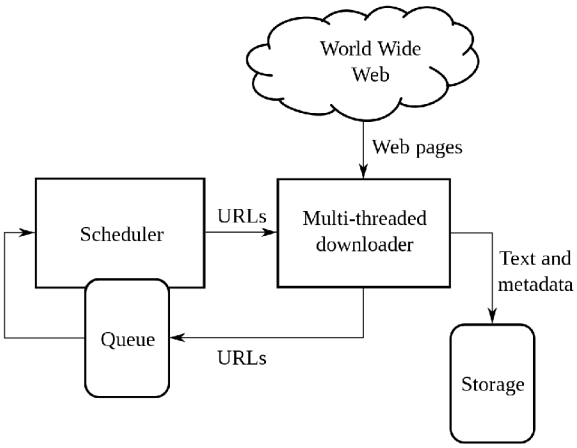
Základom každej analýzy sú dáta, pomocou ktorých je možné sledovať isté trendy, postoje. Získanie týchto dát predstavuje istý problém, pretože obchodné reťazce tieto dáta v internej podobe nezverejňujú, jedná sa o obchodné tajomstvo. Nastáva nutnosť tieto dáta získať z týchto reťazcov nepriamo. Pri vynechaní manuálneho kopírovania dát je najčastejšou technikou prechádzanie internetového obsahu. Táto podkapitola vychádza z publikácii [38] a [21].

Web Scraping

V dnešnej dobe sa na internete zhromažďuje čím ďalej tým viac informácii. Ľudia k týmto informáciám pristupujú väčšinou cez webový prehliadač. Obdobne, *web scraping* je softvérová metóda extrakcie dát z webových stránok. Takýto program simuluje ľudské chovanie prehľadávania WWW buď implementáciou HTTP protokolu na nízkej úrovni alebo integráciu webového prehliadača.

Web scraping sa úzko viaže s pojmom indexácia webového obsahu, vrámci ktorého sa indexujú informácie z internetu, hlavne webové stránky, pomocou webových prehľadávačov, pavúkov, anglicky *web crawler*. Webový prehľadávač pri inicializácii obdrží takzvaný seed stránok, od ktorých začína prehľadávanie. Prehľadáva stránky pomocou rekurzívneho algoritmu a hľadá odkazy na ďalšie stránky, tie si uloží v štruktúrovanej podobe a pokračuje cez nájdené odkazy na ďalšie stránky dokým neprejde všetky. Príklad architektúry prehla-

dávača webu je možné vidieť na obrázku 2.1 Prehľadávače obsahu využívajú hlavne webové vyhľadávacie nástroje, ako je Google[1] alebo Seznam[2].



Obr. 2.1: Architektúra web crawlera [21].

Web scraping sa zaoberá hlavne transformáciou neštruktúrovaných dát na internete, typicky vo formáte HTML, do štruktúrovanej podoby. Dáta bývajú následne analyzované a prevedené do istej podoby v lokálnej databáze. Väčšina internetových stránok nepodporuje možnosť uloženia dát, zobrazovaných na stránke, lokálne do počítača.

Príkladom takého nástroja je open-source nástroj Scrapy[3], ktorý sa používa pre širokú škálu prioritných aplikácii ako data mining, zaznamenávaní údajov priamo zo zdrojov, internetových stránok.

Vynára sa otázka legálnosti konceptu web scrapingu. Veľa ľudí má na túto techniku rôzny názor. Legálnym vyžitím je napríklad porovnanie cien a recenzií obchodov. Existuje niekoľko prípadov, ktoré skončili aj na súde. Napríklad eBay v roku 2000 voči nemenovanému človeku, kvôli použití botov na zbieranie obsahu, daný spor sa nakoniec vyriešil mimo súdnou cestou. Zaujímavý je prípad z roku 2001, kedy nemenovaná cestovná kancelária využila web scraping, aby znížila svoje ceny voči konkurencii alebo Facebook, ktorý v roku 2009 vyhral prvý prípad porušovania autorských práv. Dôležitý je účel dát získaných prehľadávaním internetu. V rámci osobného použitia sa jedná o legálne využitie, pretože to spadá pod doktrínu spravodlivého použitia dát. Problém nastáva v prípade komerčného využitia.

Extrakcia dát

Po dokončení prehľadávania internetového obsahu vznikne súbor URL adries, v ktorých sa nachádzajú požadované informácie. Posledným krokom získania dát je extrakcia užitočných informácii a konvertovanie dát. Je potrebné definovanie obsahu, ktorý je zaujímavý v rámci URL stránky. Následne nájdenie častého vzoru CSS selektorov, ktorý je zhodný aj na ostatných stránkach. Skontrolovania podpory XPath alebo jednoduchých CSS selektorov. Extraktor musí dáta konvertovať do istého formátu, napríklad CSV. V súčasnej dobe existuje knižnica pre web crawling/scrapping skoro pre každý jazyk.

[1] https://www.google.com/
[2] https://www.seznam.cz/
[3] https://scrapy.org/

2.2 Indexácia dokumentov

V tejto podkapitole sú uvedené základné techniky indexovania dokumentov. Indexáciu textu je možné považovať za prvotnú fázu pred spracovania textu na extrahovanie dôležitých štatistických údajov z hľadiska reprezentácie dostupných informácii. Operácie indexácie je možné vykonávať na akomkoľvek type textových informácii [35].

Hlavná myšlienka spočíva v rýchlom vyhľadávaní textu s určitou presnosťou. Získavanie textu (dokumentu) je možné definovať ako zhodu užívateľovho dotazu voči množine textov. Výsledkom je množina textov, zoradených podľa relevantnosti k dotazu užívateľa. Medzi základné indexačné techniky patrí invertovaný index, signatúrové súbory a suffixové stromy [35].

Na koniec podkapitoly je uvedený konkrétny systém, spravujúci indexáciu a vyhľadávanie dokumentov *Elasticsearch* [30][18].

Invertovaný index

Nazývaný aj invertovaný súbor, je dátová štruktúra obsahujúca slovnú zásobu, ktorá obsahuje všetky odlišné indexy slov nachádzajúcich sa v texte. Pre každé slovo `x` zo slovnej zásoby existuje list obsahujúci štatistické údaje o výskyte slova `x` v texte. Tento list sa nazýva invertovaný list slova `x`.

Invertovaný index je navrhnutý pre použitie odlišných slov ako vyhľadávacej jednotky, čo obmedzuje jeho použitie v aplikáciách, v ktorých nie sú slová presne definované. Štatistické údaje vyskytujúce sa v indexe sa líšia od cieľovej domény. Indexácia všetkých výskytov slov je užitočná v aplikáciách, v ktorých by sa mala brať do úvahy aj informácia o pozícii slova, umožnenie vyhľadávania fráz, blízkosť. Invertované indexy sa používajú v systémoch využívajúce modely na získavanie informácii, ako napríklad modely vektorového priestoru, ako napríklad *tf–idf* model 2.5.

Signatúrové súbory

Jedná sa o súbory, ktoré využívajú hašovaciu funkciu na mapovanie slov, nájdených v texte, na bitové masky. Text je rozdelený do blokov a každý blok `b` je indexovaný uložením výsledku bitového `OR` cez všetky masky slov nachádzajúcich sa v bloku `b` signatúrového indexu. Na vyhľadávania sa využíva operácia bitového `AND` medzi hľadanými slovami a maskami blokov v kolekcii. Keď výsledok bitového `AND` je číslo rovnajúcemu sa maske hľadaného slova, jedná sa o potenciálne výsledok. Môžu vzniknúť aj falošné zhody, preto musí byť každá zhoda skontrolovaná na výskyt daného slova.

Slúžia ako filter pre redukciu množstva textu vo vyhľadávacích operáciách. Vyhľadávanie je menej efektívne v porovnaní s invertovaným indexom, ktorý podporuje väčšiu množinu operácii. Signatúrové súbory sú výpočetne náročnejšie na vytvorenie a aktualizáciu.

Sufixové stromy

Ďalšou metódou indexácie textu sú sufixové stromy. Indexujú text ako množinu symbolov, podľa zvolenej granularity vyhľadávajúcej operácie. Napríklad indexom môžu byť všetky znaky textu. Táto flexibilita ponúka využitie v jazykoch, v ktorých nie sú slová zreteľne oddelené od seba, ako napríklad niektoré ázijské jazyky.

Každá pozícia v texte sa nazýva sufix, prípona. Každý sufix je definovaný začiatočnou pozíciou a predĺžením ľubovolne doprava až na koniec textu. Formát sufixu musí byť defi-

novaný v rámci sekvencie znakov, ktoré sa budú vyhľadávať. Pri využití sufixového modelu je nutné považovať každý vstupný bod indexovaný v texte ako sufix celého textu. Každá cesta od koreňa k listu reprezentuje unikátny sufix.

Táto štruktúra sa využíva pri komplexných vyhľadávajúcich metódach, kvôli nízkym výpočetným nákladom. Nevýhodou je priestor na uloženie indexu. Alternatívou môže byť využitie súboru prípon, pole ukazateľov na každý sufix v texte. Toto pole je abecedne zoradené, pričom sa zníži priestorová zložitosť, naopak cena vyhľadávajúcich operácii sa zvyšuje. Praktický problém využitia sufixového modelu je cena vybudovania a udržiavania. Vyhľadávanie slov je zvyčajne rýchlejšie pri použití invertovaného indexu, s výnimkou regulárnych výrazov.

Elasticsearch

Podľa portálu database-engines[4], ktorý slúži ako vedomostná základňa relačných a NoSQL systémov správy databáz, sa ako najlepšie riešenie pre indexáciu dokumentov a samotné rýchle vyhľadávanie dokumentov javí práve *Elasticsearch*[5](pre December 2019). Je na prvom mieste, v rámci vyhľadávania dokumentov a v celkovom hodnotení je na 7. pozícii.

Elasticsearch je distribuovaný, bez schémový databázový systém, ktorý sa zaraďuje do kategórie indexácie veľkých dát. Ponúka efektívne a full–textové vyhľadávanie prístupne pomocou otvoreného API. Je postavený nad Apache Lucene[6] s jednoduchým REST rozhraním. Spracovanie obrovského objemu dát so schopnosťou identifikovania kľúčových slov v reálnom čase činí elastic potenciálne vhodným kandidátom, napríklad na spracovanie dát zo sociálnych sietí ako Twitter, Facebook

Elasticsearch obsahuje isté podobnosti s SQL databázami. Namiesto databáze je `index`, namiesto tabuľky je `type` (typ dokumentu), namiesto riadku tabuľky je samotný `document`, stĺpec databáze nahradzuje položka `field`.

Keďže je elasticsearch distribuovaný, je môžné architektúru rozdeliť do kolekcie zhlukov, ktoré môžu navzájom komunikovať a zdielať zodpovednosť za uložené dáta. V rámci zhluku musí existovať aspoň jeden uzol. Elasticsearch používa takzvané úlomky a indexy. Úlomky sú inštancie *Lucene* indexov, elastic je absrakcia nad Lucene indexom v distribuovanom systéme. Každý index sa skladá z úlomkov naprieč jedným alebo viacerými uzlami. Každý primárny úlomok môže mať repliky naprieč ostatnými uzlami. Dokumenty sú distribuované rovnomerne medzi všetky primárne úlomky.

2.3 Analýza sentimentu

Táto podkapitola sa zaoberá analýzou ľudských postojov. Cieľom analýzy je určenie emočne zafarbeného postoja človeka k nejakej entite, produktu, službe. Tieto postoje sú zaujímavé napríklad z hľadiska zákazníka, ktorý sa rozhoduje ohľadom kúpi produktu z daného obchodu. Každý človek chce pre seba to najlepšie, najlepšie hodnotený produkt, službu. S rozmachom internetu v spojení so sociálnymi médiami sa čoraz častejšie dostáva človek do styku s názormi, ktoré ho ovplyvňujú.

V rámci analýzy je potrebné rozlišovať subjektívne, objektívne názory. Analýza býva prevádzaná na rôznych úrovniach granularity. Jedná sa o analýzu sentimentu na úrovni dokumentu, analýza sentimentu na úrovni vety, analýza sentimentu na úrovni aspektu.

[4] https://db-engines.com/en/ranking/search+engine
[5] https://www.elastic.co/
[6] https://lucene.apache.org/

Postoj je možné vyjadriť priamo k danej entite alebo pomocou porovnávajúceho názoru, kde sa porovnáva viacero entít naraz.

Význam emócií

Konkrétny popis emócii je zložitý, ako je možné sa dozvedieť v monografii [40]. Existuje rozdiel medzi popisným významom, to čo je v texte uvedené a samotným emociálnym významom. Popisný význam vyjadruje objektívne fakty, napríklad *„Procesor má šestnásť fyzických jadier“*, zatiaľ čo emocionálny vyjadruje subjektívne postoje *„Táto farba sa mi nepáči“*. Emocionálny význam sa líši s každým poľom pôsobnosti. K emocionálnemu významu je nutné pristupovať ako ku kompozičnému, pozostávajúceho z mnohých jazykových aspektov. Hodnotenie je vždy vyjadrením emócii a tie vždy majú hodnotiaci charakter, čo znamená, že je ich možné väčšinou kategorizovať.

Definícia názoru

Názor je možné vyjadriť voči istej entite e, ktorá je definovaná podľa publikácie [19], ako produkt, služba, osoba, organizácia, téma. Je spojená s párom $e : (T, W)$, kde T je hierarchia komponent, subkomponent a W je množina atribútov. Každá komponenta, subkomponenta má svoju vlastnú množinu atribútov. Napríklad značka mobilných telefónov je entita, napríklad *Samsung*. Má sadu komponent ako baterka, displej a sadu atribútov ako výdrž batérie, rozlíšenie displeja, hmotnosť.

Samotný názor je možné definovať ako päticu $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$, kde e_i je názov entity, a_{ij} je aspekt entity e_i, oo_{ijkl} je orientácia názoru voči aspektu a_{ij} entity e_i, h_k je držiteľ názoru a t_l je čas, kedy bol postoj vyjadrený h_k. Postoj oo_{ijkl} môže byť pozitívny, negatívny, neutrálny alebo vyjadrený pomocou rôznych úrovní intenzity.

Analýza sentimentu na úrovni dokumentu

Analýza prebieha na úrovni celého dokumentu, príspevku, recenzie. Klasifikáciu sentimentu na úrovni dokumentu d vzťahujúceho sa k entite e vyjadrujúceho názor oo na entitu e, vzťahujúci na aspekt $GENERAL$ je definovaný podľa publikácie [19] päticou $(e, GENERAL, oo, h, t)$, pri predpoklade, že e, h, t sú známe alebo irelevantné. Tento predpoklad je možné uplatniť pri recenziách, kedy jeden zákazník hodnotí jeden produkt, službu. Nemusí to platiť napríklad v prípade blogov, príspevkov na fórach atď.

Analýzu je možné definovať ako tzv. *„supervised learning problem“*, kedy sú definované tri triedy, ktoré vyjadrujú konečný postoj: pozitívny, negatívny, neutrálny. V prípade recenzií sa vo väčšine prípadov vyskytuje aj kolónka skóre v intervale $[1, 5]$, prípadne $[0, 100]$.

Podstatou riešenia tejto úlohy je forma zobrazenie textu do vektorového priestoru, klasické riešenia sú popísané v podkapitole 2.5, na ktoré naväzujú súčasné riešenia uvedené od podkapitoly 2.6. Tieto vektory sú následne vstupom klasifikačných algoritmov.

Medzi problémy analýzy sa zaraďujú rozdielne názory v rámci dokumentu, čo sa týka polarity. Fakt, že ľudia môžu vyjadriť zhodný názor viacerými spôsobmi, bez využitia tzv. názorových slov. Takýto dokument obsahuje subjektívne a objektívne vety, čo navádza k analýze sentimentu na úrovni vety [17].

Analýza sentimentu na úrovni vety

Obdobne je možné využiť podobné techniky z analýzy na úrovni dokumentu 2.3. Podľa publikácie [19] je možné definovať analýzu sentimentu na úrovni vety s pomocou dvoch

podúloh. Klasifikácia subjektivity, pozostáva z určenia, či daná veta je subjektívna alebo objektívna. Klasifikácie sentimentu na úrovni vety, ak je veta subjektívna je potrebné zistiť postoj, ktorý veta vyjadruje a to pozitívny, negatívny, neutrálny. Obe klasifikačné úlohy je možné riešiť obdobne, ako v analýze na úrovni dokumentu 2.3.

Detekcia subjektivity môže byť vykonávaná aj pomocou takzvaného subjektívneho lexikónu. Problém nastáva vtedy, keď aj objektívne vety obsahujú názory [17].

Predpoklad toho, že veta vyjadruje jeden názor od jedného držiteľa je splnený len v jednoduchých vetách. V prípade komplexnejších viet môže jedna veta vyjadrovať viacero názorov, či už pozitívnych alebo negatívnych, ktoré sa viažu k jednotlivým aspektom.

Názorový lexikón

Postoj k danej entite sa vyjadruje spolu s názorovým slovom, v literatúre sa tieto slová nazývajú aj slová sentimentu [19]. Pozitívne slová ako *krásny*, *úžasný* sa používajú k vyjadrení žiadúcich stavov, negatívne slová ako *hrozný*, *pokazený* k vyjadrení nežiadúcich stavov. Tiež je možné využitie názorových fráz, idiómov. Spoločne sa nazývajú lexikón názorov.

Okrem manuálneho zostavovania lexikónu sa používajú automatizované prístupy založené na slovníkoch alebo korpusoch [17]. V prístupe založenom na slovníkoch je potrebné vytvorenie malej množiny názorových slov, ktorých polarita je už známa. Následne využitím slovníka nájsť všetky synonymá alebo antonymá. Tento prístup nie je schopný nájsť názorové slová s špecifickou polaritou v rámci domény. Korpusový prístup využíva tiež množinu názorových slov, ktorých polarita je známa. Závisí na syntaktických alebo výskytových vzoroch, ktoré sa vyhľadávajú naprieč korpusom.

Analýza sentimentu na úrovni aspektu

Klasifikácia sentimentu na úrovni dokumentu alebo vety neponúka všetky potrebné informácie. Pozitívne hodnotený dokument alebo veta vzťahujúca sa k nejakej entite neznamená, že autor sa vyjadril pozitívne ku každému aspektu, obdobne negatívne hodnotený dokument alebo veta. V dokumente sa autor vyjadruje k viacerým aspektom s rôznou polaritou názoru. Analýzu je potrebné robiť na nižšej úrovni a to úrovni aspektu.

Podľa [19] je potrebné v rámci dokumentu d nájsť každú päticu $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$. Vynárajú sa dve hlavné úlohy a to jednak extrakcia aspektov a následná klasifikácia sentimentu na úrovni aspektu.

Extrakcii aspektov sa venuje sekcia 3.6, pričom keď sa hovorí o aspekte, je potrebné vedieť ku ktorej entite sa viaže. Pri klasifikácii sentimentu je možné využiť metódy vektorizácie textu obdobne ako pri úrovni dokumentu 2.3.

Napríklad vo vete *„Zvuk tohto reproduktoru je vynikajúci“* je možné sledovať vyjadrenie pozitívneho postoja vyjadreného slovom *vynikajúci*, aspektu *zvuk*, k entite *reproduktor*. Problém vzniká pri vyjadrení zmiešaného názoru vo vete alebo využitím fráze. Tento problém je možné riešiť lexikónom.

Aspekty sa rozdeľujú na explicitné a implicitné. Prvé z nich označujú cieľ stanoviska priamo a druhé predstavujú aj cieľ postoja k dokumentu, ale v texte sa explicitne nenachádzajú.

2.4 Klasifikačné metódy

V tejto podkapitole sú popísané klasifikačné metódy, využívané pri klasifikácii sentimentu. Doterajšie riešenia spočívali v klasických algoritmoch, medzi ktoré patrí Naive Bayes, SVM [12]. Mnohé štúdie analýz sentimentu používajú vyššie spomínané klasifikačné metódy.

Súčasné riešenia sú založené na technológií neurónových sietí [25][32], pričom tieto algoritmy sú využité výslednom systéme sumarizácie postojov recenzií.

Naive Bayes

Klasifikačný model Naive Bayes počíta aposteriórnu pravdepodobnosť triedy na základe rozloženia slov v dokumente. Model je možné využiť napríklad s *BOW* modelmi, ktoré ignorujú pozíciu slova v dokumente, popísané v podkapitole 2.5. Využíva sa *Bayesov teorém* na predikciu pravdepodobnosti, že daná množina prvkov (*features*) patrí do konkrétnej značky (*label*):

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)} \tag{2.1}$$

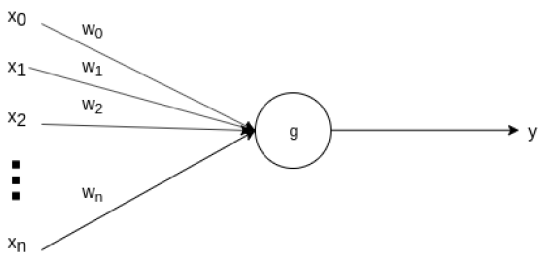
Kde $P(label)$ je pravdepodobnosť, že náhodný prvok patrí do značky. $P(features|label)$ je pravdepodobnosť, že daná množina prvkov je klasifikovaná do danej značky a $P(features)$ je pravdepodobnosť výskytu množiny prvkov.

SVM

Princíp klasifikačného modelu je v určení hyper-roviny vo vyhľadávacom priestore, ktorá rozdeľuje priestor na čo najlepšie priestory pre rôzne triedy. Optimálna hyper-rovina maximalizuje vzdialenosť medzi triedami. SVM sú vhodné na klasifikáciu textu, kvôli jeho riedkej povahe, málo prvkov je irelevantných a vo všeobecnosti sú korelované medzi sebou, organizované do lineárne oddeliteľných kategórii.

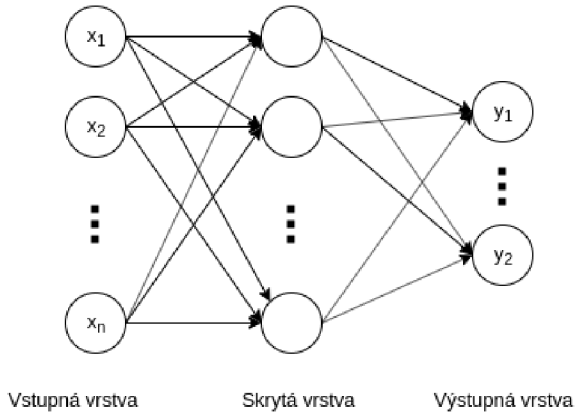
Neuronové siete

Neuronové siete kopirujú štruktúru ľudského mozgu. Základnou abstrakciou je neurón, jeden blok siete, obrázok 2.2. Vo väčšine prípadov neurón tvorí vážený súčet veľkého počtu vstupov $x_1, x_2, \ldots x_n$, ich váh $w_1, w_2, \ldots w_n$ (aktivácií iných neurónov) $w_0x_0 + \sum_{i=1}^{n} w_ix_i$, pričom w_0x_0 sa nazýva bias. Aplikuje nelineárnu aktivačnú funkciu prenosu na túto sumu a výsledný výstup vysiela na veľké množstvo ďalších neurónov. Abstrakcia činnosti biologického neurónu. Medzi aktivačné funkcie patrí step, sigmoid, tanh a ReLU funkcia [25].



Obr. 2.2: Matematický model neurónu.

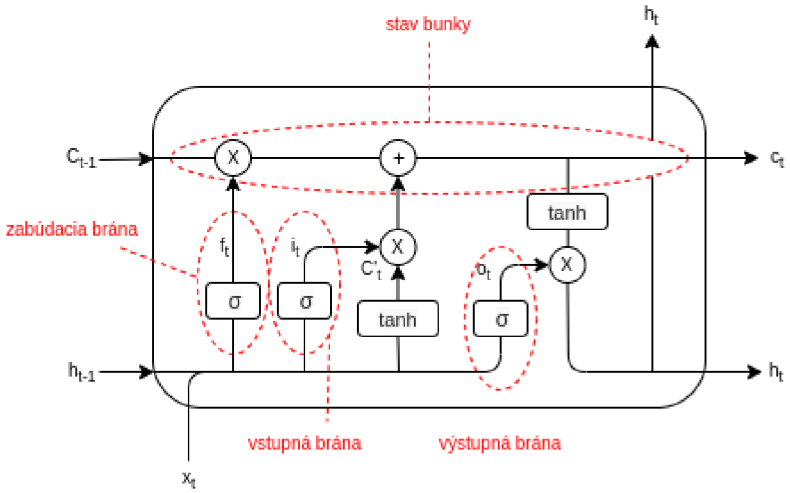
Neurónovej sieť predstavuje orientovaný graf, uzly predstavujú jednotlivé neuróny, ktoré sú prepojené hranami udávajúcimi váhy. Topológia pozostáva z takzvaných vrstiev, kolekcie neurónov, zobrazených na obrázku 2.3. Tieto neuróny sú prepojené s neurónmi ďalších vrstiev. Najľavejšia vrstva sa nazýva vstupná vrstva, najpravšia vrstva sa nazýva výstupná. Vrstvy medzi sa nazývajú tzv. skryté vrstvy. Každá vrstva môže mať inú aktivačnú funkciu. Podľa architektúry neurónovej siete sa signál môže šíriť dopredne, spätne.



Obr. 2.3: Model neurónovej siete.

LSTM

Architektúra LSTM (angl. *long short term memory*) je zástupca rekurentných neurónových sietí, pričom je schopná sa naučiť dlhodobé závislosti. Jadro architektúry pozostáva zo stavu bunky, ktorý je kontrolovaný pomocou brán. Brány predstavujú spôsob, akým je možné voliteľne prepustiť informácie. Pozostávajú zo sigmoidnej vrstvy neurónovej siete a operácie bodového násobenia. LSTM obsahuje tri takéto brány na chránenie a kontrolu stavu bunky. Schéma architektúry je zobrazená na obrázku 2.4.



Obr. 2.4: Architektúra LSTM [26].

Prvý krok pozostáva z rozhodnutia, ktoré informácie sú zahodené zo stavu bunky pomocou brány zabúdania, popísané rovnicou 2.2. Výstupom je hodnota v intervale $[0, 1]$ pre každú hodnotu v stave bunky C_{t-1}.

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{2.2}$$

Ďalší krok pozostáva z rozhodnutia, ktoré nové informácie sa uložia do stavu bunky. Vstupná brána rozhodne, ktoré hodnoty i_t sa aktualizujú. Tahn vrstva vytvorí vektor nových kandidátov $C_t^{'}$, ktoré môžu byť pridané do stavu bunky. Následne ich kombináciou sa aktualizuje starý stav bunky C_{t-1} na nový stav C_t.

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{2.3}$$

$$C_t^{'} = \tanh(W_C \times [h_{t-1}, x_t] + b_C) \tag{2.4}$$

$$C_t = f_t * C_{t-1} + i_t * C_t^{'} \tag{2.5}$$

Výstup je založený na filtrovanom stave bunky. Pomocou výstupnej brány sa rozhodne, ktoré hodnoty stavu bunky budú súčasťou výstupu. Následne stav bunky je prevedený pomocou tanh funkcie a vynásobený s výstupom výstupnej brány.

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{2.6}$$

$$h_t = o_t * \tanh(C_t) \tag{2.7}$$

2.5 Klasické prístupy mapovania slov

Vstupom algoritmov strojového učenia sú tzv. vektory numerických príznakov. Príznaky vyjadrujú určitú vlasnosť. Anglický termín *word embedding*, je možné preložiť ako mapovanie slova do vektorového priestoru. Táto technika je kľúčová vrámci spracovania prirodzeného jazyka. Medzi základné prístupy mapovania slov do vektorovej reprezentácie patria modely BOW, na ktoré naväzuje LDA.

Bag of words

Model reprezentovanie textu ako multimnožiny slov [27], pričom sa stráca syntaktická informácia o poradí slov, sémantická informácia o kontexte slova v dokumente. Idea modelu spočíva v tom, že podobné dokumenty obsahujú podobný obsah.

BOW model spočíva vo vytvorení slovníku slov, pričom každému slovu je pridelený jednoznačný index. Dokument je reprezentovaný vektorom dĺžky n, v ktorom i–ty záznam obsahuje počet výskytov slova v dokumente.

Index môže mať charakter one-hot encoding podľa slov v slovníku (1 – slovo sa nachádza v slovníku, 0 naopak), frekvenciu slova v dokumente, TF–IDF frekvenciu a podobne.

Bag of n–grams

Pri použití BOW sa stráca informácia o poradí slov. Túto informáciu je možné čiastočne získať použitím modelu, ktorý nepracuje len so samotnými slovami, unigrammi, ale priamo so sekvenciou slov dĺžky n [27], n–gramoch. Nevýhodou tejto metódy je nelineárna závislosť slovníka od počtu jednotlivých slov, ktorá je pre rozsiahle korpusy veľká.

TF–IDF

Poslednou technikov BOW je metóda *term frequency-inverse document frequency* [27]. Táto metóda odráža dôležitosť slova, n-gramu pre dokument v korpuse. Určuje jeho váhu na základe výskytu v rôznych dokumentoch. Hodnota sa zvyšuje úmerne s výskytom slova v dokumente a je kompenzovaná počtom dokumentov v korpuse, v ktorých sa vyskytuje dané slovo. Vhodné využitie pre doménovo špecifický slovník.

$$TF_i = \text{frekvencia } term_i \text{ v dokumente} \tag{2.8}$$

$$IDF_i = \log(\frac{\#\text{ dokumentov v korpuse}}{\#\text{ dokumentov obsahujúcich } term_i}) \tag{2.9}$$

$$TF - IDF_i = TF * IDF_i \tag{2.10}$$

Latent Dirichlet allocation

Jedná sa o hierarchický bayesov model [42], ktorý dokáže vysvetliť podobnosť medzi dokumentami na základe tém, ktoré sa v nich vyskytujú. Pri predpoklade toho, že každý dokument pozostáva z malého počtu tém. Téma je charakterizovaná prítomnosťou slova v dokumente.

Naviazanie na BOW spočíva v tom, že je ho možné chápať ako zjednodušený pravdepodobnostný model dokumentov pomocou distribúcie slov [27]. BOW vektor predstavuje najlepšiu aproximáciu nenormalizovanej distribúcie slov v každom dokumente. V LDA je dokument základná pravdepodobnostná jednotka. Témy sú charakterizované distribúciou slov, zatiaľ čo dokumenty distribúciou cez témy.

Tento pravdepodobnostný model dokumentu zodpovedá generatívnemu modelu dokumentov. Na vygenerovanie množiny dokumentov M dĺžky $\{N_i\}$, sa predpokladá vopred určený počet tém K, kde $Dir()$ označuje Dirichletovu [42] distribúciu:

1. Pre každú tému v, nájdi distribúciu slov $\phi_v \sim Dir(\beta)$.
2. Pre každý dokument i, nájdi distribúciu tém $\theta_i \sim Dir(\alpha)$
3. Na vygenerovanie dokumentu i dĺžky N_i, pre každé slovo j:
   - Nájdi tému $z_{ij} \sim Multinomial(\theta_i)$ pre slovo j
   - Nájdi slovo $j \sim Multinomial(z_{ij})$

Kde θ_i je distribúcia tém pre každý dokument i vektorov dimenzií K.

Vzniká vektorový priestor dimenzie K, ktorý zachytáva témy v korpuse a spôsob, akým sú v ňom zdielané. Je možné ho považovať za *embedding* pre tieto dokumenty. V závislosti na hodnote K, môže mať podstatne menší rozmer.

2.6 Statické mapovanie slov

Predchádzajúca podkapitola 2.5 vysvetľuje metódy mapovania slov do vektorového priestoru, pričom sa stráca sémantika a syntax. Zaradenie slova do istého kontextu vystihuje známy citát: *You shall know a word by the company it keeps (Firth, J. R. 1957:11)*. Kontext slova je možné odvodiť od slov, ktoré ho sprevádzajú. Napríklad vektorová reprezentácia slova *muž* bude veľmi podobná vektoru slova *žena*. Naopak vektor slova *kameň* bude odlišný. Podobnosť je definovaná výskytom slov použitých v rámci kontextu.

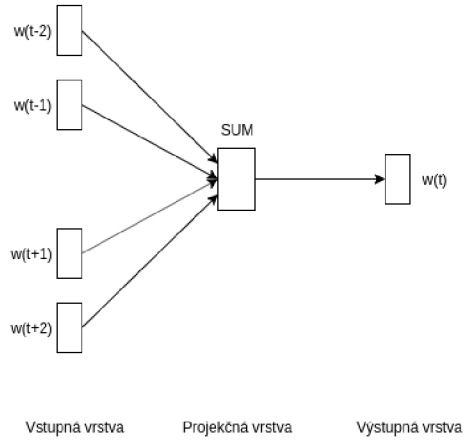
Word2vec

Bolo navrhnutých mnoho modelov pre odhad nepretržitej postupnosti slov, napríklad LDA 2.5. LDA sa stáva výpočetne veľmi náročné pri veľkých datasetoch. Podľa článku [22] môže byť využitý jazykový model neurónovej siete v dvoch krokoch. V prvom sa pomocou jednoduchého modelu naučia súvislé slovné vektory a potom n–gram NNLM (dopredného jazykového modelu neurónovej siete) je naučená nad touto distribuovanou reprezentáciou slov.

Dopredný jazykový model neurónovej siete pozostáva zo vstupných, projekčných, skrytých a výstupných vrstiev. Vo vstupnej vrstve je zakódovaných N slov kontextového okna pomocou $1 - z - V$ kódovania, kde V je veľkosť slovníka. Vstupná vrstva je premietnutá do projekčnej vrstvy rozmeru $N \times D$, využitím projekčnej matice. Projekčná vrstva je pripojená do skrytej vrstvy o rozmere H. Skrytá vrstva slúži na výpočet rozdelenia pravdepodobnosti na všetky slová v slovníku, čo vedie k výstupnej vrstve rozmeru V, tento krok je výpočetne náročný, jeho zložitosť je $H \times V$, dá sa optimalizovať využitím *hierarchického softmaxu*, ktorý redukuje $H \times \log_2(V)$.

Hierarchický softmax [23] využíva binárnu stromovú reprezentáciu výstupnej vrstvy s W slovami ako jeho listy a pre každý uzol explicitne predstavuje relatívnu pravdepodobnosť jeho synov, ktorí definujú náhodný krok, ktorý priradí pravdepodobnosť slovám.

Cieľom je však extrakcia *word embeddings*, pričom je možné kompenzovať presnosť odobratím nelineárnej skrytej vrstvy a použiť logaritmický model, ako CBOW, Skip-gram.



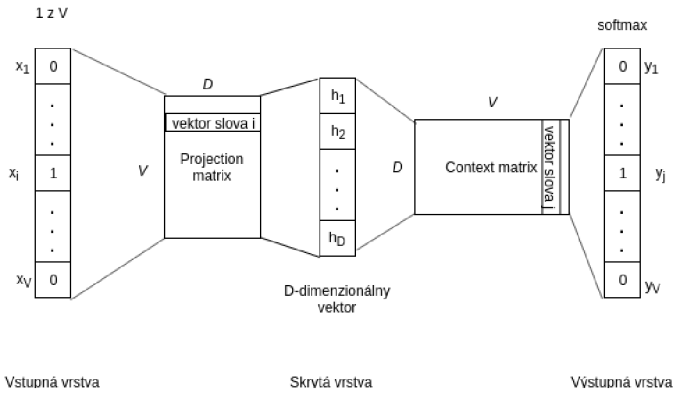
Obr. 2.5: Architektúra modelu CBOW.

CBOW

Prvým z logaritmických modelov je *Continuous Bag-of-Words Model*. Nezohľadňuje poradie slov v histórii. Architektúra je zhodná s NNLM s výnimkou nelineárnej skrytej vrstvy, ktorá je odstránená a projekčná vrstva je zdielaná pre všetky slová, nie len projekčnou maticou. Teda všetky slová sa premietnu do tej istej pozície a ich vektory sa spriemerujú. Architektúra CBOW je zobrazená na obrázku 2.5, ktorá predikuje cieľové slovo $w(t)$, na základe statického okolia slov, ktoré ho sprevádzajú v určitom okne o veľkosti N.

Skip–gram

Druhým z logaritmických modelov je *Continuous Skip–gram Model*, je podobný modelu CBOW, ale nepredikuje slovo na základe kontextu. Snaží sa maximalizovať klasifikáciu slova, na základe iného slova v rovnakej vete. Používa každé aktuálne slovo ako vstup do logaritmického lineárneho kvalifikátora so súvislou projekčnou vrstvou a predpovedá slová v určitom rozsahu pred a za súčasným slovom. Architektúra je zobrazená na obrázku 2.6.



Obr. 2.6: Architektúra modelu Skip–gram.

GloVe

Pojem *GloVe* značí *„globálne vektory“*. Jedná sa o model [28], ktorý zachytáva globálne a lokálne štatistiky v korpuse, aby mohol generovať vektory slov, pričom nevyužíva neuronové siete. Model je založený na myšlienke využitia matice spoločného výskytu, z ktorej je možné odvodiť sémantické vzťahy medzi slovami. Nech existuje korpus o veľkosti V slov, matica spoločného výskytu X rozmeru $V \times V$, kde $X[i, j]$ značí počet spoločných výskytov slov i a j.

GloVe predpokladá okolité slová maximalizáciou pravdepodobnosti výskytu kontextového slova pri danom stredovom slove, vykonaním logistickej regresie. Pri predpovedaní korelácie slovných vektorov sa definuje neznáma funkcia F 2.11, ktorá má za vstupy vektory slov i, j, k [15].

$$F(w_i, w_j, \tilde{w_k}) \approx \frac{P_{ij}}{P_{jk}} \tag{2.11}$$

P_{ij} značí pravdepodobnosť výskytu slova j v kontexte slova i. Funkcia využíva dva typy embeddingov, vstupný w a výstupný $\tilde{w}$ pre kontextové a cieľové slovo. Glove si dáva za cieľ vytvoriť vektory so zmysluplnými dimenziami pri využití jednoduchej aritmetiky. Najjednoduchší spôsob je zmeniť vstup funkcie F na rozdiel vektorov, ktoré sa porovnávajú. Vytvorením lineárnej relácie medzi rozdielom vektorov a výstupným vektorom pomocou skalárneho súčinu vznikne rovnica 2.12.

$$dot(w_i - w_j, \tilde{w_k}) \approx \frac{P_{ij}}{P_{jk}} \tag{2.12}$$

Využitím logaritmu pravdepodobností je možné previesť pomer na odčítanie pravdepodobností a pridaním biasu pre každé slovo sa zachytí fakt, že niektoré slová sa vyskytujú

častejšie ako iné. Využitím matice susednosti sa rovnica prevedie na rovnicu nad jedným záznamom. Po absorbovaní konečného termínu na pravej strane a pridaní biasu pre výstupný vektor, kvôli symetrii, sa dá dopracovať ku rovnici, ktorá stojí za GloVe 2.13.

$$dot(w_i, \tilde{w_k}) + b_i + \tilde{b_k} = \log(X_{ik}) \tag{2.13}$$

Spoločné výskyty slov, ktoré sa nevyskytujú tak často, majú tendenciu spôsobovať šum a byť nespoľahlivé. Využíva sa váhová funkcia f 2.14.

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{ak } x \leq x_{\max} \\ 1 & \text{inak} \end{cases} \tag{2.14}$$

GloVe a word2vec používajú na optimalizáciu odlišné metódy, pričom sú matematicky podobné. Word2vec nerozkladá maticu spoločného výskytu, ale optimalizuje sa cez jednu maticu pomocou prúdu viet.

2.7 Dynamické mapovanie slov

Statické modely nedokážu zachytiť takzvanú mnohoznačnosť. Generujú ten istý vektor pre tie isté slová aj v inom kontexte. Dynamické mapovanie slov [33] cieli na zachytávanie sémantiky slov v rôznych kontextoch. Výstupom dynamických modelov je natrénovaný model a vektory.

Tradičné vektory slov predstavujú plytkú reprezentáciu, jednu vrstvu váh známych ako *embeddings*. Predchádzajúce vedomosti začleňujú len do prvej vrstvy modelu. Zbytok siete musí byť natrénovaný od začiatku na novú úlohu.

Úlohou jazykového modelovania je priradenie rozdelenia pravdepodobnosti podľa postupnosti slov, ktoré zodpovedajú distribúcii jazyka. Jazykové modelovanie (t.j. ELMo 2.7, Bert 2.8) predpovedá pravdepodobnosť výskytu slova v kontexte.

ELMo

Pojem *ELMo* znamená ***E**mbeddings from **L**anguage **Mo**dels* [29][14]. Elmo využíva jazykové modely k získaniu embeddingov pre jednotlivé slová, pričom zohľadňuje celú vetu. Význam slova je závislý od kontextu, v ktorom sa vyskytuje. Konkrétne využíva pred–trénovaný, viacvrstvový, obojsmerný jazykový model založený na LSTM 2.4. Extrahuje skrytý stav každej vrstvy pre vstupnú postupnosť slov, potom vypočíta váženú sumu týchto skrytých stavov, aby získal embedding pre každé slovo, pričom váha každého stavu je závislá na úlohe.

Kontext slova je možné prezentovať napríklad na slove *vlna*. Vo vete *„Pri pláži sa objavila veľká vlna“* a vete *„Vlna vzniká strihaním oviec“*. Vektor, reprezentujúci slovo vlna, by sa mal meniť vždy od kontextu, v ktorom sa nachádza.

LSTM jazykový model

Jazykový model dokáže určiť pravdepodobnosť, že daná sekvencia slov bude reálny text. Tento model sa dá naučiť pomocou predpokladania nasledujúceho slova vo vete. Presnejšie povedané, určenie pravdepodobnosti nasledujúceho slova pre všetky slová v slovníku.

Autori článku [29] ponúkajú metódu trénovania jazykového modelu pomocou reverzovaných viet, *spätný jazykový model*. Podobný prístup ako využitie obojsmerných LSTM

pre klasifikáciu viet. Architektúra využíva viacej vrstiev LSTM. Pri trénovaní n-vrstiev LSTM je možné získať $2n + 1$ vektorov na reprezentáciu kontextu pre každé slovo. Modely ako *word2vec* sa snažia predikovať slová, na základe ich kontextu, naučiť model základné vlastnosti jazyka, ktorý spracováva.

Model

Reprezentáciu slova je možné vyjadriť pomocou rovnice:

$$ELMo_k = \gamma \Sigma_j s_j h_{kj} \tag{2.15}$$

kde k je index slova, j je index vrstvy, z ktorej sa prvok extrahuje, h_{kj} je výstup j–tej vrstvy LSTM pre slovo s_j s váhou h_{kj}. γ závisí od úlohy, slúži na optimalizáciu, škálovanie vektoru. Váha sa počíta pre každú úlohu a normalizuje sa pomocou soft-max.

Vyššie vrstvy zachytávajú kontextovo závislé aspekty slovných embeddingov, zatiaľ čo nižšie vrstvy zachytávajú modelové aspekty syntaxe [20].

Podrobnejšie informácie o presne použitej architektúre sú dostupné v publikácii [29]. Trénovanie modelu je náročné. Autori ponúkajú pred-trénované modely[7] pre rôzne jazyky, pričom sa odporúča takzvané ladenie modelu na doménovo špecifických dátach. ELMo dosiahlo state–of–the–art výkon v úlohách ako:

- Odpovedanie na otázky - dátový súbor Standford Question Answering (SQuAD), v ktorom sú odpovede na otázky obsiahnuté v texte v odsekoch na Wikipedii.
- Textové zhrnutie – zistenie pravdivosti prehlásenia na základe predpokladu.
- Sémantické označovanie rolí – model musí určiť štruktúru vety predikovaného argumentu.
- Rozlíšenie koreferencie – model musí identifikovať, ktoré časti textu sa vzťahujú na istú entitu.
- Extrakcia menovaného subjektu – model klasifikuje rôzne pomenované entity do súboru rôznych tried, ako sú napríklad osoba a miesto.
- Analýza sentimentu – jednoduchá klasifikácia textu, pri ktorej označenie zodpovedá miere pozitivitu, negativitu v texte.

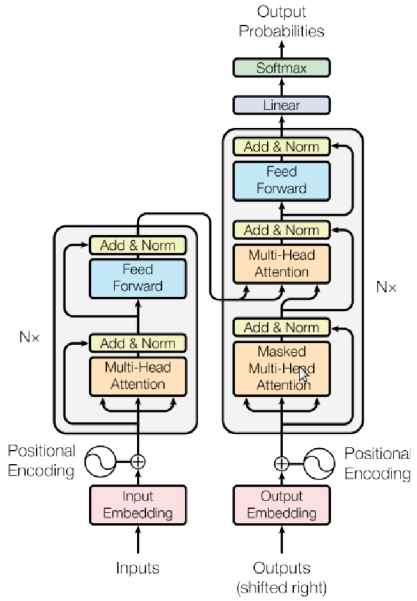
2.8 Transformers

Publikácia [39] uvádza novú architektúru nazývanú *Transformer*, ktorá pracuje na princípe pozornosti. Vstupom modelov pozornosti je sekvencia slov a všetky jej skryté stavy s ňou spojené. Výstup je založený na zistení, ktorá časť vstupu je užitočná. Sekvenčná povaha sa dá zachytiť využitím napríklad LSTM, v prípade architektúry tranformer, je táto povaha zachytená pomocou mechanizmu pozornosti.

Tento mechanizmus sleduje vstupnú sekvenciu slov, v každom kroku rozhoduje, ktoré ďalšie časti sekvencie sú dôležité. Napríklad pri čítaní textu sa človek sústredí na aktuálne slovo a súčasne si jeho myseľ pamätá dôležité kľúčové slová textu, aby mohla poskytnúť kontext [1].

[7] https://allennlp.org/elmo

Architektúra transformerov pozostáva zo štruktúry kodér-dekodér, zobrazená na obrázku 2.7, ktorá je súčasť pôvodných tzv. pozorných sietí – pomocou vstupnej sekvencie sa vytvorí jej kódovanie na základe kontextu a dekóduje kontextové kódovanie na výstupnú sekvenciu.



Obr. 2.7: Transformer – architektúra modelu [39].

Vrstva pozornosti nahradzuje LSTM, sekvenčná povaha je identifikovaná pomocou pozičného kódovania. Všetky komponenty sú tvorené z úplne spojených vrstiev, všetky neuróny jednej vrstvy sú spojené so všetkými neurónmi druhej vrstvy, teda je jednoduché sieť paralelizovať [10].

Sekvencia (veta) najskôr prebehne tokenizovaním, pričom vstupná sekvencia pozostáva z embeddingov každého tokenu vety. Embedding môže predstavovať napríklad *1–z–V* kódovanie. Ďalej je nutné zahrnúť informáciu o relatívnej pozícii tokenu v sekvencii pomocou pozičného kódovania. Cieľom je modifikovať reprezentáciu tokenu, na základe jeho pozície. Autori [39] navrhli použiť sínusoidu na reprezentovanie pozície tokenu, pozícia môže byť reprezentovaná vzdialenosťou medzi rozdielnymi slovami v sekvencii.

Kodér

Vrstva pozornosti (*self–attention*) [10] sa snaží zakódovať slovo pomocou ostatných slov v sekvencii. Pre každý vstup kodéru (embedding) sa vygenerujú 3 vektory s dimenzionalitou menšou ako embedding: dotaz, kľúč, hodnota. Tieto vektory sú vytvorené pomocou násobenia embeddingu a troch matíc $W_{\text{query}}, W_{\text{key}}, W_{\text{value}}$.

Nech existujú tokeny $x_1, x_2, \ldots, x_1$ chce poznať svoju hodnotu v porovnaní s tokenom x_2. x_1 sa *dotazuje* x_2, x_2 mu poskytne odpoveď vo forme svojho vlastného *kľúča*, ktorý sa použije na zistenie hodnoty, pomocou ktorej je možné reprezentovať, ako moc si váži x_1 pomocou skalárneho súčinu s *dotazom*. Výsledok bude jedno číslo, pretože oba vektory majú rovnakú veľkosť. Táto operácia sa prevedie s každým slovom. Tieto hodnoty predstavujú nenormalizované skóre.

x_1 vypočíta všetky *skóre*, podelí ich odmocninou dĺžky vektora a prevedie soft–max, aby bola hodnota ohraničená a relatívny rozdiel zachovaný. Toto skóre pozornosti sa vypočíta pre každý token vo vstupnej sekvencii. x_1 využije toto *skóre* a *hodnotu*, aby získalo novú hodnotu samého seba. Ak slovo nie je relevantné voči x_1 skóre bude malé a zodpovedajúca hodnota bude znížená faktorom tohto skóre, ak je slovo dôležité, tak táto hodnota bude posilnená.

Nová *hodnota* pre x_1 bude výsledkom sčítania všetkých prijatých hodnôt, pričom táto *hodnota* predstavuje nový embedding. Architektúra transformerov používa komplexnejšiu *Multi-Head attention* vrstvu. Kroky pozornosti na výpočet *dotazu*, *kľúča* a *hodnoty* sú aplikované na jednotlivé tokeny $x_1, x_2, \ldots$ separátne a nové embeddingy v_1, v_2 sú vytvorené pre každý set. Tieto embedingy sú konkatenované a vynásobené pomocou matice Z, ktorá je trénovaná spoločne a redukuje tieto embeddingy na jeden embedding pre $x_1^{'}, x_2^{'}, \ldots$ Každý v_i reprezentuje hlavu modelu pozornosti.

Intuitívne každá z hláv pozerá na pôvodný embedding z rozdielneho kontextu, pretože každá z matíc Q, K, V je na začiatku inicializovaná náhodne a následne modifikovaná počas tréningu. Embedding sa naučí brať do úvahy rôzne kontexty súčasne.

Normalizácia vrstvy normalizuje vstupy naprieč všetkými prvkami. Normalizácia sa vykonáva so zvyškami, čo umožňuje si ponechať určitú informáciu z predchádzajúcej vrstvy. Kodér aj dekodér obsahujú blok neurónovej siete s posuvom vpred, ktorý pozostáva z dvoch lineárnych vrstiev s využitím relu aktivačnej funkcie. Vstupom sú embeddingy $x_1^{'}, x_2^{'}, \ldots$ a výstupom $x_1^{''}, x_2^{''}, \ldots$ rovnakých rozmerov, mapovaných do latentného priestoru spoločného pre celý jazyk.

Dekodér

Pri kódovaní slova je nutné vedieť celý jeho kontext vo vete, preto vo vrstve pozornosti je vykonaný *dotaz* so slovami voči všetkým slovám. Dekódovanie sa snaží predpokladať ďalšie slovo vo vete, logicky by nemalo vedieť ktoré slová ho nasledujú. Embedding týchto slov je maskovaný pomocou násobenia s nulou. Predpokladanie slova je teda založená na predchádzajúcich slovách.

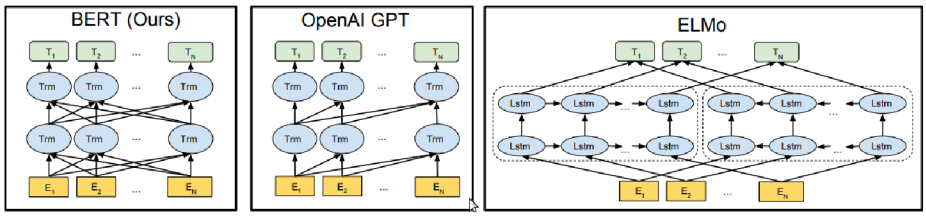
Skutočnosť, že výstupné embeddingy sú posunuté o jednu polohu spolu s maskovaním zaisťuje, že predpovede závisia iba na známych výstupoch v pozíciách menších ako i [39].

Vo vrstve *Multi–Head attention* sú vektory *hodnôt* (V) a *kľúčov* (K) prijaté ako výstup kodéru, vektory *dotazov* ako výstup z predchádzajúcej vrstvy dekodéru. Dekodér sa *dotazuje* všetkých slov so zakódovaným embeddingom z pôvodnej sekvencie, ktoré nesú pozičnú a kontextovú informáciu. Nové embeddingy dostanú tieto informácie, na základe ktorých je založený predpoklad výstupnej sekvencie.

Výstup kodéru je vylepšenou verziou vstupných embeddingov. Dodatočné vylepšenie pozostáva z využitia viacerých kodérov/dekodérov (stohu). V publikácii [39] využili model so štyrmi vrstvami kodér-dekodér, pričom výstup posledného kodéru sa privádza na vstup dekodérom.

Bert

Pojem *Bert* [9] znamená ***B**idirectional **e**ncoder **r**epresentations from **t**ransformers*. Bert je zástupca z rodiny transformerov 2.8. Porovnanie architektúr založených na pre-trénovacích modelov je možné vidieť na obrázku 2.8.



Obr. 2.8: Rozdiely pre-trénovacích architektúr Bert, GPT, ELMo [9].

Bert využíva obojsmerný transformer, GPT[8] využíva transformer z ľava–do–prava a ELMo z ľava–do–prava a z prava–do–ľava LSTM na generovanie príznakov.

Dôležitosť každej vrstvy je viazaná ku konkrétnej úlohe. Úlohy zakladajúce na syntaktických informáciách sú zachytené v nižších vrstvách a sémantické vo vyšších, obdobne ako v prípade modelu ELMo 2.7.

Vstupný text je najskôr podrobený tokenizácii, využitím princípu kusov slov (*wordpieces*) namiesto slov, efektívne v redukovaní veľkosti slovníka. Každému tokenu je pridelený unikátny index do slovníka.

Každý token je mapovaný do tzv. embeddingu, vektora reálnych čísel istej dimenzie. Elementy tohto vektora sú brané ako parametre modelu a sú optimalizované behom učenia. Sekvencie slov sú zarovnané pomocou tokenov *<pad>*. V prípade Berta je vybraných 12 alebo 24 blokov kodéru.

Pre–trénovanie Berta

Pri trénovaní modelu bolo náhodne maskovaných 15% slov v každej sekvencii. Maskované slová neboli vždy maskované tokenom `[MASK]`, v 10% prípadoch boli nahradené náhodným slovom, v 10% prípadoch neboli nahradené vôbec. Každá sekvencia slov je doplnená o token:

- `[CLS]` – prvý token sekvencie, využíva sa v spojení so softmax vrstvou pri klasifikačných úlohách.
- `[SEP]` – token oddeľovania sekvencií (viet), používa sa v úlohách na predikciu ďalšej vety.

Jazykové modely nezachytávajú vzťah medzi po sebe nasledujúcimi vetami. Pre túto úlohu bol Bert naučený pomocou párov viet, pričom polovica trénovacích dát týchto párov pozostávala zo skutočne po sebe nasledujúce vety, zbytok boli náhodné vety z korpusu.

Trénovanie na špecifických korpusoch dáva lepšie výsledky v rámci domény využitia, známe sú napríklad BioBERT (medicínske texty), SciBERT (vedecké publikácie). Bert dosiahol *state of the art performance* v obdobných úlohách ako ELMo 2.7.

Sentence–Bert

Vo voľnom preložení *vetný Bert* je modifikácia pre-trénovanej siete BERT, ktorá využíva siamské siete a štruktúry vetných trojíc na odvodenie sémanticky významových vetných vektorových reprezentácii, ktoré môžu byť porovnané pomocou kosínusovej podobnosti [31].

Táto metóda ponúka využitie modelu Bert pre nové úlohy ako rozsiahle porovnávanie sémantickej podobnosti, zhlukovanie a získavanie informácii pomocou sémantického vyhľadávania. Bežnou metódou zhlukovania a sémantického vyhľadávania je mapovanie každej

[8] https://openai.com/blog/better-language-models/

vety do vektorového priestora tak, aby boli sémanticky podobné vety blízko seba. Napríklad pomocou spriemerovania vektorových reprezentácii tokenov vety, pričom tento spôsob nie je až tak vhodný.

S–Bert pridáva do výstupu Berta poolingovú operáciu na odvodenie vetných vektorových reprezentácii, stratégiami spriemerovania, využitia CLS tokenu, maximálnu hodnotu. Cieľom ladenia Berta je vytvorenie siamské siete a siete trojíc, pričom výsledné vektorové reprezentácie môžu byť porovnané pomocou kosínusovej podobnosti.

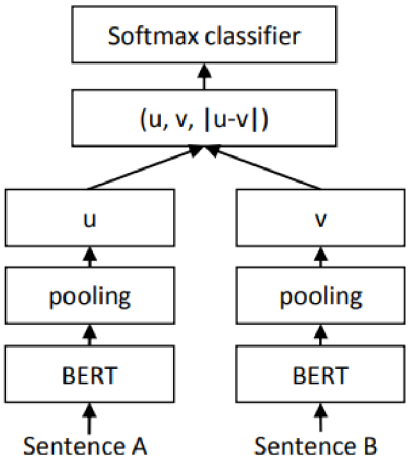
V publikácii [31] sú popísané tri experimenty so štruktúrou siete a to klasifikačná funkcia, regresná funkcia, funkcia trojice.

Klasifikačná funkcia

Konkatenáciou vektorových reprezentácii viet u a v s elementárnym rozdielom $|u-v|$ a násobením s trénovateľnými váhami $W_t \in R^{3n \times k}$:

$$o = \text{softmax}(W_t(u, v, |u - v|)) \tag{2.16}$$

kde n je dimenzia vetných vektorových reprezentácii a k je počet štítkov (labels). Optimalizuje sa strata cross-entropy. Štruktúra je zobrazená na obrázku 2.9.



Obr. 2.9: Architektúra SBERT s klasifikačnou úlohou [31].

Regresná funkcia

Využitím kosínusovej podobnosti medzi dvoma vetnými vektorovými reprezentáciami u a v. Využitie funkcie priemernej kvadratickej chyby.

Funkcia trojice

Nech je daná kotviaca veta a, pozitívna veta p a negatívna veta n, strata tripletov vyladí sieť tak, aby vzdialenosť medzi a a p bola menšia ako vzdialenosť medzi a a n. Matematicky môžeme minimalizovať stratovú funkciu:

$$\max(\|s_a - s_p\| - \|s_a - s_n\| + \epsilon, 0) \tag{2.17}$$

kde s_x je vetná vektorová reprezentácia pre $a/n/p$, $\|\cdot\|$ je metrika vzdialenosti a okolie ϵ zaisťuje, že vzdialenosť s_p je aspoň o ϵ bližšia k s_a ako s_n.

Kapitola 3

Návrh riešenia

Táto kapitola popisuje návrh riešenia aplikácie pre sumarizovanie postojov z recenzií výrobkov. Celá aplikácia sa skladá z troch častí: systému analýzy recenzií, indexačného systému a grafického užívateľského rozhrania.

Aplikácia si spravuje svoj dataset recenzií pomocou indexačného systému, ktorý dokáže pravidelne aktualizovať, pomocou prechádzania stránok porovnávačov tovaru. Výsledný systém primárne pracuje s recenziami v českom jazyku. Pomocou automatických aktualizácii sa snaží vytvoriť jeden z najväčších datasetov recenzií pre český jazyk. S využitím tohto datasetu je možné vykonávať rôzne klasifikačné a predikčné úlohy pomocou natrénovaných modelov systému analýzy recenzií, ktoré sú uvedené v rámci návrhu riešenia.

Aplikácia sa snaží zoradiť vety recenzií do zhlukov podobných viet, pričom v tejto kapitole sú uvedené isté metódy extrakcie aspektov a následne načrtnuté výsledné riešenie.

Grafické užívateľské rozhranie zobrazuje jednak stiahnuté recenzie, výsledky klasifikácie dát, vykonané experimenty a ponúka funkcie na export dát v určitých formátoch.

3.1 Motivácia

Analýza recenzií výrobkov je široký pojem. Recenzie uvádzajú istý postoj recenzenta ku kvalite výrobku/služby a jeho skúsenosti. Obsahujú potrebné informácie spojené s výrobkom, na základe ktorých sa ostatní ľudia rozhodujú o kúpe daného tovaru. Rozhodovanie je podmienené prečítaním recenzií a vytvorením si vlastného názoru. Idea systému spočíva v urýchlení tohto procesu, kedy užívateľ systému nebude musieť manuálne prejsť všetky recenzie produktu/kategórie, čo je zdĺhavý proces. Je nutné pripomenúť, že človek neprečíta všetky recenzie a pozrie si len „pár“ prvých recenzií, na základe ktorých si vytvorí svoj názor na produkt. Nie všetky recenzie sú relevantné a vyjadrujú sa k daným aspektom produktu. Nemajú správne percentuálne hodnotenie alebo text v nich odpovedá polarite hodnotenia.

Systém pracuje primárne s porovnávačom tovaru *heureka.cz*, ktorý agreguje recenzie z jednotlivých obchodov, čo podstatne uľahčuje prácu v kroku extrakcie dat. Extrakcia dát pozostáva z pravidelného prehľadávania internetového portálu s cieľom vyhľadania recenzií produktov a ich následnou indexáciou. Recenzie na tomto portály sú písané prevažne v českom jazyku. Pre český jazyk sa jedná o vytvorenie jedného z najväčších datasetov recenzií vôbec.

Nad týmito recenziami sa vykonajú analýzy zamerané na sumarizáciu obsahu recenzií pre užívateľa s náhľadom jednotlivých viet k daným oblastiam, predpokladu skutočného hodnotenia recenzie, kontrolu polarity textu a vyznačenie kľúčových slov.

Analýzy založené na strojovom učení s učiteľom závisia na kvalite datasetu, na základe ktorého sa daný algoritmus učí klasifikovať množinu vlastností, anglicky *features*, do tried alebo predpokladať číselné ohodnotenie. Tieto analýzy sa dajú vykonávať pomocou manuálne anotovaných datasetov. Čiastočnou anotáciou dát heuréky sa zmenšuje nutnosť vytvorenia špeciálnych datasetov pre jednotlivé analýzy.

Systém cieli na prevádzanie analýz s minimálnym využitím manuálnej práce pri anotovaní dát.

Datové sety

Portál *heureka.cz* obsahuje čiastočne anotované dáta. Každá recenzia obsahuje kolónky pro, proti, zhrnutie, percentuálne hodnotenie. Tieto kolónky sú podstatné pre jednotlivé analýzy. Pre bežného užívateľa spočíva elementárne využitie systému v prezeraní dát a výsledkov jednotlivých analýz.

V prípade, že užívateľ je znalý v oblasti strojového učenia, vyskytuje sa možnosť extrakcie dát pre iné experimenty. Systém bude ponúkať rozhranie pre extrakciu recenzií pre rôzne modely strojového učenia, respektíve formáty. Prirodzene, vďaka čiastočnej anotácii sa môže jednať o bipolárnu analýzu sentimentu, predpoveď hodnotenia textu alebo vygenerovanie datasetu vo forme čistého textu.

Systém podporuje export celého indexovaného datasetu, kvôli lepšej manipulácii s dátami. Export je spojený aj s rozšíriteľnosťou systému a jeho aspoň čiastočnej funkcionality.

Predpoveď percentuálneho ohodnotenia

Recenzent má možnosť ohodnotiť celkovo svoju recenziu, buď pomocou percentuálneho hodnotenia (100%) alebo číslom, napríklad 1/5. Jednou z analýz systému bude aj percentuálne ohodnotenie recenzie a jeho porovnanie s autorovým ohodnotením. Portál *heureka.cz* používa percentuálne ohodnotenie. Z tohto ohodnotenia je možné vyvodiť istú mieru správnosti recenzie, ako moc sa líši autorove ohodnotenie od natrénovaného modelu.

Bipolárna analýza postojov

Na základe čiastočne anotovaných dát, v podobe položiek pro a proti je možné natrénovať model na bipolárnu analýzu sentimentu na úrovni vety. S využitím tohto modelu je možné kontrolovať jednak správnosť položiek pro a proti, jedná sa o prostú kontrolu recenzentovej vôle. Tento model je možné využiť aj na klasifikáciu polarity sumarizujúceho textu, v ktorom je možné vidieť jeho polarity.

Irelevantné recenzie

V prípade datasetu recenzií sa predpokladá výskyt irelevantných recenzií, na ktoré systém upozorní, prípadne ich rovno odfiltruje pri aktualizácii. Užívateľa nezaujímajú recenzie, ktorých vyjadrený postoj nesúvisí s produktom alebo sa výrazne líši a nekoreluje s hodnotením. Jedná sa prevažne o recenzie typu „Zatím nic.“ alebo *Daný výrobek máme krátce.*.

Sumarizácia recenzií

Navrhovaný systém ponúka jednak zobrazovanie recenzií produktov rôznych kategórii zrovnávačov tovaru, spolu s jednoduchými štatistickými informáciami ako bežné zrovnávače. Portál *heureka.cz*, síce obsahuje isté zhrnutie recenzií, no jedná sa o prosté štatistické informácie najčastejšie vyskytujúcich slov v recenziách a priemerné hodnotenie produktu na základe recenzií.

Aj keď u niektorých produktov ponúka priamo zobrazenie jednotlivých aspektov a ich polarít k recenziám no nie u všetkých produktov. Navrhovaný systém cieli na vykonanie tejto analýzy čo najviac automatizovane, bez využitia trénovacieho datasetu, ale na základe podobnosti jednotlivých viet recenzií. Tento systém bude adaptovateľný na rôzne domény produktov bez potreby vytvorenia špeciálneho datasetu na klasifikáciu viet do tried aspektov. Užívateľ bude môcť vykonávať analýzy ad-hoc nad vybranou kategóriou výrobkov. Systém prevedie vety do vektorovej reprezentácie pomocou vybraného algoritmu a rozdelí ich do zhlukov na základe vybranej zhlukovacej metódy. Pomocou sumarizačného algoritmu bude užívateľ vedieť charakteristické slová tém zo zhlukov textu a bude si môcť vytvoriť obraz o jednotlivých názoroch, prípadne prezrieť jednotlivé vety.

Slová charakterizujúce témy vo zhlukoch je možné brať ako kľúčové slová charakterizujúce aspekty. Túto analýzu je možné využiť aj v náhľade recenzií so zvýraznením potenciálneho aspektu vo vete sumarizujúceho textu.

Architektúra

Samotný dataset recenzii bude indexovaný v systéme podobného databáze, ale optimalizovanom na vyhľadávanie dokumentov, ktorý bude jednoducho rozšíriteľný. Grafické prostredie bude pozostávať z webovej stránky, hlavne kvôli dostupnosti systému.

Zovšeobecnenie systému

Práca je zameraná na prácu s portálom na porovnávanie tovaru *heureka.cz*. Na jej základe sa vytvorí architektúra pavúka, ktorý môže byť upravený na využitie pre iné internetové obchody a portály.

Výsledný systém môže slúžiť ako podklad na vytvorenie všeobecného systému pre indexáciu dát recenzií. Pri rozšírení systému na nový obchod, ako je napríklad spomínaná *alza*, je potrebné upraviť slovník vstupných url adries domén produktov a následne aj pravidiel prechádzania internetového obsahu.

Funkcionalita systému, ktorá je uvedená nižšie, sa pri obdobnej charakteristike recenzie nezmení. Nutné bude doplnenie indexov na indexáciu recenzií.

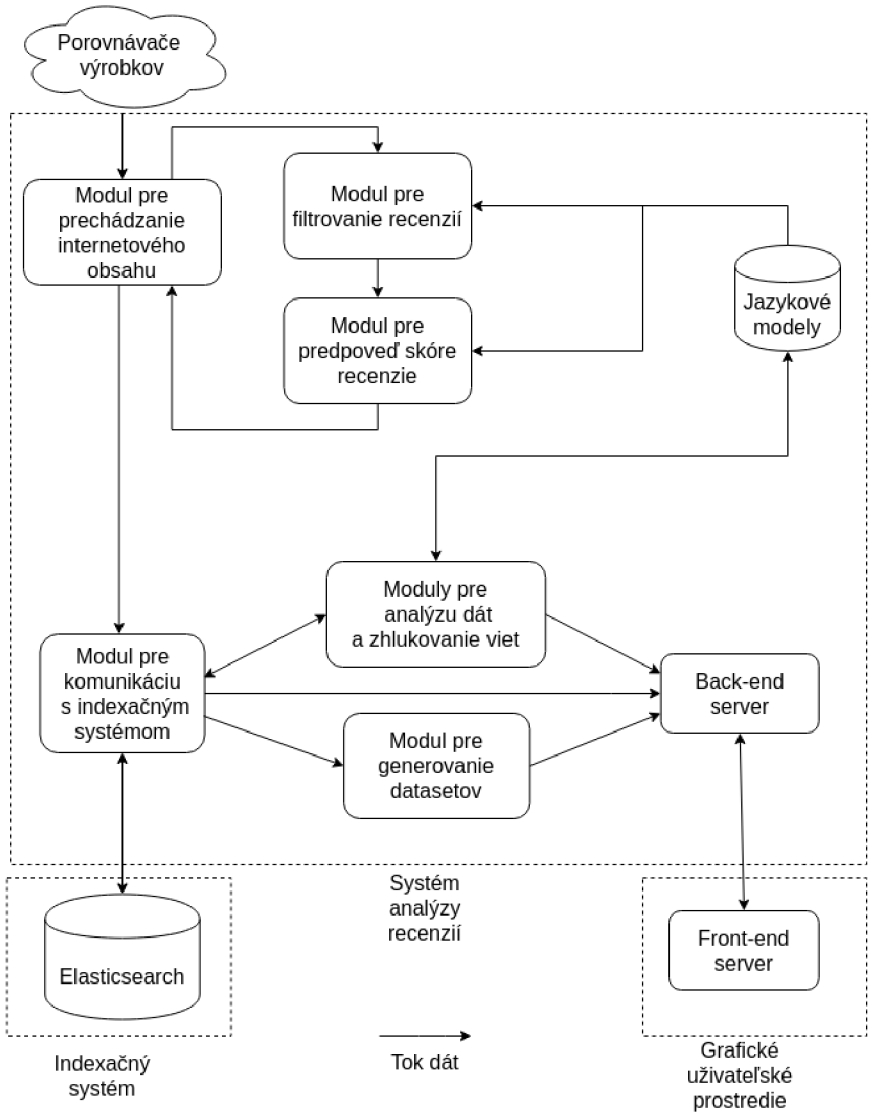
Systém pracuje s recenziami napísanými v českom jazyku, pretože portál *heureka.cz* združuje obchody pôsobiace v Českej republike. Obdobne existuje aj slovenská odnož heuréky. Aktuálne sú slovenské recenzie (uvedené na portály *heureka.cz*) spracovávané modelmi, natrénovanými nad českými textami, prípadne viacjazyčnými modelmi.

V prípade prenosu systému na slovenské recenzie sa za ideálne riešenie pokladá natrénovanie klasifikátorov nad slovenskými textami. Na druhej strane sú slovenčina a čeština veľmi blízke jazyky.

3.2 Schéma systému

Schéma navrhnutého systému je zobrazená na obrázku 3.1, zobrazuje základné moduly, potrebné pre fungovanie systému ako celku. Tok dát je reprezentovaný pomocou šípok.

Navrhnutá aplikácia potrebuje pre svoju plnú funkcionalitu súbežný beh týchto troch služieb: elasticsearch, back–end server, front–end server. Tieto tri služby môžu byť spustené na rôznych systémoch.



Obr. 3.1: Navrhovaná schéma výslednej aplikácie.

Modul pre prechádzanie internetového obsahu predstavuje základ systému analýzy recenzii. Prechádza recenzie vystavené na príslušnom porovnávači tovaru, ktorému odosiela požiadavky na zobrazenie URL stránky a porovnávač výrobkov následne vracia html stránku. Tento modul komunikuje s modulom zabezpečujúcim indexáciu dát. Ukladajú sa recenzie, produkty a obchody. Dáta (recenzie, produkty, obchody), ktoré sa už nachádzajú v systéme sa následne nebudú indexovať. Využíva modul pre filtrovanie recenzií na odhalenie irelevantných viet a modul na predpoveď skóre recenzie.

Pred indexovaním recenzie a jej položiek, ako sú napríklad plusy alebo mínusy, prebehne overenie pomocou modelu pre filtrovanie recenzií na výskyt irelevantných viet a slovných spojení typu *„Zatím nic.“* alebo *„Produkt máme krátce.“*. Tento modul využíva jazykový model, ktorý je ladený pre klasifikáciu viet na dve triedy relevantné a irelevantné vety. Irelevantné vety sa odstránia zo stiahnutej recenzie.

Následne prebehne analýza recenzie na predpoveď skóre recenzie, na odhalenie potenciálne irelevantných recenzií. Ak bude delta rozdielu medzi ohodnotením pomocou modelu a autora väčšia ako istá konštanta, tak to systém zobrazí v GUI.

Vytváranie dotazov pre konkrétny systém zabezpečuje modul pre komunikáciu s indexačným systémom (elasticsearch). Modul ponúka isté API CRUD dotazov k jednotlivým indexom. Posiela požiadavky idexačnému systému vo forme DSL dotazov.

Jadro práce zabezpečujú moduly pre analýzu dát a zhlukovanie viet recenzií. Zabezpečujú analýzu bipolárneho sentimentu, predpoveď skóre hodnotenia nad recenziami s využitím natrénovaného jazykového modelu, klasifikátor relevantnosti viet. Ponúkajú rozhranie pre zhlukovanie viet recenzií pomocou rôznych mapovaní viet do vektorového priestoru s využitím modelovania tém. Výsledky analýz sa ukladajú do indexačného systému.

Datasety je možné vytvárať pomocou modulu pre generovanie datasetov, ktorý komunikuje s modulom komunikujúcim s indexačným systémom a dotazuje sa ho na dáta, na základe ktorých môže vygenerovať príslušný dataset pre danú úlohu.

Koncový server (back–end) ponúka rozhranie takzvaných koncových bodov pre požiadavky od klienta. Komunikuje priamo s modulmi zabezpečujúcimi analýzu dát, zhlukovanie viet, generovanie datasetov a prístup k indexovaným recenziám.

Užívateľské rozhranie, takzvaný „front–end“, zobrazuje dáta systému. Toto rozhranie zobrazuje užívateľovi jednak samotné recenzie produktov a obchodov, výsledky prevádzaných analýz nad recenziami výrobkov, obchodov, štatistiky aktualizácii. Ponúka rozhranie s dialógom pre zhlukovanie viet na základe podobnosti, kde systém na základe užívateľovho vstupu zhlukne vety kategórie/produktu a ponúkne užívateľovi dialóg na ich prípadnú úpravu a doplnenie dodatočného významu. Súčasťou je aj rozhranie pre generovanie datasetov pre definované klasifikačné úlohy, ako napríklad bipolárna klasifikácia sentimentu.

3.3 Prechádzanie internetového obsahu

Na vykonávanie klasifikačných a regresných úloh sú potrebné dáta, respektíve dataset. Jeho získanie je prvý krok návrhu takéhoto systému. Táto práca sa zaoberá analýzou recenzií výrobkov. Takýchto recenzií existuje nespočetne veľa. Každý internetový obchod si už integroval vlastnú reprezentáciu publikovania recenzií ku vystaveným výrobkom. V dnešnej dobe prebieha rozmach internetového nákupu výrobkov, či už elektronika alebo oblečenie.

Naivný spôsob získania dát je prechádzanie týchto obchodov pomocou *URL* obchodov. Tu nastáva problém reprezentácie dát. Každý obchod používa iný formát zobrazenia dát a tým pádom je nutné implementovať algoritmus prechádzania recenzií pre každý obchod samostatne. Lepším riešením je využitie existujúcich riešení typu zrovnávačov výrobkov. Pre Českú republiku predstavuje takýto zrovnávač portál *heureka*[1].

Dataset Heuréka

Pojem *heureka* pochádza z gréčtiny a znamená *našiel som to*. Zrovnávač výrobkov bol založený v roku 2007 so sídlom v Liberci. Portál slúži ako porovnávač výrobkov rôznych

[1] https://www.heureka.cz/

obchodov, teda agreguje výrobky z rôznych obchodov, ponúka krátku špecifikáciu ku každému výrobku a následne aj zoznam obchodov, kde sa daný výrobok dá kúpiť za akú cenu. Agreguje recenzie z ostatných obchodov. Táto informácia je pre prácu kľúčová, pretože už nie je nutné prechádzanie ostatných obchodov.

V závislosti od počtu, kvality recenzií sa môžu pridať aj ostatné väčšie obchody, ako napríklad alza[2], ktorej výrobky nie sú agregované heurékou kvôli istému sporu pár rokov naspäť. Podľa štatistík, heuréka ponúka viac než 21 miliónov produktov, čo tvorí základný predpoklad, že veľkosť datasetu bude postačujúca.

Pri bližšej analýze, heuréka využíva podobný formát pre zobrazovanie jednotlivých kategórií naprieč doménami výrobkov, teda formát dát zobrazovaných na heuréke bude konzistentný. Za dôležité informácie sú považované jednak recenzie, samotné produkty a do úvahy spadajú aj recenzie obchodov.

Prvým krokom bude samotná indexácia heuréky cez jej domény. Heuréka obsahuje 14 hlavných doménových kategórií, od elektroniky, cez oblečenie až po kozmetiku. V rámci každej tejto domény je potrebné naindexovať najskôr všetky produkty heuréky. Tento krok bude náročný pri predpoklade existencie 21 miliónov produktov, ale na druhej strane zaujímavé sú len produkty s recenziami. Druhým krokom môže byť prechádzanie cez jednotlivé URL adresy výrobkov a sťahovanie recenzií a ich následné spracovanie do internej podoby. Tento krok môže byť chápaný zo strany heuréky ako pokus o jemný *DOS* útok, berie sa do úvahy, že takýto veľký portál bude mať implementované prostriedky proti takýmto pokusom, teda je možné čakať vyššiu odozvu na požiadavku (*GET request*).

Metadáta

Text recenzie nie je jedinou zaujímavou informáciou vzťahujúcou sa k práci. Príklad takejto recenzie produktu z heuréky je možné vidieť na obrázku 3.2. V rámci recenzie je možné vidieť aj autorom uvedený čiastočný sentiment v podobe plusov, respektíve mínusov. Tento fakt je kľúčový pre generovanie datasetu pre bipolárnu klasifikáciu sentimentu. Ďalej je možné vidieť celkové percentuálne hodnotenie výrobku, v niektorých prípadoch aj užívateľské meno autora príspevku, obchod, v ktorom bol daný výrobok zakúpený, dátum pridania recenzie, informáciu o doporučení výrobku uvedenú v pravo hore a spätnú väzby ostatných užívateľov, v pravo dole a nakoniec celkové zhodnotenie recenzie.

Tieto údaje sú vytvorené priamo užívateľmi, preto má recenzia hodnotenie 90%, ale napriek tomu obsahuje dve negatívne sekcie.

Tieto informácie je nutné extrahovať z URL a premeniť ich do vnútornej reprezentácie objektu recenzie. Každá recenzia je vytvorená autorom, ktorý v nej publikuje svoj subjektívny názor na kvalitu a rôzne vlastnosti produktu. Tento fakt je potrebné v neskoršej fáze zohľadniť.

Okrem recenzií výrobkov, by mohol systém obstarávať aj dataset recenzií obchodov, ktorých recenzie a výrobky agreguje heuréka. Jednalo by sa o ďalšiu doménu recenzií.

Predspracovanie dát

V rámci extrakcie dát z heuréky, hlavne recenzií, je vhodné tieto dáta rovno predspracovať. Jedná sa hlavne o tokenizáciu a POS tagovanie. Síce aktuálne riešenia mapovania sekvencií do vektorového priestoru využívajú vlastné tokenizery, ako napríklad kúsky slov, je vhodné

[2] https://www.alza.cz/

Obr. 3.2: Príklad recenzie z heuréky.

tieto dáta tokenizovať, lemantizovať, poprípade previesť do kmeňového tvaru (stem), odstrániť stop slová, pre ďalšie spracovanie.

3.4 Indexovanie dát

Keďže dataset heuréky obsahuje, podľa tvrdení heuréky, 21 miliónov produktov je potrebné využitie indexačného nástroja, ktorý by dáta naindexoval a súčasne mohol vytvárať dotazy podobné jazyku SQL na získavanie recenzií produktov istej domény heuréky. V rámci teoretického rozboru bol zmienený systém *Elasticsearch* 2.2, ktorý sa javí ako najlepšie riešenie.

Systém bude musieť získavať dáta pre generovanie datasetov, vizuálizáciu, pričom bude potrebné, aby čas získania dát bol čo najmenší. Predpokladá sa využitie SQL dotazov na získavanie množiny recenzií na základe klasifikačnej úlohy, ktorú si užívateľ vyberie. Elasticsearch ponúka takéto API, čiže bude horúcim kandidátom.

3.5 Analýza recenzií výrobkov

Táto práca sa viaže na analýzu recenzií výrobkov. U recenzií sa sleduje hlavne vyjadrený sentiment recenzenta k výrobku. Sentiment je možné sledovať na troch úrovniach: *dokumentu*, *vety*, *aspektu*.

Úroveň dokumentu je chápaná, ako postoj recenzenta na celú recenziu, príspevok. Medzi detailnejšiu analýzu patrí analýza sentimentu na úrovni vety. V tomto type analýzy nastáva problém, keď recenzent vyjadrí názor k viacerým vlastnostiam (aspektom) produktu. Najdôležitejšie analýza je zistenie sentimentu na úrovni aspektu.

Aspekt je vlastnosť výrobku, ku ktorej recenzent vyjadruje svoj postoj. Výsledkom tejto analýzy bude zoznam jednotlivých aspektov nájdených v recenzii a následne autorov sentiment, ktorý môže spadať do troch kategórií – *kladný*, *záporný*, *neutrálny*. Analýza sentimentu je kľúčovou úlohou tejto práce, pričom nie je jedinou.

Klasifikácia relevantnosti recenzie

Každá recenzia obsahuje položky, ku ktorým sa autor vyjadril pozitívne a položky, ku ktorým sa autor vyjadril negatívne. Predpokladá sa korelácia medzi počtom týchto položiek a výsledným hodnotením. Dataset by nemal obsahovať nesprávne recenzie, respektíve recenzie, ktoré majú celkové hodnotenie 100% a obsahujú samé negatívne položky. Na takéto recenzie by mal systém upozorniť pri ich sťahovaní.

Táto úloha môže byť formulovaná pomocou regresie. Systém bude obsahovať regresívny model, ktorý na základe textu recenzie bude predpokladať percentuálne ohodnotenie sentimentu a v prípade, ak sa bude líšiť väčšou hodnotou ako určené δ, tak upozorní užívateľa na danú/dané recenzie. Tým pádom bude dataset obsahovať len „správne“ dáta. Týmto princípom sa môžu vyselektovať recenzie, ktorých skóre nezodpovedá uvedenému sentimentu alebo by sa skóre mohlo upraviť.

Prevažne v negatívnych sekciách recenzií (proti) sa vyskytujú aj slovné spojenia typu *„Nic.“*, *„Zatím nic.“*, *„Výrobek máme krátce.“*. Tieto slovné spojenia nevykazujú polaritu sentimentu, ich výskyt v sekciách pre/proti nemá význam, teda sa jedná o irelevantné vety v týchto užívateľmi anotovaných sekciách. Predpokladá sa natrénovanie klasifikátora na predpoveď relevantnosti textu na úrovni sekcie pro/proti. Dataset na trénovanie klasifikátora bude pripravený manuálne anotovaním viet recenzií na relevantné a irelevantné.

Klasifikácia sentimentu recenzií

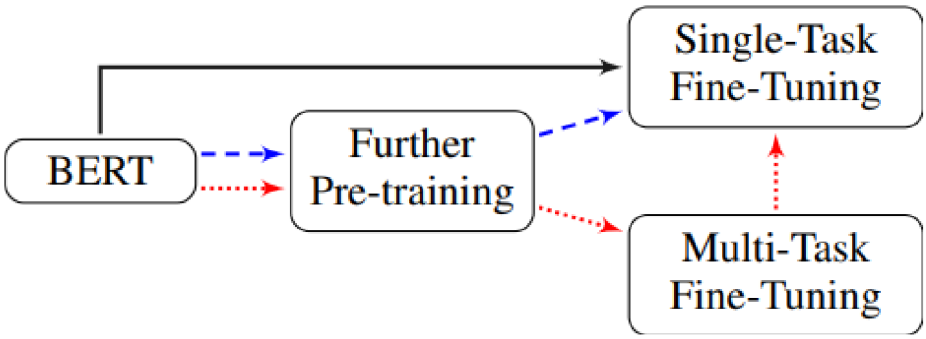
Pri procese klasifikácie sentimentu sa predpokladá porovnanie doterajších klasifikačných algoritmov na vygenerovanom datasete. Doterajšie riešenia 2.4 spočívali vo využití SVM, Naive Bayes a pod. Aktuálne používané modely využívajú technológiu neurónových sietí, ako modely využívajúce siete LSTM alebo mechanizmus pozornosti (transformery, napríklad model Bert 2.8).

Modely pozornosti

Model Bert je zástupca transformerov. Jeho voľba je vhodná, hlavne kvôli dostupnosti pred-trénovaných viacjazyčných modelov, pretože natrénovať model nad datasetom recenzií bude výpočetne náročné. Lepším riešením je využitie volne dostupných pred trénovaných modelov, pričom pre český jazyk je možné využiť dostupný viacjazyčný model.

Využitie Berta [37] spočíva v jeho *ladení* nad danou klasifikačnou úlohou, ako klasifikácia správnosti recenzie, klasifikácia sentimentu v rámci istej domény. Ladenie Berta je možné vykonať v troch spôsoboch, zobrazených na obrázku 3.3.

- Ladiace stratégie – ladenie modelu Bert na základe konkrétnej úlohy, rôzne vrstvy zohľadňujú odlišné vlastnosti jazyka čo sa týka sémantiky a syntaxe.
- Pretrénovanie – model Bert je pretrénovaný na všeobecnej doméne, ktorá má odlišnú distribúciu dát, ako sú napríklad recenzie výrobkov. Tento krok je ale výpočetne náročný a pretrénované modely od spoločnosti Google sú verejne dostupné.
- Viac-úlohové ladenie – toto učenie ukázalo svoju efektivitu využívania zdielaných znalostí medzi viacerými úlohami. Ak je v cieľovej doméne niekoľko dostupných úloh, kladie sa otázka využitia, či je ešte výhodné doladiť model Bert na všetky úlohy súčasne.



Obr. 3.3: Ladenie modelu Bert [37].

Analýza sentimentu na úrovni aspektu a Bert

Analýza sentimentu na úrovni aspektu bola prvý raz uvedená na *SemEval–2014*, pri ktorej bol poskytnutý dataset anotovaných recenzií reštaurácii a notebookov, *SemEval–2015* obsahoval celé recenzie, následne sa dataset už moc nemenil.

Úlohy pozostávali z klasifikácie kategórie aspektu, ktorá cieli na identifikáciu páru témy a aspektu vyjadrených v texte. Cieľové vyjadrenie názorov je úloha zaoberajúca sa extrakciou lingvistického výrazu použitého vo vete a klasifikácie polarity sentimentu pre každý identifikovaný pár téma a aspekt.

Podľa publikácie [13] boli najlepšie hodnotené techniky založené na *SVM*. S nástupom transformerov je možné túto úlohu formulovať, ako úlohu klasifikácie dvojíc viet využitím modelu Bert, vytvorením pomocnej vety sa výsledky zlepšujú porovnaním s predchádzajúcimi metódami.

Navrhnutý model klasifikácia sentimentu textu na základe aspektu je možné implementovať pomocou úlohy klasifikácie dvojíc viet [13] ako:

- veta A: text, ktorý bude vyhodnotený
- veta B: aspekt
- značka: vyjadrujúca sentiment

Obdobne je možné modelovať úlohu relatívnosti aspektu k danej vete. Tieto dva modely je možné kombinovať do jedného, ktorý na základe vstupného textu a aspektu bude predikovať sentiment, ak daný aspekt je viazaný k textu. V opačnom prípade bude výstup charakterizovať nesúvisiacu značku.

Predpoveď skóre sentimentu

Každá recenzia obsahuje položku *rating*, teda skóre sentimentu uvedeného autorom. Predpokladá sa, že takéto skóre hodnotenia recenzie je v istej korelácii s uvedeným postojom, ktorý je popísaný v texte recenzie. Predikcia skóre hodnotenia recenzie je zaujímavou úlohou. Jedná sa o regresnú úlohu. Tento model by mohol byť využitý vrámci kontroly správnosti recenzie pri aktualizovaní celého datasetu recenzií alebo pri analýze recenzií.

Analýza trendov

Sumarizovanie informácii z recenzií je ďalšia časť analýzy. Vyhľadanie tém, o ktorých recenzenti rozprávajú v rámci kategórie produktu alebo obchodu prinesie dodatočné informácie

pre koncového užívateľa. Dokument, recenziu je možné chápať ako súbor tém, ktoré sú v určitom pomere. Tému je podobne možné chápať ako kolekciu kľúčových slov, tiež v istom pomere. Na modelovanie tém je vhodné využiť model strojového učenia bez dohľadu učiteľa, ako napríklad LDA 2.5.

Štatistické informácie o recenziách tiež do istej miery udávajú trendy zákazníkov pri nákupe tovaru. Síce systém necieli na získavanie informácii o dobe kúpi produktu, ale naopak o dobe pridania recenzie do systému. Na základe tejto informácie je možné sledovať výkyvy pri pribúdaní recenzií v priebehu roka, či už pri recenziách produktov, obchodov alebo pri pravidelných aktualizáciách.

Pri recenziách je možné sledovať aj takzvané položky „recommends“, jedná sa o odporúčanie autora pre produkt alebo celkovú spokojnosť recenzentov s produktom/obchodom.

Experimenty

Tiež je vhodné porovnať výsledky klasifikácie existujúcich riešení pre český jazyk. V rámci analýzy by to mohlo byť napríklad publikácia [44], ktorá využíva dataset publikovaný v práci [40].

Ďalej bude zaujímavé natrénovanie modelu klasifikácie sentimentu nad istou doménou, napríklad *elektronika*, a následné porovnanie klasifikátor nad bližšou doménou *biela technika*, prípadne *kozmetika*. Prípadne klasifikácie domény, ku ktorej sa recenzia viaže a podobne. V mnohých publikáciách sú vytrénované modely klasifikácie sentimentu, ktoré sú doménovo špecifické.

3.6 Extrakcia aspektov

V publikácii [8] sú prezentované kľúčové 4 kategórie možnosti extrakcie aspektov:

- Frekvenčne založené metódy.
- Metódy založené na vzťahoch.
- Metódy založené na strojovom učení.
- Modelovo založené metódy.

Frekvenčne založené metódy vychádzajú zo štatistík, aspekty predstavujú väčšinou podstatné mená alebo frázy, približne v 60% prípadoch. Využitie tejto metódy je spojené s POS značkovaním. Spomínané je aj využitie *Pointwise Mutual Information* skóre na vyhodnotenie asociácií medzi frázami.

Prístupy založené na vzťahoch využívajú syntaktické vzťahy medzi vetami na extrahovanie aspektov a sentimentu. V publikácii [8] sa spomínajú prístupy s využitím asociačných algoritmoch, z anotovaného datasetu algoritmus dokáže derivovať asociačné pravidlá s formátom $X \rightarrow Y$ s istou pravdepodobnosťou. Iný prístup môže pozostávať z detekcie viet vyjadrujúcich sentiment, takéto vety sú zacielené na nejaký aspekt produktu. S využitím parseru závislostí na identifikovanie a extrakciu aspektov a sentimentu.

Techniky strojového učenia sú často prispôsobené problému získavania informácií, ktorý sa nazýva pomenovanie identifikácie entity, anglicky *named–entity recognition* (NER). Problémom NER je odhaliť napríklad mená ľudí, miest, organizácií z textu využitím POS značiek.

Modelovanie tém, v kontextu strojového učenia, je naučenie sa abstraktných konceptov o dostupných témach z textových korpusov. Na detekciu aspektov sa môžu využiť modely, ako napríklad LDA 2.5.

Analýzou publikácii rôznych dostupných riešení vyplynulo, že prevláda postup využitia manuálne pripraveného datasetu, kde boli využití ľudskí pracovníci, aby manuálne označili jednotlivé aspekty vo vetách príslušnou značkou, a následne pomocou percentuálnej zhody medzi nimi pripravili označkovaný dataset dvojíc viet a kategórii aspektov. Tento postup je pracný a nemožno ho automatizovať.

Napríklad pre český jazyk je to publikácia [36], v ktorej český anotátori označovali 1500 viet, pričom zhoda bola okolo 85.5%. Dataset aspektov nie je voľne dostupný. Medzi dostupné datasety patrí napríklad dataset zo série *SemEval2016* pre český jazyk publikovaný v rámci práce [16]. Dataset obsahuje vety, u ktorých sú anotované jednotlivé aspekty a postoje recenzenta.

Riešenie extrakcie aspektov bez využitia manuálne anotovaného datasetu vyúsťuje do nasledujúcej sekcie 3.7, v ktorej navrhujem rôzne spôsoby zhlukovania viet na základe podobnosti na základe rôznych publikácii.

3.7 Zhlukovanie viet

Podobnosť viet charakterizuje ako „blízko“ sú dve časti textu v povrchovej (lexikálnej) blízkosti a významovej (sémantickej) podobnosti. Napríklad, nech existujú dve vety „Mačka zjedla myš.“ a „Myš zjedla mačkine jedlo.“ V rámci lexikálnej podobnosti na úrovni slova tieto vety obsahujú tri podobné slová, na druhej strane sa neberie kontext. V prípade kontextovej podobnosti je potrebné sa zamerať na sémantiku namiesto porovnávania slov a to na podobnosť častí textu. Aj keď sa slová môžu prekrývať, tieto dva bloky môžu mať v skutočnosti rôzny význam [34].

Ponúka sa myšlienka reprezentáciou textu, dokumentov, viet pomocou vektorovej reprezentácie a následné meranie vzdialenosti medzi týmito prvkami. Existuje mnoho spôsobov merania vzdialenosti, algoritmov využívajúcich len vektory viet alebo aj posilované učenie na základe predom anotovaného datasetu. Autor v svojom článku [34] uvádza radu metód, pri návrhu zhlukovania viet uvediem pár z nich.

Vektorové mapovanie a K–means

Naivné riešenie spočíva vo využití mapovanie textu do vektorovej reprezentácie pomocou pred trénovaných modelov Bert, FastText, Glove atď. na českých datasetoch, respektíve viac jazyčných datasetoch. Tieto jazykové modely generujú vektorové reprezentácie slov.

Každá veta, resp. blok textu potom obsahuje mnoho týchto vektorov. Vo fáze implementácie je potom nutné rozhodnúť aký výsledný vektor sa využije. Do úvahy pripadá spriemerovanie všetkých vektorov vety/bloku textu, maximum/minimum, špeciálne značky, ako v prípade modelu Bert.

Za zhlukovací algoritmus je možné zvoliť jednoduchý k-means s metrikou vzdialenosti kosínusovej podobnosti. Prípadné experimentovanie s ostatnými zhlukovacími algoritmami je tiež vhodné.

Vektorové mapovanie viet

Generovanie vektorových reprezentácii viet alebo slovných sekvencií môže byť vykonané napríklad natrénovaním modelu na inú úlohu, ako je napríklad predpoveď nasledujúcej vety. Na natrénovanie takéhoto modelu je potrebný manuálne anotovaný dataset.

Model Aurora a kol. [2] je inšpirovaný spriemerovaním slov po aktualizovaní vektorových reprezentácii slov ich ladením na parafrázovaných párach textu.

Autor v článku [11] popisuje rozšírenie modelu Aurora a kol. V pôvodnom modeli Aurora a kol sa slová generujú dynamicky náhodným chodením vektora diskurzu s časovým variantom, ktorý reprezentuje „o čom sa hovorí“. Pravdepodobnosť generovania slova w v čase t je daná logaritmickým produkčným modelom. Predpokladaním, že sa diskurzný vektor príliš nemení, model nahradí sekvenciu diskurzných vektorov jedným vektorom. Neskoršie úpravy modelu náhodného prechádzania dovoľujú, aby slová boli generované s istou pravdepodobnosťou.

Vektor vety je definovaný ako odhad MAP (maximálna aposteriórna pravdepodobnosť) vektora diskurzu, ktorý vygeneroval vetu. Model sa považuje ako model s istou formou učenia s učiteľom, pretože obsahuje hyperparameter, ktorý musí byť ladený na validačných dátach.

Autor [11] popisuje nový model na riešenie mätúceho efektu dĺžky slova modelu náhodnej chôdze, kde pravdepodobnosť pozorovania slova w v čase t je nepriamo úmerná uhlovej vzdialenosti medzi vektorom diskurzu a časovým variantom. Uhlovú vzdialenosť je možné interpretovať ako dĺžku najkratšej cesty medzi nenormalizovaným slovným vektorom L_2 a diskurzným vektorom L_2 na jednotkovej guli. Výber uhlovej vzdialenosti na rozdiel od kosínusovej podobnosti je rozhodujúci pre zabránenie hyperparametrového ladenia.

Potom sa vektorová reprezentácia vety s definuje ako odhad MAP vektora diskurzu c_s. Za predpokladu rovnomerného rozloženia cez možné c_s je odhad MAP tiež odhadom MLE pre c_s. Odhadom maximálnej pravdepodobnosti je približne vážený priemer slovných vektorov, kde častejšie slová sú znížené. V skutočnosti sa veľmi podobá váhovej schéme SIF („smoothed inverse frequency“)[2].

Autor detailne popisuje vylepšenia modelu vo svojej publikácii [11]. Odhadne sa m spoločných diskurzných vektorov, ako prvé m singulárne vektory z rozkladu vážených priemerov vektorov s jedinečnou hodnotou. Nech λ_i sú váhy spoločných diskurzných vektorov, ktoré môžu byť interpretované ako pomer rozptylu jedinečných hodnôt diskurzu v časovom variante. V skutočnosti sú jedinečné pre slovo, pre ktoré sa hodnotí pravdepodobnosť pozorovania slova v čase. Podľa [2], je odstraňovanie spoločných diskurzných vektorov forma odstránenia šumu, zvyšovanie m by malo zlepšiť výsledky. Tento postup sa nazýva čiastočné odstraňovanie spoločných komponent.

Pretože váhovacia schéma nepotrebuje žiadne ladenie parametrov, nazýva sa „unsupervised smoothed inverse frequency“ (uSIF). S využitím tohto modelu je možné vyriešiť problém podobnosti textu bez prípravy datasetu, na ktorom sa bude daný model učiť.

S–Bert

Publikácia [31] sa venuje sémantickej podobnosti viet, pričom predstavuje nový model *Sentence–Bert* popísaný v sekcii 2.8. Porovnanie časti výsledkov experimentov tohto modelu je zobrazené v tabuľke 3.1, konkrétne nad testovacím datasetom pre prvú úlohu v podujatí *SemEval–2017*. Do úvahy pripadá využitie práve mapovania vektorov slov alebo viet (*embeddings*). V prevádzaných experimentoch využili rôzne mapovania textu pomocou mo-

delov, ako *Glove* 2.6, *Universal Sentence Encoder* [6], *Bert/RoBERTa*. 2.8. Model *S–Bert* bol trénovaný nad kombináciou datasetov SNLI [5] a MNLI [41].

Model	**STS 17 benchmark [7] skóre**
Avg. GloVe embeddings	58.02
Avg. BERT embeddings	46.35
BERT CLS–vector	16.50
InferSent–Glove	68.03
Universal Sentence Encoder	74.92
SBERT–NLI–base	77.03
SBERT–NLI–large	**79.23**
SRoBERTa–NLI–base	77.77
SRoBERTa–NLI–large	79.10

Tabuľka 3.1: Spearmanova korelácia ρ medzi kosínusovou podobnosťou vetných reprezentácií a značkami pre rôzne úlohy týkajúce sa podobnosti textu (STS). Výkon sa konvenčne uvádza ako $\rho \times 100$.

3.8 Súčasný stav

Základom analýzy sentimentu je pripravený dataset, ktorého kvalita ovplyvňuje výslednú úspešnosť klasifikácie/predikcie. Jednotlivé klasifikačné algoritmy sú hodnotené na pred–pripravených, manuálne vytvorených datasetoch. Pre medzinárodné jazyky, ako napríklad angličtina to nie je žiadny problém. Takýchto datasetov existuje veľké množstvo. V doméne recenzií produktov je to napríklad dataset obchodného portálu Amazon publikovaný v [24].

Do istej úrovne sa dá tento proces automatizovať, napríklad použití recenzií pro/proti a získanie polarity sentimentu na úrovni sekcie pro/proti.

Táto práca je zameraná na český jazyk. Väčšina analyzovaných publikácii vychádza z datasetu publikovaného v práci [40], pričom sa jedná o recenzie obchodného portálu *Mall.cz*, česko–slovenskej filmovej databáze (csfd) a príspevkov sociálnej siete *Facebook*. Tento dataset je rozdelený podľa jednotlivých polarít, nie sú v ňom anotované aspety, ku ktorým sa ľudia vyjadrujú.

Dataset *heuréky* obsahuje čiastočne anotované dáta, sekcie plusov a mínusov. Dataset obsahuje 14 domén produktov a doménu recenzií obchodov. Jednalo by sa o jeden z najväčších datasetov recenzií pre klasifikačné úlohy pre český jazyk.

Kvalitu týchto dát je možné porovnať s využitím štúdie pre český jazyk [44] a jej výsledkami pri klasifikácii bipolárneho sentimentu. Autori použili doterajšie spôsoby analýz bipolárneho sentimentu pomocou vektorizéru a štandardných klasifikačných algoritmov. Zaujímavým zistením bude úspešnosť klasifikácie obdobných algoritmov na náhodne vygenerovanom datasete a porovnanie súčasných metód analýz sentimentu, napríklad spojením s transformermi 2.8.

Analýza sentimentu na úrovni aspektu predstavuje štandard. Pre túto analýzu je potrebný manuálne anotovaný dataset, kde anotátor identifikuje aspekty vo vete a priradí im polaritu. Tento proces je zdĺhavý. Práca sa venuje istej forme zhlukovania podobných viet založenej na podobnosti viet s využitím rôznych mapovaní viet do vektorového priestoru. S využitím zhlukovacieho algoritmu, ako je napríklad jednoduchý k–means. Systém zoradí vety podla podobnosti, pričom vrámci zhluku je možné využiť sumarizujúci model textu,

ako je napríklad LDA 2.5 na ešte drobnejšiu analýzu. Jedná sa o zoradenie viet bez natrénovaného datasetu podobnosti viet. Tieto zhluky a témy bude možné upravovať pomocou príslušného GUI.

3.9 Exportovanie datasetu

Systém pracuje s datasetom recenzií. Jeho najvhodnejším využitím bude generátor samotného datasetu recenzií na základe užívateľovho výberu parametrov, ktoré sa zohľadnia pri jeho generovaní. Výber môže byť závislí od typu úlohy (klasifikačná/regresívna), na základe ktorej bude dataset vygenerovaný, napríklad bipolárna analýza sentimentu nad doménou *kozmetika* s testovaním úspešnosti nad doménou *oblečenie* s vyžitím blokov textu, teda spojením všetkých viet z položky pozitívne/negatívne v rámci recenzie určitej dĺžky, s využitím predtrénovaného modelu Bert.

Generovanie bude prebiehať nad indexovanými dátami, teda bude spojené priamo s využitím indexovacieho nástroja, ktorý bude musieť implementovať nejaký dotazovací jazyk, ako napríklad Elasticsearch spomínaný v podkapitole 3.4, kvôli vytváraniu zložitých dotazov nad systémom recenzií.

3.10 Vizualizácia

Všetky výsledky analýzy dát a klasifikácií je potrebné zobraziť, aby boli lepšie pochopiteľné. V dnešnej dobe prevládajú skôr webové technológie zobrazovania rôznych analýz, preto sa za výstup predpokladá jednoduchá webová stránka zobrazujúca potrebné informácie.

Produkty a obchody

Systém si udržuje dataset recenzii produktov a obchodov z portálu heureka. Základným rozcestím by mali byť uvedené domény produktov a doména obchodov. U každého produktu je vhodné si uchovať takzvaný reťaz kategórii, ako napríklad *Heureka.cz | Bílé zboží | Malé spotřebiče | Péče o tělo | Příslušenství k holicím strojkům*. Na základe takéhoto reťazca sa dá vytvoriť strom, ktorý je možné prezentovať vo forme vyhľadávača pre užívateľa. Pri kliknutí na doménu sa zobrazia kategórie a následne produkty s počtom recenzií.

Pri zobrazení produktu alebo obchodu je vhodné uviesť krátke informácie o danom produkte, respektíve obchode. U produktov je vhodné zobraziť aj obrázok, ďalej odkaz na produkt/obchod, ktorý ponúka heuréka, základné štatistiky odporúčania obchodu/produktu, priemerný sentiment, graf príbytku recenzií v čase.

Zobrazenie plusov a mínusov z analýzy zhlukovania viet. Z ktorej sa môže užívateľ dozvedieť zhrnutie, o čom ľudia hovoria v rámci recenzií produktu/obchodu. Následne zobrazenie kľúčových slov, viažúcich sa k plusom, respektíve mínusom.

U každého produktu a obchodu sa predpokladá zobrazenie recenzií v istej forme tabuľky/zoznamu. Každý riadok tabuľky by mal obsahovať meno autora, dátum pridania recenzie, autorove ohodnotenie, metadáta analýz. Pri meta dátach sa jedná hlavne o zobrazenie prevádzaných analýz, informácie o výskyte sekcií pro a proti, zhrnutia. Za výsledky analýz je možné považovať, či bola recenzia vyhodnotená modelom pre predpoveď skóre recenzie a modelom pre vyhodnotenie irelevantných viet recenzie.

Recenzie

Recenzie tvoria základ tejto práce. Spôsob ich zobrazenie je kľúčový. Každá recenzia patrí k danému produktu alebo obchodu. V rámci analyzovaných recenzií je vhodné zvýrazniť kľúčové slová (aspekty) a k nim viažúci sa sentiment. V zobrazení sa predpokladá zobraziť sekcie pre a proti spolu s náhľadom na analýzu sentimentu každej položky pre a proti všetkými dostupnými modelmi. V tomto zobrazení sa ukážu doménovo špecificky natrénované zaujímavosti modelov. Isté vyjadrenie v rámci jednej domény nemusí znamenať to isté v prípade druhej. Súčasťou pohľadu je aj predpoveď skóre recenzie, pomocou regresného modelu a následné porovnanie s autorovím hodnotením.

Táto analýza bude trvať istý čas, predpokladá sa naivný prístup analýzy pri požiadavke na server. Vhodné bude reindexácia dát po aktualizácii, ich ohodnotenie modelmi. Dataset je veľký, pričom je isté, že všetky dáta nebudú reindexované, preto pri každej požiadavke na server, server zistí, či sa dané analýzy vykonávali. Ak áno, tak vráti pred spracované dáta, inak využije načítané modely pre „rýchlu analýzu“.

Štatistiky

Výsledný systém by mal v rámci užívateľského prostredia zobrazovať základné štatistiky datasetu heuréky, napríklad počet recenzií v čase, objem stiahnutých dát pri aktualizácii, počet produktov atď. Samozrejme sa berie ohľad na štatistiky analýz sentimentu nad datasetom.

Generovanie datasetu

Predpokladá sa užívateľské rozhranie, v rámci ktorého užívateľ zadá potrebné informácie do pripraveného formulára. Formulár bude pozostávať z výberových elementov, aby užívateľ nemohol zadať nezmysly. Formulár bude obsahovať na výber:

- Doménu/Domény – z ktorých sa bude dataset generovať.
- Kategória.
- Typ úlohy – aká úloha bude predmetom exportu datasetu.
  - Export prostých viet z recenzií.
  - Bipolárna klasifikácia sentimentu z položiek pre a proti.
  - Predpoveď hodnoty skóre sentimentu pomocou položky *rating*.
- Granularitu – závisí na úlohe, generovanie viet, súvetí, celých recenzií.
- Interval dĺžky záznamu.
- Výstup pre daný model, napríklad pre model Bert.
- Skupina príznakov, ako napríklad vyrovnaný dataset (pre bipolárnu klasifikáciu).

Výber kategórie sa viaže znova k obdobnému listu, ako v prípade zobrazenia produktov a obchodov.

Zhlukovanie viet

Prevádzanie analýzy na zhlukovanie viet zo sekcií plus a mínus bude možné vykonať prostredníctvom užívateľského rozhrania. Pohľad bude obsahovať výber kategórie z domény produktov, respektíve obchod alebo konkrétny produkt. Po kliknutí na zvolený produkt alebo kategóriu alebo obchod sa zobrazí dialóg s nasledujúcimi možnosťami:

- metóda mapovania slov/viet
- zhlukovacia metóda
- počet zhlukov
- počet tém pre zhluk

Editácia zhlukovacích experimentov

Aplikácia bude obsahovať aj istý pohľad na zobrazenie vykonaných experimentov zhlukovania, obdobne ako pri pohľade na recenzie produktu/obchodu. Pri zobrazení konkrétneho zhlukovacieho experimentu sa predpokladá istá editácia zhlukov, tém a viet do nich zaradených. Prípadne vytváranie nových zhlukov a tém pri zásahu užívateľa systému. Samozrejmosťou je mazanie zhlukovacieho experimentu.

Zobrazenie zhlukov obdobne v tabuľke spolu so zhlukami a počtom viet. Predpokladá sa zobrazenie histogramu rozloženia viet cez zhluky.

Systém užívateľov

Výsledná aplikácia predpokladá istý počet pohľadov na komunikáciu so systémom. V systéme sa definujú úrovne užívateľov na prezeranie výsledkov analýz recenzií. Tiež do systému nebude povolené vstúpiť každému, teda sa predpokladá autentifikácia užívateľa na strane front–endu.

Základný (jednoduchý) užívateľ bude môcť prezerať recenzie produktov a výsledky aktualizácii. Pokročilý užívateľ bude eskalovať práva základného užívateľa a bude mu sprístupnená možnosť vytvárať datasety, zhlukovať recenzie a podobne.

Demo natrénovaných modelov

Z predpokladu faktu, že všetky recenzie nebudú pred spracované všetkými modelmi, vyplýva isté nahratie modelov do pamäti na strane serveru (back–end). Do úvahy pripadá naivné využitie modelov vo forme dema, kedy bude GUI obsahovať formuláre pre odoslanie časti textu a zvolenie istého modelu a jeho následné vyhodnotenie a zobrazenie na strane GUI.

Kapitola 4

Realizácia systému

V tejto kapitole sú uvedené výstupy práce analýze recenzií výrobkov, stručný popis implementácie výslednej aplikácie, subsystémov, indexácie recenzií, samotný spôsob sťahovania recenzií, exportovanie dát prostredníctvom generovania datasetov na klasifikačné úlohy.

Popísanie klasifikácie a spôsob začlenenia modelov do implementácie, štatistiky modelov využitých v práci, spôsoby implementácie zhlukovania viet na základe podobnosti vetných reprezentácii, implementácia koncového serveru. Nasleduje implementácia klientskeho serveru spolu s ukážkami jednotlivých pohľadov na výslednú aplikáciu. Na záver kapitoly sú uvedené problémy a možné rozšírenia systémov analýzy recenzií.

4.1 Architektúra aplikácie

Systém je prevažne implementovaný v jazyku *Python 3*, ktorý je obľúbený v rámci analýzy dát a strojového učenia. Klientsky server je implementovaný v *Javascripte* s využitím *Vue.js*. Aplikácia je zložená z troch systémov, pričom každý z týchto systémov môže byť prevádzkovaný na inom fyzickom počítači.:

- Indexačný systém – Elasticsearch
- Systém analýzy recenzií a koncový server
- Klientsky server

4.2 Indexácia

Na indexáciu dát bol zvolený systém elasticsearch 2.2. Dosahuje najlepšie výsledky podľa portálu database–engines[1]. Naviac ponúka aj vlastný dotazovací DSL jazyk založený na JSON.

ElasticSearch

V práci je využitá verzia Elasticsearch 7.4.6. Dotazovanie so systémom elasticsearch zabezpečuje trieda `Connector`, jedná sa o `CRUD` dotazy nad dokumentami. Elastic podporuje aj SQL dotazy a transformáciu SQL dotazov do DSL jazyka.

[1] https://db-engines.com/en/ranking/search+engine

V tabuľke 4.1 je možné vidieť dáta, obsiahnuté v indexačnom systéme Elasticsearch analýzy výrobkov. Najviac recenzií v kategórii produktov obsahuje index *kozmetika*, index *produkt* slúži na indexáciu produktov, ku ktorím sa recenzie viažu. Najväčší počet dát obsahujú práve recenzie obchodov.

index	docs.count	store.size
shop_review	**3 559 732**	**2.2gb**
kosmetika_a_zdravi	**503 706**	**329mb**
elektronika	348 269	314mb
hobby	321 042	215mb
dum_a_zahrada	284 353	209mb
chovatelstvi	252 190	167mb
filmy_knihy_hry	233 121	190.6mb
bile_zbozi	229 928	210mb
detske_zbozi	210 683	163.9mb
sport	181 424	152mb
jidlo_a_napoje	100 460	51mb
auto–moto	80 761	80mb
obleceni_a_moda	38 742	22.9mb
stavebniny	36 320	26mb
sexualni_a_eroticke_pomucky	29 765	22mb
product	618 933	214.4mb
shop	1 042	706.9kb
domain	18	14.2kb
users	2	10.2kb
actualize_statistic	434	76.9kb
experiment_cluster	29	23.6kb
experiment	2	19kb
experiment_topic	85	51.6kb
experiment_sentence	15 112	4.5mb

Tabuľka 4.1: Prehľad indexov Elasticsearch a pamäťového miesta.

Počas pravidelných aktualizácii dokázal systém indexovať cez šesť miliónov štyristotisíc recenzií produktov a obchodov. Celková veľkosť datasetu je štyritisíc tristo megabajtov.

Pri indexácii recenzií sa využíva *POS tagger Morphodita*[2], pomocou ktorého sa spracujú záznamy `pros`, `cons`, `summary`. Odstráni sa diakritika, stop slová, emotikony sa nahradia frázou a následne sa text tokenizuje pomocou *Morphodity*. Túto funkcionalitu zabezpečuje trieda `MorphoTagger`. Výsledná recenzia obsahuje aj položky `pros_POS`, `cons_POS`, `summary_POS`.

Elastic Connector

Na komunikáciu systému s ElasticSearch slúži modul `elastic_connector` a jeho trieda `Connector`. Bez argumentov konštruktora sa pripája na localhost. Ponúka CRUD rozhranie ku každému indexu. Pri inicializácii si vytvára slovníky názov domény na index, názov kategórie na index.

[2] http://lindat.mff.cuni.cz/services/morphodita/info.php

Dataset recenzií je objemný. Výsledok niektorých dotazov obsahuje niekoľko desiatok tisíc dokumentov. Pri dotaze na dáta sa využíva metóda `_get_data()`. Jej argumentami sú doména, ktorá sa prevedie na index, kategória a telo dotazu popísané v DSL jazyku. Pomocou metódy `get_count()` sa zistí počet dokumentov, ktoré sú výsledkom dotazu. V prípade, že počet dokumentov je väčší ako desať tisíc tak sa využije takzvané „scroll api“, využité v metóde `__scroll()`, na získanie väčšieho počtu dokumentov.

4.3 Sťahovania recenzií

Funkcionalitu zabezpečujú dve triedy `HeurekaIndex` a `HeurekaCrawler`. Prehľadávanie heuréky s cieľom indexácie recenzií je uskutočnené v dvoch krokoch.

V prvom kroku s využitím triedy `HeurekaIndex`, kde sa prehľadáva heuréka pomocou URL adries doménových kategórii, pomocou metódy `parse_domain()` sa prehľadávajú jednotlivé podkategórie domény a vyhľadávajú sa produkty, ktoré obsahujú recenzie. URL adresy produktov sa následne uložia do textových súborov reprezentujúcich doménové kategórie.

V druhom kroku s využitím triedy `HeurekaCrawler`, kde v metóde `task()`, sa prechádzajú jednotlivé textové súbory URL adries produktov a pomocou triednej metódy `parse_product_page()` sa naindexuje produkt s recenziami do indexačného systému elasticsearch, ktorý je popísaný v podkapitole 4.2, pomocou metódy `add_to_elastic()`.

Aktualizácia recenzií produktov je implementovaná pomocou triedy `HeurekaCrawler` a jej metódy `task_actualize()`. Heuréka ponúka pre každú kategóriu produktu zobrazenie najnovších recenzií, S využitím dotazu do systému elasticsearch na získanie URL adries všetkých podkategórii domény, dotazom pomocou metódy `get_category_urls()` triedy `Connector` začne pavúk prechádzať najnovšie recenzie categórii produktov volaním metódy `actualize_reviews()`. Následne pre každú novú recenziu kontroluje, či sa už nenachádza v systéme, pomocou metódy `get_review_by_product_author_timestr()` triedy `Connector`, ak áno, tak končí aktualizácia pre danú pod-kategóriu produktov a pokračuje ďalšou.

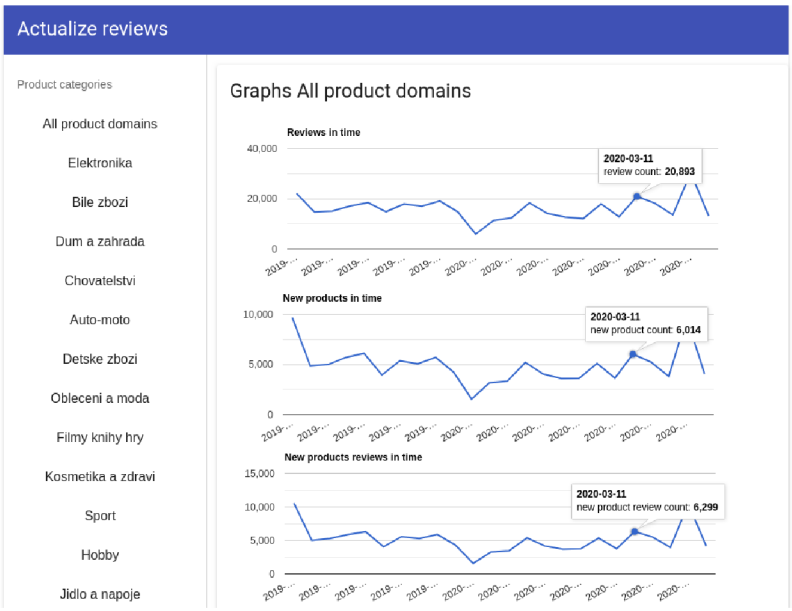
Metóda `actualize_reviews()` naplní slovník aktualizovaných recenzií produktov, ktoré sa následovne prechádzajú v cykle, pričom sa volá metóda `add_to_elastic()` na indexovanie nových produktov s recenziami alebo samotných recenzií.

Indexácia obchodov a ich recenzií je prevádzaná v jednom kroku s využitím triedy `HeurekaCrawler`. Pomocou metódy `task_shop()`, v ktorej sa prechádzajú jednotlivé URL adresy doménových kategórii. Obchody sa indexujú v metóde `parse_shop_page()` a ich recenzie pomocou metódy `parse_shop_revs()`.

Pri aktualizácii recenzií obchodov je nutné prechádzať každý obchod, heuréka neponúka zobrazenie najnovších recenzií pre všetky obchody. V rámci každého obchodu prechádzať recenzie, dotazovať sa systému na každú recenziu a v prípade, že sa recenzia nachádza v systéme tak skončiť a pokračovať ďalším obchodom.

Štatistiky aktualizácie produktov sú zobrazené v grafe 4.1, kde je možné vidieť isté trendy pribúdania recenzií produktov v čase. Aktualizácia prebiehala raz za týždeň.

Štatistiky sťahovania recenzií produktov sa ukladajú do elastic search pomocou metódy `submit_statistic()`, jedná sa o počty nových recenzií, nových produktov, nových recenzií nových produktov a počet produktov s novými recenziami. V metóde `add_to_elastic()` sa vykoná aj prvotná analýza recenzií s využitím filtrovacieho modelu a modelu na predpoveď skóre recenzie. Spôsob analýzy je uvedený v podkapitole Klasifikácia 4.5. Aktualizácia



Obr. 4.1: Pohľad štatistík aktualizácie systému.

recenzií produktov sa spúšťa pomocou skriptu `crawler.sh`. Tento skript môže byť spúšťaný pomocou nástroja pre správu úloh, ako napríklad `cron`.

4.4 Generovanie datasetu

Funkcionalitu zabezpečuje trieda `Generator` spolu so skriptom `generate_dataset.py` dokáže generovať dataset na vygenerovanie viet recenzií, bipolárnu klasifikáciu sentimentu alebo predpoveď skóre sentimentu (regresia). Skript je spustiteľný samostatne, ponúka vlastné rozhranie na generovanie datasetu, pričom daný dataset vytvorí v pracovnom adresári.

Je možné si zvoliť doménu, napríklad *biela technika*, spomínaný typ úlohy, vyrovnanosť datasetu. Voľbu kategórii buď pomocou integeru, zostupne zoradené kategórie podľa počtu recenzií alebo podľa názvu kategórie. Dataset je možné generovať z celých recenzií alebo zo sekcii +/- na štyroch úrovniach:

- na úrovni tokenov [, .]
- na úrovni položky +/-
- agregáciou všetkých položiek +/- recenzie
- celá recenzia

Ďalším parametrom je veľkosť vety, ktorý predstavuje interval $[min, max]$. Pri veľkosti vety v intervale $[1, 2]$ sa prevažne vyskytujú nezmyselné vety typu „*Nic.*“ alebo „*Zatím nic.*“, „*Nevím.*“ a ich obmeny. Tieto vety sa vyskytujú prevažne v sekcii záporov recenzie. Voliteľná je značka na generovanie vyrovnaných datasetov.

Pri klasifikačných úlohách sa predpokladá využitie modelu Bert, generátor dokáže priamo generovať súbory *train.tsv*, *dev.tsv*, ktoré môžu byť v konkrétnych implementáciách v podkapitole 4.5.

Trieda `GeneratorController` je využitá na strane koncového servera, kľúčová je metóda `generate()`, ktorá slúži ako vstup, podľa predaných argumentov sa môžu volať metódy `__embeddings_task()` na vygenerovanie viet, `__cls_task()` na generovanie klasifikačných datasetov, `__regression_task()` na vygenerovanie regresívneho datasetu.

Datasety recenzií sa generujú postupným prechádzaním kategórii produktov v metóde `get_data_api_call()`. Pomocou metódy `parse_reviews()` sa dotáže elastic connectora na recenzie kategórie a následne sa vytvárajú vety podľa zadaných argumentov.

4.5 Klasifikácia

Práca sa zaoberá analýzou recenzii so zameraním na analýzu sentimentu, v rámci ktorej je vhodné mať pred trénovaný model jazyka. Model Bert bol zvolený ako zástupca transformerov aj kvôli dostupnosti pred trénovaných modelov pre český jazyk. Moduly zaoberajúce sa klasifikáciou sú dostupné v balíku *clasification*

Knižnice ako *Tensorflow* a *Pytorch* ponúkajú rozhranie, pomocou ktorého je možné využiť tento model. Bohužiaľ *Tensorflow* aktuálne nie je kompatabilný s najnovším ovládačom od *Nvidie* a skutočnosti, že na využitie modelu Bert-base je potrebná grafická karta s aspoň 11 GB pamäte bola zvolená knižnica *Pytorch* ako východzia.

Pytorch

Jedná sa o open-source knižnicu pre strojové učenie, vytvorená spoločnosťou Facebook. Poskytuje rozhranie v jazyku Python. Umožňuje využitie viacerých GPU. Vektory, respektíve matice, sú reprezentované tenzormi, nad ktorými prebieha výpočet. Tvar tenzoru určuje počet a veľkosť dimenzií.

Scikit-learn

Open-source knižnica, ktorá ponúka nástroje na prediktívnu analýzu dát, či už sa jedná o klasifikáciu, regresiu, zhlukovanie, preprocessing dát a podobne. Na rozdiel od Pytorch nie je framework Scikit-learn určený ako takzvaný „deep learning framework“ a natívne nepodporuje výpočty na grafickej karte.

Bert

Model je využitý pri klasifikácii sentimentu a predpovedi skóre recenzie. Bola využitá knižnica *transformers* od týmu Huggingface[3]. Ako pred trénovaný model bol využitý model `BERT-Base, Multilingual Cased` s parametrami:

- 12 hláv a vrstiev pozornosti
- veľkosť skrytého stavu 768
- natrénovaný na 104 jazykoch
- citlivý na veľké písmená

[3] https://huggingface.co/transformers/

Spracovanie dát je implementované pomocou skriptu `processors_cls.py`, v ktorom sa nachádzajú jednotlivé procesory dát. Procesor dát pre úlohu bipolárnej klasifikácie sentimentu je implementovaný s využitím triedy `BipolarSentiment`, procesor pre úlohu predikcie skóre sentimentu je implementovaný pomocou triedy `ReviewRatingRegression`. Tokenizér nemodifikuje veľkosť písma, text sa rozdelí pomocou bielych znakov na tokeny. Model využíva tokenizér *WordPiece*, ktorý rozdelí tokeny na najčastejšie n-gramy pozostávajúce zo skupiny znakov. Veľkosť slovníka činí okolo 120 tisíc tokenov *WordPiece*.

Predtrénovaný model je dostupný práve pre knižnicu Tensorflow, pomocou skriptu `tf_to_torch.py` sa prevedie jednoduché načítanie váh modelu a následné konvertovanie modelu do formátu, ktorý využíva pytorch.

Následne klasifikácia, resp predikcia pomocou modelu Bert je implementovaná v skripte `transformers_cls.py`. Model bol ladený v šiestich epochách. Pre validáciu modelu boli využité metriky z knižnice *scikit-learn*.

Trénovanie modelu je možné s využitím skriptu `run_cls.sh`, v ktorom sú nastavené cesty a argumenty využité pri trénovaní modelov analýzy recenzií. Obdobne využitím skriptu `run_cls_pred.sh` sa spustí validácia modelu na trénovacom datasete, pričom sa vygeneruje dodatočne aj súbor s mapovaním testovacieho datasetu a predikcií modela *eval_results.tsv*, súbor s aplikovanými metrikami ako F1, presnosť, konfúzna matica, korelácia, stratu krížovej entropie *eval_results.txt*.

Každý natrénovaný model môže byť využitý pomocou modulu `Bert_model.py` a jeho triedy `Bert_model`,v ktorej sa model prepne do vyhodnocovacieho stavu. Následne táto trieda slúži ako jednoduchý wrapper na každý Bert model. Vyhodnotenie klasifikácie vety natrénovaným modelom je implementované s využitím metódy `eval_example()`.

FastText

Využitie FastText modelu spočíva hlavne v získaní vektorových reprezentácii slov. V práci využívam predtrénovaný model, ktorý bol trénovaný na datasete wikipédie pomocou FastTextu. Vektory obsahujú 300 dimenzi a boli získané pomocou *skip-gram* modelu popísaného publikáciou [3]. Pred trénované modely sú voľne dostupné na portály fasttext[4].

Pre prácu s modelom FastTextom je využitá knižnica Genism, inštanciu modelu reprezentuje trieda `FastText`. Pred trénovaný model je možné načítať s využitím funkcie `load_facebook_model()`.

Na FastText sú priamo využité váhovacie schémy SIF a uSIF pri filtrovaní recenzií a zhlukovaní podobných viet na získanie vektorových reprezentácii viet.

Fast sentence embeddings

Získavanie vektorových reprezentácii viet/textu je implementované pomocou knižnice `fse`, publikovanej pomocou Githubu[4]. V práci využívam triedy `SIF` a `uSIF` implementujúce váhovanie popísané v sekcii Vektorové mapovanie viet 3.7.

Konštruktoru týchto tried je predaná inštancia modelu FastText, buď s využitím pretrénovaného modelu fasttext pomocou metódy `load_facebook_model()`. Ďalším spôsobom je vytvorenie inštancie triedy `FastText`, na základe predaného pola viet.

[4] https://fasttext.cc/docs/en/pretrained-vectors

SVM

Model SVM je využitý pri filtrovaní irelevantných viet, kde dosahuje lepšie výsledky v porovnaní s modelom Bert. Oba tieto modely boli nátrénované na tých istých dátach. SVM je implementované pomocou knižnice Scikit-learn triedou *C-Support Vector Classification.*

Bipolárna analýza sentimentu

Analýza sentimentu na úrovni vety, respektíve dokumentu je vykonávaná pomocou modelu Bert. Tento model je ladený v doméne produktov nad klasifikačnou úlohou bipolárnej klasifikácie. Dokopy je natrénovaných 14 modelov pre každú doménu produktov a jeden všeobecný model.

Pre každý model bol vygenerovaný dataset domény, vytvorenie týchto datasetov je jednoduché, hlavne kôli čiastočnej anotácii užívateľov heureky pri vytváraní recenzií. Jedná sa o položky pre a proti . Štatistiky sú popísané v podkapitole Natrénované modely 4.6.

Filtrovanie irelevantných recenzií

Pri filtrovaní irelevantných viet boli prevádzané experimenty s využitím modelu Bert a modelu SVM na bipolárnu klasifikáciu viet na irelevantné a relevantné vety. Z celkového počtu viet v recenziách predstavujú irelevantné vety menšiu časť, preto vytvorenie vyrovnaného datasetu je problém, pričom je potrebné manuálne prejsť recenzie. Irelevantné vety sa na druhej strane často opakujú v datasete. Jedná sa o vety prevažne v zápornej sekcii recenzie typu: „Zatím nic.“, „Zatím nemohu říct.“, „Zatím jsem na nic nepřišla.“, „ Mám ho krátce.“.

Následne som vytvoril manuálne anotovaný dataset obsahujúci cez 7000 viet, pričom dataset nie je vyrovnaný. Po rozdelení dát na trénovací a testovací dataset dosahoval model Bert F1 85.17. Pri bližšej analýze som zistil, že až v tretine prípadov model klasifikoval irelevantné vety ako false positive. Následne som využil model SVM spolu s využitím vetných vektorových reprezentácii pomocou knižnice fse a váhovacej schémy uSIF. Model dosiahol úspešnosť s macro F1 90.00.

Model uSIF je vytvorený na základe predtrénovaného model FastText, na ktorý je aplikovaná váhovacia schéma uSIF, následne je ladený na datasete viet z domény produktov. Jedná sa o 4 798 429 viet v dobe vytvorenia modelu.

SVM model je reprezentovaný modulom `SVM_model` a triedou `SVM_Classifier`. Trieda dokáže znova vyvoriť príslušné modely uSIF a SVM pomocou metódy `create_model()` spolu s vypočítaním metrík. Vyhodnotenie relevantnosti vety je vykonané pomocou metódy `eval_example()`.

V prípade využitia modelu pri filtrovaní viet recenzii pri automatickej aktualizácii je využitá trieda `HeurekaFilter` modulu `heureka_filter`. Pri aktualizácii sa každá položka pro/proti/zhrnutia recenzie vyhodnotí pomocou metódy `is_irrelevant()`.

Jednoslovné sekcie sa automaticky zahadzujú, pretože neobsahujú názorové slovo a pomenúvajú aspekt, náhodné slová, pričom trénovanie modelu na jedno slovných vetách je nezmysel. Ak má veta viac ako jedno slovo je vyhodnotená pomocou modelu a v prípade ak sa jedná i irelevantnú vetu tak sa zapíše do špeciálneho logovacieho súbora *irrelevant_sentences.tsv.*

Logovací súbor predstavuje spôsob automatického zväčšovania datasetu. Po každej aktualizácii je potrebné prejsť nové vety, skontrolovať klasifikáciu modelu, rozšíriť pôvodný dataset a po prípade znova natrénovať klasifikátor.

U recenzií, ktoré boli spracované filtrovacím modelom sa pri indexácii pridáva flag *filter_model=True* na odlíšenia analyzovaných recenzií.

Predpoveď hodnotenia recenzie

Pri aktualizácii sa využíva aj regresný model Bert, ktorý je naučený na celom datasete recenzií. Model je reprezentovaný triedou `Bert_model`. Pri aktualizácii sa využíva trieda `HeurekaRating` modulu `heureka_rating`. Pomocou metódy *merge_review_text* sa konkatenujú pre/proti/zhrnutie a následne s vyžitím metódy `eval_sentence()` sa vyhodnotí predpoveď skóre modelom. Predpoveď skóre sa indexuje do recenzie pomocou položky `rating_model`.

4.6 Natrénované modely

V rámci práce sa používa 15 modelov zabezpečujúcich bipolárnu klasifikáciu sentimentu, založených na architektúre Bert. Model predpovede skóre recenzie, založený na architektúre Bert. Model SVM na klasifikáciu irelevantnosti viet v recenzii, ktorý využíva uSIF váhovú schému na vektorovú reprezentáciu textu pomocou modelu FastText.

V tabuľke 4.2 sú zobrazené modely bipolárnej klasifikácie sentimentu na úrovni sekcie pro/proti využité v práci. Vygenerované datasety sú vyvážené, pretože pri analýze dát vyplynulo, že užívatelia generujú viacej pozitívnych sekcií ako negatívnych. Najlepšiu úspešnosť dosahuje model *general*, ktorý obsahuje recenzie všetkých domén produktov.

model	ACC	F1	train	dev
general	95.62	95.65	565 144	141 286
kosmetika_a_zdravi	95.30	95.32	85 212	21 302
bile_zbozi	95.09	95.11	80 526	20 130
detske_zbozi	94.72	94.79	46 340	11 584
elektronika	94.52	94.57	120 368	30 092
chovatelstvi	93.91	93.94	46 174	11 542
sport	93.84	93.91	52 576	13 142
filmy_knihy_hry	93.72	93.77	44 380	11 094
dum_a_zahrada	93.49	93.59	51 100	12 774
auto–moto	92.70	92.71	12 952	3 236
hobby	91.92	92.07	15 960	3 990
stavebniny	90.71	90.92	5 514	1 378
sexualni_a_eroticke_pomucky	90.58	90.79	10 322	2580
jidlo_a_napoje	90.31	90.45	9 754	2 438
obleceni_a_moda	89.69	89.97	3 924	980

Tabuľka 4.2: Modely bipolárnej klasifikácie sentimentu na úrovni sekcie pre/proti.

V tabuľke 4.3 je zobrazená priemerná štvorcová chyba modelu predpovedi skóre recenzie. Pri trénovaní modelu sa zvolil vyrovnaný dataset kvôli faktu, že ľudia väčšinou hodnotia recenzie so skóre 100%. Maximálna predpoveď hodnoty skóre sa pohybuje okolo $90\% - 100\%$, minimálna v intervale $10\% - 30\%$.

Každá kategória obsahuje 6 439 viet, strop je daný kategóriou hodnotenia recenzií 10%. Testovací dataset obsahuje 3220 viet.

skóre	MSE	max	min
0.1	0.105	0.90	0.12
0.2	0.062	0.90	0.10
0.3	0.039	0.80	0.15
0.4	0.034	0.90	0.19
0.5	0.037	1.0	0.23
0.6	0.041	1.0	0.21
0.7	0.027	0.90	0.29
0.8	0.047	0.90	0.19
0.9	0.043	1.0	0.31
1.0	0.052	1.0	0.31

Tabuľka 4.3: Stredná štvorcová chyba, maximálna, minimálna predpoveď hodnoty skóre recenzie na testovacom datasete.

V tabuľke 4.4 sú zobrazené modely klasifikácie irelevantných recenzií. Klasifikačný model SVM spolu so uSIF váhovou schémou na FastText vektorovú reprezentáciu viet dosahuje podstatne lepšie výsledky ako model Bert. Ako som už spomínal v prípade modelu Bert, model až v tretine prípadov klasifikoval irelevantné vety ako false positive. Dataset relevantných a irelevantných viet je nevyvážený, čo môže spôsobovať klamlivé výsledky.

model	ACC	F1
Bert	85.17	90.41
SVM + uSIF emb	92.00	90.00

Tabuľka 4.4: Modely klasifikácie irelevantných recenzií.

4.7 Podobnosť viet

Úloha podobnosti textu som riešil pomocou dvoch modelov vektorovej reprezentácie textu. Naivné riešenie spočíva vo využití modelu LDA na získanie distribúcie tém cez jednotlivé dokumenty, no jedná sa o prosté štatistickú distribúciu slov.

Jedno z riešení spočíva vo využití modelu Bert a jeho vektorových reprezentácii *WordPiece* tokenov. Druhé riešenie spočíva vo využití čisto učenia bez dozoru s využitím natrénovaných vektorových reprezentácii FastText a váhovacej schémy SIF/uSIF.

LDA

Balík `experiments` obsahuje jednotlivé moduly, využité pri experimentovaní s dostupnými modelmi. Modul `LDA_model`, obsahuje triedu `LDA_model`, ktorá sa využila pri experimentovaní s vyhľadaním tém v recenziách a následne ich priraďovaním. Trieda využíva knižnicu *Gensim* pri práci s LDA modelom. Na zobrazovanie štatistík sa využíva knižnica *pyLDAvis*.

Pri spracovaní textu je využité predspracovanie textu pomocou triedy `MorphoTagger` a jej metódy `pos_tagging()`, ktorý prevedie slová do základného tvaru, odstráni stop slová a podobne. Využíva sa TF-IDF slovník slov, konkrétna implementácie LDA algoritmu je charakterizovaná triedou `LdaMulticore` od *Gensim*.

LDA je štatistické rozloženie slov medzi témami a tém medzi dokumentami. Tento model je využitý v rámci zhlukov viet na ešte drobnejšiu analýzu.

Bert embeddings

Publikácia [34] uvádza ako jedno z lepších riešení vo využití vektorových reprezentácii textu pomocou modelu Bert. Využil som pred trénovaný model Bert, uvedený v podkapitole Klasifikácia 4.5. Na získanie vektorových reprezentácii je využitá knižnica „Bert As a Service“. Vektorové reprezentácie WordPiece tokenov sú prevzaté z predposlednej vrstvy, odôvodnenie je popísané v dokumentácii služby[5]. Za združovaciu stratégiu vektorových reprezentácii bola zvolená REDUCE_MEAN na získanie vektorovej reprezentácie vety.

Implementáciou tohto experimentu sa zaoberá modul `bert_service`, metóda `cluster()`. Následne bol vygenerovaný dataset viet. Vektorové reprezentácie tokenov boli získané z predposlednej vrstvy, využití stratégie REDUCE_MEAN a následne využitím zhlukovacieho algoritmu kmeans, implementovaný triedou `KMeansClusterer` od frameworku `nltk` so stratégiou kosínusovej vzdialenosti.

Výsledky zhlukovania nedopadli dobre, podobné vety neboli zaradené do podobných zhlukov. Po dlhšom experimentovaní s počtom zhlukov, vrstvami embeddingov a združovacích stratégií som došiel k záveru, že využitie predtrénovaného modelu nie je vhodné. Lepšie je využitie ladeného modelu, napríklad na predpoklade nasledujúcej vety a následne využitie vektorovej reprezentácie CLS značky.

Prípadne využitím špecializovanej architektúry S–Bert a datasetu vetných trojíc. V tomto prípade je nutné vytvorenie takéhoto datasetu pre český jazyk, jedná sa o možné vylepšenie, respektíve ďalší postup práce.

Modul obsahuje aj metódu `bert_service_dialog()`, ktorá využíva 5 manuálne anotovaných zhlukov viet. Tieto zhluky sú prevedené na vektory viet a následne v dialógu sa po napísaní vety zobrazí 10 najpodobnejších viet, pomocou skalárneho súčinu normalizovaných vektorov=kosínusová podobnosť [43].

Za zmienku stojí aj metóda `visualize()`, ktorá pre mapuje vektorovú reprezentáciu tokenov do 2D grafu na zobrazenie.

SIF/uSIF vahovacia schéma

Finálne riešenie problému zhlukovania viet, na základe podobnosti spočíva vo využití pretrénovaného modelu, ako je napríklad FastText. Ziskanie vektorovej reprezentácie vety je vykonané pomocou váhovej shcémy SIF/uSIF. Pre implementáciu SIF/uSIF slúži knižnica `fse`, FastText model je zhodný s modelom použitým v podkapitole Klasifikácia 4.5.

Funkcionalita je implementovaná triedou `FastTextModel`. V metóde `__create_model()` sa vytvorí SIF model na základe buď dostupného pred trénovaného modelu FastText, alebo sa model vytvorí na základe predaných viet, ktoré sa budú zhlukovať. Následne sa SIF model natrénuje na vetách, ktoré sú predmetom zhlukovania.

Metóda `cluster_similarity()` slúži ako vstup do systému zhlukovania. Najskôr sa vytvorí SIF model pomocou metódy `__create_model()` a následne na základe vybratej metódy zobrazenia textu do vektoroveho priestoru vygeneruje matica. Naivné riešenie je priamo využitie vektorových reprezentácii SIF modelu, pričom vznikne matica veľkosti $n \times d$, kde n je počet viet zhlukovania a d je dimenzia vektoru.

Zaujímavejším riešením je vypočítanie kosínusovej podobnosti, respektíve vzdialenosti vety s každou vetou viet, ktoré sú predmetom zhlukovania. Takto vznikne matica veľkosti $n \times n$, kde n je počet viet.

[5] https://bert-as-service.readthedocs.io/

Na výslednú maticu sa využije zvolený algoritmus zhlukovania. V aktuálnej implementácii je využitý algoritmus kmeans, ktorý je implementovaný triedou `KMeansClusterer` od frameworku `nltk` pomocou metódy `__kmeans()`.

Zhlukovanie viet je priamo riadené kontrolérom endpointu na strane koncového servera, ktorý zabezpečuje ďalšie rutiny.

4.8 Back–end

Koncový server ponúka endpointy, ku ktorým sa je možné dotazovať. Ponúka rozhranie kontrolérov v balíku `controllers`, ktoré sa starajú o spracovanie dát v jednotlivých endpointoch. V práci využívam framework `Flask` na implementáciu koncového servera.

V jednotlivých sekciách uvediem zaujímavejšie kontroléry ponúkajúce zdroje pre jednotlivé koncové body API a ich dôležité metódy. Koncové body API sú popísané pomocou nástroja swagger[6] s využitím frameworku `Flask restful`[7]. Táto dokumentácia sa generuje pri spustení servera ako počiatočný bod[8]. Dokumentácia je zobrazená na obrázku B.1.

ExperimentClusterController

Kontorolér zabezpečuje operácie zhlukovania viet recenzií a jednotlivé CRUD operácie nad zhlukmi, témami, vetami. Zhlukovanie prebieha volaním metódy `cluster_similarity()`. Na základe vstupných argumentov sa vygenerujú datasety viet pre pozitívne a negatívne sekcie, následne sa vytvorí záznam v indexy experiment.

Pomocou metódy `__cluster()` predstavuje jadro funkcionality zhlukovania. Volá sa pre pozitívne a negatívne vety. Pomoc metódy `cluster_similarity()` spomínanej v sekcii 4.7 sa prevedie operácia zhlukovania viet pomocou zvolenej metódy reprezentácie textu, metódy zhlukovania.

Následne sa v každom zhluku nájde istý počet tém daný argumentom, pomocou modelu LDA. Tento model je implementovaný triedou `LDA_model`, pričom sa volá metóda `load_sentences_from_api()`, kde sa získajú aj príznačné slová, anglicky „salient words“, charakterizujúce dané témy v zhlukoch. Následne sa indexuje každá veta pomocou identifikátora experimentu, zhluku, témy.

ReviewController

Zabezpečuje dodatočnú analýzu recenzie pri zobrazení recenzie prostredníctvom GUI. Metóda `get_review_experiment()`. Metóda na základe id recenzie nájde recenziu v indexačnom systéme. Nie všetky recenzie sú predspracované, teda ich položky pro a proti vyhodnotené pomocou bipolárnych klasifikátorov domén, ohodnotené modelom na predpoveď skóre. Hlavný dôvod je veľkosť datasetu.

V konštruktore sa volá metóda `_load_models()`, v ktorej sa načítajú všetky modely bipolárnej klasifikácie, dokopy 15 modelov. Tak isto sa načíta aj irelevantný model, spolu s modelom uSIF na generovanie vektorových reprezentácii viet v rámci modelu.

V prípade, že recenzia neobsahuje položku *rating_model*, recenzia sa vyhodnotí pomocou modelu na predpoveď skóre, ak neobsahuje položku *pos_model*, respektíve *con_model* tak sa vyhodnotí každá položka pro/proti pomocou modelov domén v prípade dotazu.

[6] https://swagger.io/
[7] https://flask-restful.readthedocs.io/
[8] http://pcknot5.fit.vutbr.cz:42024/

Ak bola recenzia produktu súčasťou zhlukovacieho experimentu ktegórie produktu, respektíve obchodu, v každej sekcii pro/proti a v zhodnotení recenzie budú zvýraznené príznačné slová, vytvorené pomocou modelu LDA.

Metódy `get_sentence_polarity()`, `get_text_rating()`, `get_irrelevant()` natrénované modely pri klasifikácii textu.

GenerateDataController

Ponúka rozhranie na export datasetu, na základe predaných argumentov, pomocou metódy `generate_dataset()`, v ktorej sa volá metoda `generate()` triedy `GeneratorController`. Výsledkom je buď archív, v ktorom sa nachádzajú vygenerované súbory alebo chybová hláška.

ProductController

Zabezpečuje získavanie dát pre produkty/obchody. Obrázky produktov sa získavajú pomocou metódy `get_product_image_url()`, štatistiky trendov spokojnosti pomocou metódy `get_statistics()`.

Nie všetky recenzie sú predspracované všetkými dostupnými modelmi a neobsahujú položky *rating_model*, *filter_model*, ktoré sú vhodné pri zobrazovaní analyzovaných recenzií. Kvôli jednotnému zobrazeniu dát v GUI, sa v metóde `get_product_reviews()` sa kontrolujú tieto položky, a ak sa v recenzii nenachádzajú tak sa nastavia východzie hodnoty.

UserController

Pri posielaní dotazu na koncový bod je potrebné použiť autentifikačný token. Každá funkcia implementujúca obslužnú rutinu pre koncový bod API je zabalená pomocou wrapera `token_required()`.

Na vytváranie tokenu je využitý framework *jwt*. Token je vytváraný na základe užívateľovho mena a má platnosť po dobu 90 minút. Dôležitou položkou je *SECRET_KEY*, ktorá sa generuje náhodne pri každom spustení servera a má veľkosť 32 bajtov.

Tento token je generovaný pri prihlásení užívateľa do systému. Kontrolér ponúka metódy na prihlásenie registráciu užívateľa, pomocou metódy `create_user()` a autentifikáciu metódou `authenticate()`.

4.9 Klientsky server

Vizualizáciu zabezpečuje webové rozhranie, implementované v jazyku Javascript pomocou frameworku *Vue.js*[9] ako samostatná klientska aplikácia.

Komponenty sú prehľadne rozdelené podľa pohľadov v priečinku `components`. Smerovanie medzi komponentami je implementované pomocou rozšírenia *vue-router* triedou `Router`. Komunikácia medzi komponentami prebieha pomocou balíku *EventBus*.

Získanie REST zdrojov od API koncového servera majú na starosti funkcie jednotlivých Javascriptových súborov v priečinku `services`. Na odoslanie požiadavky na REST API koncového serveru je využitý balíček *vue-axios*.

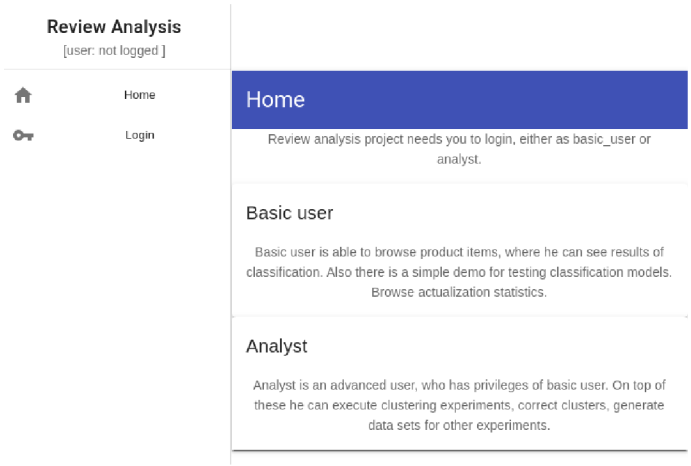
Balík *vuex* je použitý na manažovanie stavov aplikácie. Je využitá ako centralizované úložisko pre všetky komponenty, implementované triedou *Vuex.Store*. V úložisku sa ukladá

[9] https://vuejs.org/

reprezentácia aktuálne prihláseného užívateľa, autentifikačný token, ktorý je vygenerovaný koncovým serverom, pole experimentov, aktuálne zvolený produkt/obchod, stromová štruktúra kategórii. Aktuálny stav týchto objektov sa mení pomocou takzvaných mutácii.

Systém užívateľov

Klientská aplikácia využíva systém práv užívateľov. Užívateľ s právom bežný užívateľ, *basic_user* a užívateľ s právom analytik, *analyst*. U neprihláseného užívateľa sa zobrazí pohľad zobrazený na obrázku 4.2, pomocou komponenty `NotLoggedHome`.



Obr. 4.2: Pohľad na domovskú stránku neprihláseného užívateľa.

Po kliknutí na položku *Login* v hlavnom menu sa zobrazí pohľad na prihlásenie sa do systému implementovaný komponentou `LoginUser`. Po zadaní užívateľského mena a hesla a následným kliknutím na tlačítko *Login* sa zobrazia pohľady s úrovňou prístupu užívateľa. Autentifikačný token sa uloží do spoločného úložiska a inštancia užívateľa sa uložia do spoločného úložiska.

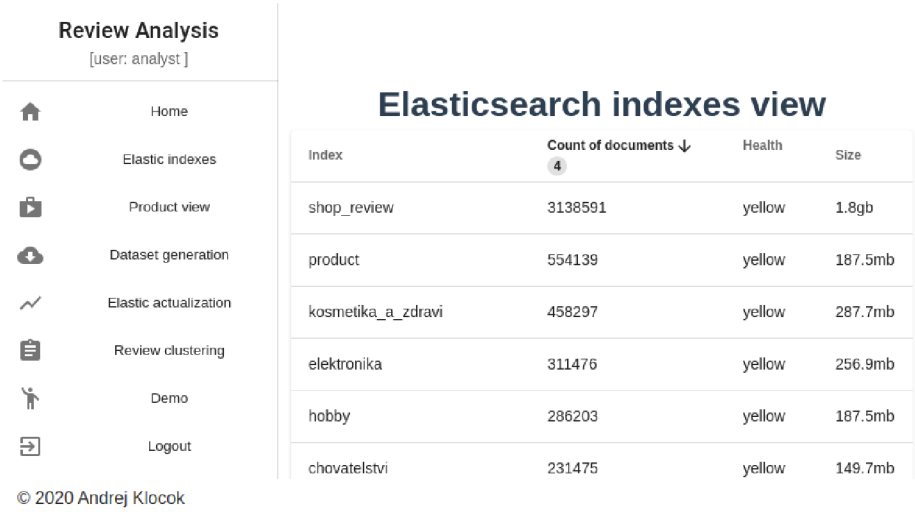
Pohľad bežného užívateľa je zobrazený na obrázku B.2 komponentou `UserHome`. Má k dispozícii funkcionalitu prezerania produktov/obchodov a ich recenzií, prezeranie štatistík aktualizácii, vyskúšanie natrénovaných modelov.

Pokročilí užívateľ – analytik eskaluje práva bežného užívateľa, dodatočne má prístup k funkcionalite generovania datasetov a prevádzania zhlukovacích experimentov, prezerania indexov. Pohľad pokročilého užívateľa je zobrazený na obrázku B.3.

Indexy

Pohľad zobrazenia indexov elasticsearch je zobrazený na obrázku 4.3. Slúži na vytvorenie prvotnej predstavy o systéme analýzy recenzií a veľkosti dostupného datasetu recenzií. Poskytuje základné informácie o počtu dokumentov v každom indexe, pamäťového priestoru a zdravia indexu.

Tento pohľad môže slúžiť v rámci administrátorského pohľadu, kde administrátor môže manažovať rôzne uzly elasticsearch klientov, pričom by sa mohli dáta distribuovať na viacej uzlov. Vytváranie záloh, obnova systému z istej zálohy a podobne.

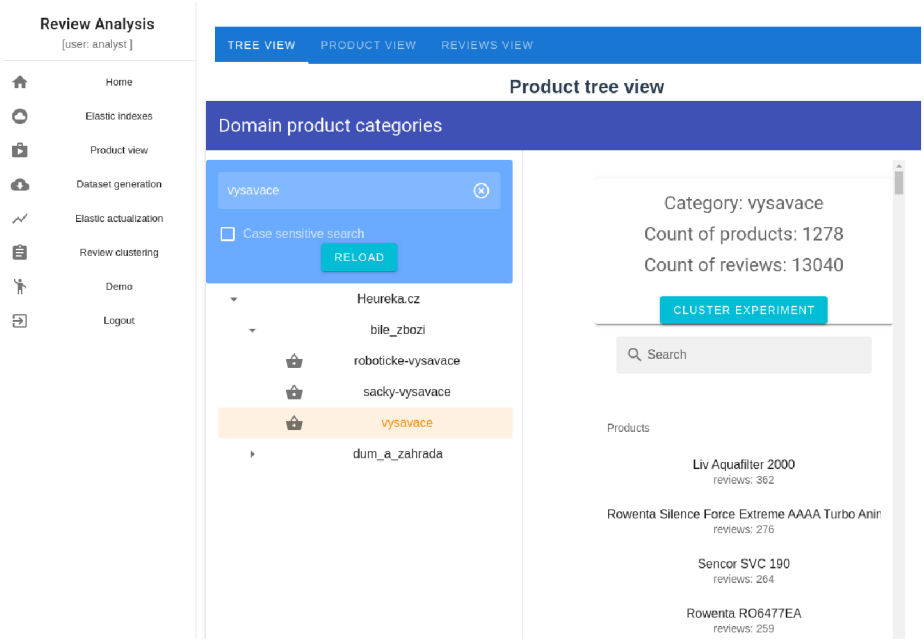


Obr. 4.3: Pohľad na indexy elasticsearch.

Produkty/Obchody

Funkcionalita prechádzania domén a ich kategórii smerom k produktom a ich recenziám je implementovaná viacerými komponentami. Na navigáciu medzi komponenty je možné využiť záložkové menu zobrazené v hornej strane pohľadu.

Po kliknutí na položku menu *Product view* sa zobrazí pohľad charakterizovaný obrázkom 4.4. Tento pohľad je implementovaný komponentou `ProductTreeView`.



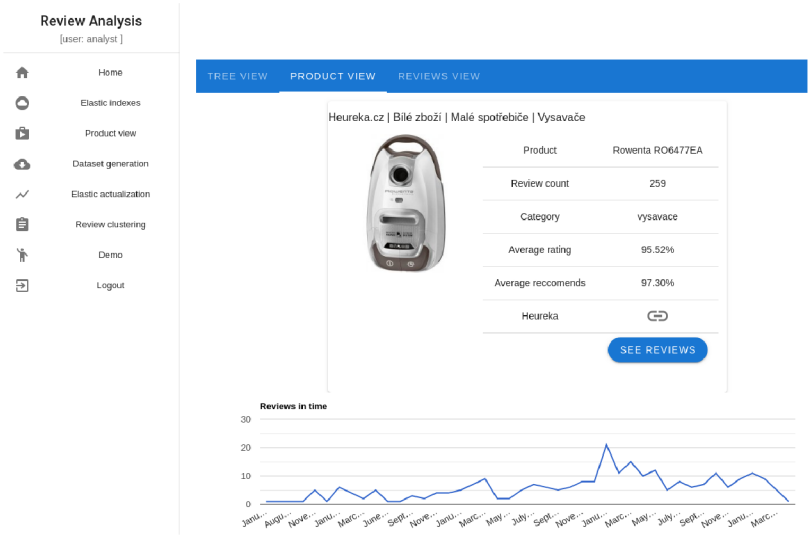
Obr. 4.4: Pohľad na kategóriu *Vysávače*.

Jednotlivé kategórie domén sú reprezentované v stromovej štruktúre, kde koreňom je *Heureka.cz*. Následne sú kategórie abecedne zobrazené. V kategóriách sa dá vyhľadávať aj pomocou textového filtra. Po vybratí kategórie sa v druhej polovici pohľadu zobrazia

jednotlivé produkty spolu s počtom recenzií zoradené podľa počtu recenzií. V hornej časti sú zobrazené štatistiky kategórie ako počet produktov a počet recenzií. Po stlačení tlačítka *cluster experiment* sa zobrazí výsledok zhlukovania viet recenzií. Tento pohľad uvediem neskôr.

Po kliknutí na produkt sa zobrazí pohľad produktu zobrazený na obrázku 4.5. Tento pohľad implementuje komponenta `ProductView`. Na obrázku je možné vidieť obrázok produktu, ktorý sa dostane dynamicky zo stránky heureky pomocou koncového servera. Ďalej cestu *breadcrumbs* kategórii produktov od názvu obchodu, domény, jednotlivé úrovne kategórii, tabuľku jednoduchých štatistík s odkazom na heuréku.

V spodnej časti pohľadu je možné vidieť graf pribúdania recenzií produktov v čase podľa mesiacov. Na spodnej strane pohľadu sa nachádza výsledok zhlukovania viet recenzií produktu, príznačné slová pre sekcie pro/proti.



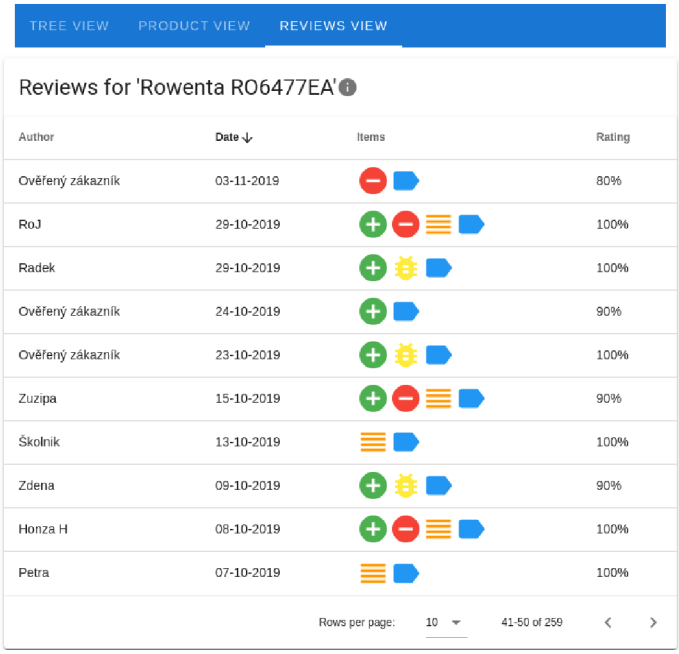
Obr. 4.5: Pohľad zobrazenia produktu *Rowenta RO6477EA*.

Po kliknutí na tlačítko *See Reviews* sa zobrazí náhľad recenzií produktu, ktorý je zobrazený na obrázku 4.6. Tento pohľad implementuje komponenta `ProductReviewView`. V pohľade je možné vidieť jednotlivé recenzie zobrazené v tabuľke. V tabuľke je možné radiť recenzie pomocou dátumu vytvorenia recenzie.

Každý záznam tabuľky obsahuje položku *items*, jedná sa o metadáta recenzie, meno autora recenzie, dátum vytvorenia recenzie a autorove skóre recenzie. Kliknutím na ikonku *info* za názvom produktu, sa zobrazí tabuľka charakterizujúca všetky symboly. Význam jednotlivých ikon je zobrazený v pohľade B.6.

Po kliknutí na záznam v tabuľke recenzií sa zobrazí pohľad na samotnú recenziu. Tento pohľad je zobrazený na obrázku 4.7. Nie všetky recenzie sú analyzované všetkými modelmi. Ak recenzia nie je analyzovaná, prebehne jej analýza pri dotaze, čo trvá istú chvíľu, pretože modely nie sú načítané v grafickej pamäti, ale v RAM a klasifikácia sa vyhodnocuje pomocou procesora. Po kliknutí sa zobrazí dialóg, ktorý informuje užívateľa o prevádzaní analýzy, pričom po jej skončení sa zobrazí náhľad recenzie produktu.

V tomto pohľade je možné vidieť základné metadáta recenzie v tabuľke *Review informations*, ako meno autora recenzie, dátum vytvorenia recenzie, autorove odporúčanie. V tabuľke *Review rating* je zobrazené autorove hodnotenie recenzie a hodnotenie pomocou natrénovaného modelu a ich rozdiel. Následne pod touto tabuľkou sa nachádza zhodno-



Obr. 4.6: Pohľad zobrazenia recenzií produktu *Rowenta RO6477EA*.

tenie recenzie, v ktorom sú zvýraznené vety pomocou natrénovaného modelu bipolárnej klasifikácie sentimentu (všeobecný model), pričom sú zvýraznené príznačné slová.

Posledne v dolnej časti pohľadu sa nachádzajú sekcie pozitívnych a negatívnych recenzií. Ak všeobecný model vyhodnotil sekciu pre ako pozitívnu, sekcia dostane zelené plus, v prípade že ju vyhodnotí ako negatívnu červené plus. Negatívne sekcie obdobne. V sekcii sú zvýraznené príznačné slová obdobne, ako v zhrnutí recenzie. Tieto príznačné slová sa vytvárajú vo fázy zhlukovacích viet celej subkategórie, pomocou modelu LDA. Ak recenzia produktu nie je súčasťou zhlukovania, slová sa nezvýraznia.

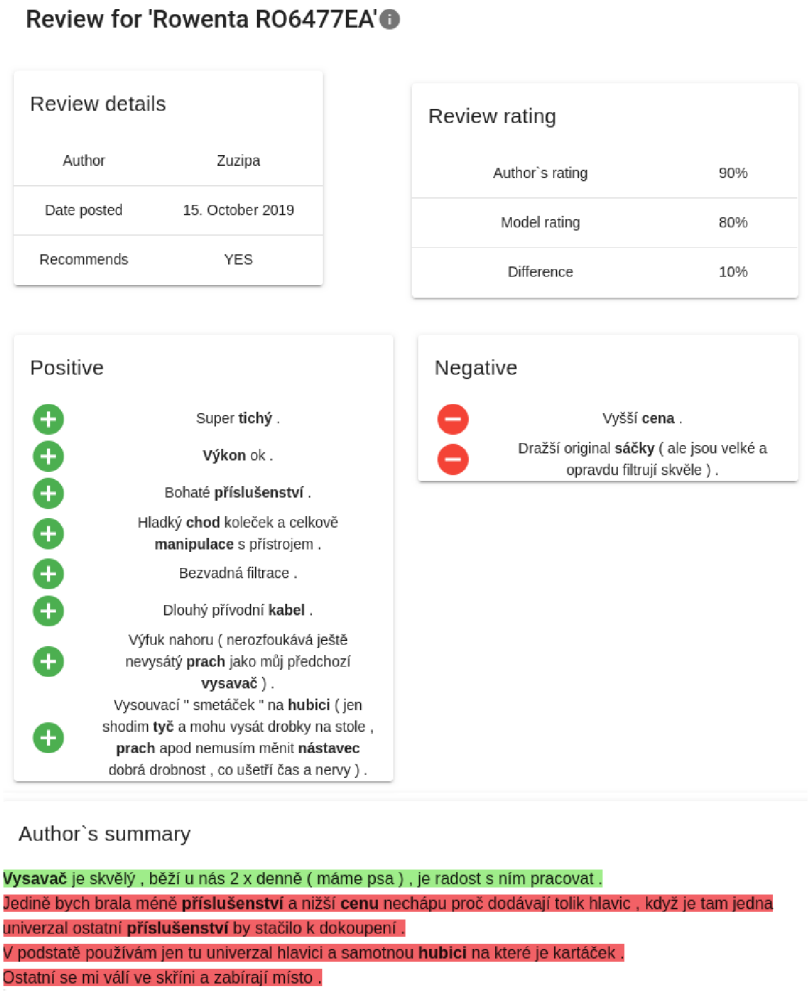
Zaujímavejšiu analýzu predstavuje pohľad na klasifikáciu položky pre/proti pomocou všetkých natrénovaných modelov klasifikácie bipolárneho sentimentu. Táto analýza je vykonaná pomocou pohľadu zobrazeného na obrázku B.4.

V tomto pohľade je možné vidieť vetu náhľad vety *Dlouhý přívodní kabel.* spolu so zobrazením výsledku bipolárnej analýzy sentimentu pomocou dostupných modelov domén. Jednotlivé výsledky klasifikácie odhaľujú doménovo príznačné vyjadrenia. Analýza recenzie je vykonaná v sekcii Rozbor príkladu recenzie 5.5.

Generovanie datasetu

Exportovanie dát indexovacieho systému elasticsearch je využité hlavne vo fázach trénovania modelov. Jedná sa o vhodnú funkcionalitu využitú pri tvorbe systému analýzy recenzií. Tento pohľad je zobrazený na obrázku B.5. Tento pohľad je implementovaný pomocou komponenty `GenerateDataset`. Je nutné dodať, že všetky datasety analýzy recenzií boli vytvorené pomocou tejto funkcionality.

V tomto pohľade je možné si pomocou stromovitej štruktúry cesty hlavných domén na kategórie zvoliť príslušné dátové sady. Na obrázku je zvolená doména *filmy_knihy_hry*, pričom je zvolených 9 podkategórii. V pravej časti pohľadu je zobrazený počet zvolených podkategórii, typ úlohy na generovanie dát, či sa jedná o prosté vety, bipolárnu klasifikáciu,



Obr. 4.7: Pohľad zobrazenia recenzie produktu *Rowenta RO6477EA*.

regresívnu úlohu. Ďalej typ modelu, napríklad Bert, formátovanie vety a možnosť voľby vyrovnaného datasetu. Táto voľba je potrebná, pretože v datasete prevláda počet pozitívnych recenzií.

Pretože na celý dataset recenzií nie je aplikovaný model filtrovania irelevantných recenzií, tak v rámci formulára sa nachádza aj ponuka zvolenia minimálnej a maximálnej dĺžky vety. V pohľade je zvolený interval $[2, 24]$. Minimálna dĺžka vety zabraňuje výskytu jednoslovných recenzií. Následne po stlačení tlačítka *Generate* sa zobrazí dialóg oznamujúci prebiehajúce generovanie datasetu. Po jeho skončení sa automaticky stiahne dataset vo formáte archívu *zip* alebo sa zobrazí chybová hláška.

Aktualizácie

Po každej aktualizácii systému sa do elasticsearch poznačia aj štatistiky aktualizácie. Tento pohľad je zobrazený na obrázku 4.1, implementovaný komponentou `ActualizeElastic`.

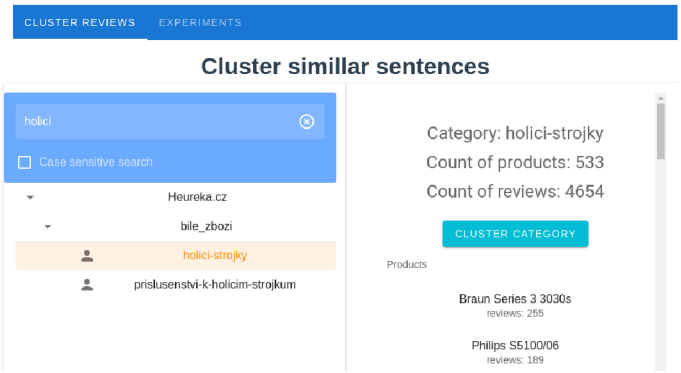
Po aktualizácii všetkých kategórii v rámci domény sa do elasticsearch uložia štatistiky o počte recenzií *reviews in time*, o počte nových produktov *new products in time*, počet nových recenzií *new products reviews in time* a nakoniec o počte ovplyvnených produktov.

Tento pohľad obsahuje menu na prehľadávanie štatistík aktualizácie pomocou jednotlivých domén produktov alebo zhrnutie celej aktualizácie v položke *All product domains*.

Podobnosť viet

Zistenie preberaných tém, aspektov produktov, respektíve kľúčových slov alebo príznačných termínov pre danú kategóriu, alebo produkt je vykonané pomocou zhlukovanie podobných viet. Systém je schopný na základe vetných vektorových reprezentácii textu zoradiť podobné vety do zhlukov, v zhlukoch nájsť bližšie témy. Z týchto tém extrahovať príznačné slová pre pozitívnu/negatívnu sekciu. Prístup k tejto funkcionalite má len užívateľ s minimálnou úrovňou práv analytik. Pohľad spustenia experimentu je implementovaný pomocou komponenty `ClusterReviews`.

Výber kategórie, ktorá bude predmetom experimentu podobnosti viet je zobrazený na pohľade 4.8. Výber je obdobný ako pri prechádzaní kategórii domén v prípade prezerania produktov. Po kliknutí na tlačítko *cluster category* sa zobrazí dialóg, pomocou ktorého sa nastavia parametre experimentu, pričom predmetom experimentu budú recenzie celej kategórie. V prípade kliknutia na produkt budú predmetom experimentu recenzie daného produktu a nie celá kategória.



Obr. 4.8: Pohľad výber kategórie „holíci–strojky“ pri experimente s podobnosťou viet.

Dialóg zhlukovacieho experimentu je zobarzený v pohľade B.7. V menu zhlukovanieho experimentu je možné vybrať metódu reprezentácie textu z možností *fse_dist*, *fse_sim*, *fse_vec*. Jedná sa o výpočet vektorových reprezentácii textu pomocou váhovej funkcie *SIF*, s výberom prostých vektorov, matice vzdialenosti vsetkých viet, matice podobnosti všetkých viet. Následne model reprezentácie textu fasttext buď predtrénovaný, alebo vytvorenie modelu na základe vstupných viet, pričom vektoru každého slova ma dimenziu 300. Výber zhlukovacej metódy, zatiaľ metóda K–means. K–means potrebuje dopredu počet zhlukov, teda výber počtu pozitívnych a negatívnych zhlukov. Nakoniec počet tém pre každý zhluk, kvôli detailnejšej analýze a nájdení príznačných slov pre sekciu plusov/mínusov.

Po stlačení tlačítka *Peek similarity* sa zobrazí dialóg s počtom viet pre plusy/mínusy. Stlačením tlačítka *Cluster similarity* sa vykoný experiment. Zobrazí sa dialóg s informáciami o prebiehajúcom experimente, po jeho skončení sa v dolnej časti pohľadu zobrazia zhluky a štatistiky.

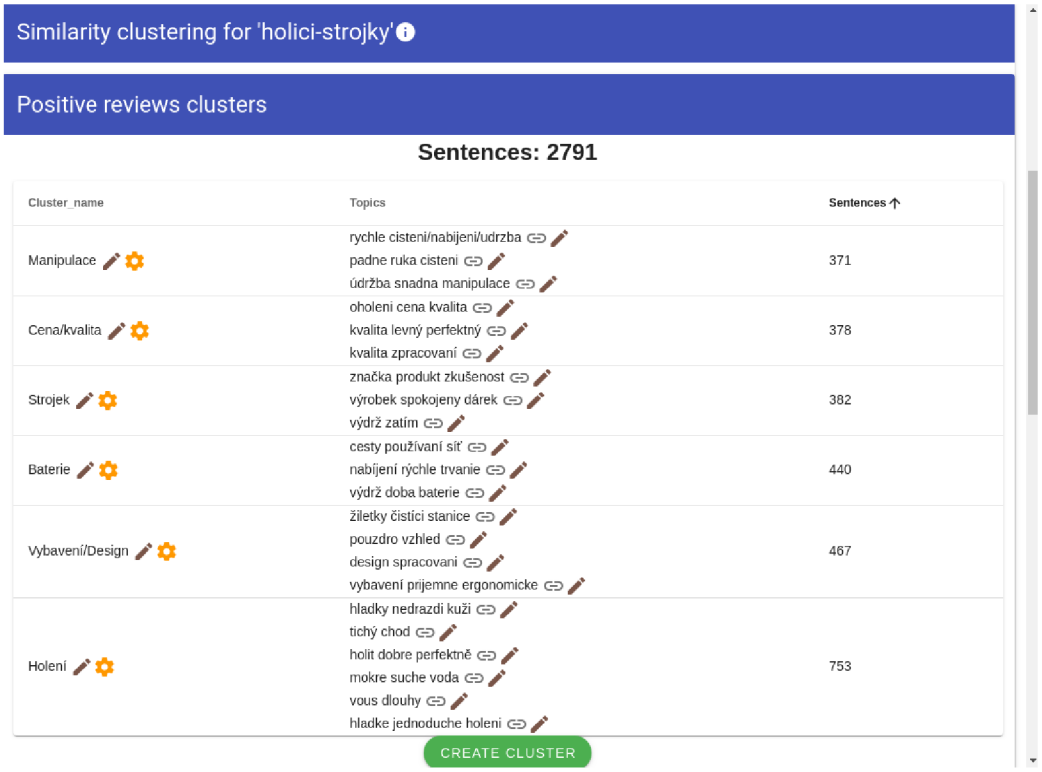
Pomocou záložkového menu je možné sa prepnúť na zobrazenie vykonaných experimentov. Centrum pohľadu je jednoduchá tabuľka s jednotlivými záznamami o prevádzaných experimentoch. Po kliknutí na položku tabuľky sa zobrazí dialóg s popisom experimentu, ktorý je implementovaný pomocou komponenty `ExperimentClusterEditView`. V dolnej

časti pohľadu potom výsledné zhluky pre sekciu plusov a mínusov a ku každej sekcii je zobrazený histogram počtu viet medzi jednotlivými zhlukmi. Zobrazenie zhlukov je opäť charakterizované pomocou tabuľky.

Príklad sekcie plusov je zobrazený na pohľade 4.9. Jedná sa o upravený zhlukovací experiment užívateľom. Systém dokáže utriediť vety do zhlukov a následne tém. Mená tém dokáže vygenerovať podľa LDA. Ponúka rozhranie na úpravu jednotlivých zhlukov. Histogram je možné vidieť v pohľade B.8. Kliknutím na ikonu *hnedého pera*, či už pri zhluku alebo téme sa zobrazí editovací dialóg. V tomto dialógu je možné upravovať názvy zhlukov/tém, spájať jednotlivé zhluky/témy, vytvárať nové témy v zhlukoch. Dialóg pre zhluk je zobrazený na pohľade B.9, pre tému na pohľade B.10.

Po kliknutí na ikonu *ozubeného kolieska* sa zobrazí dialóg s náhľadom viet zhluku s možnosťou premiestňovania viet, tento pohľad je zobrazený na obrázku B.11. Klinutím na ikonu *merge* sa otvorí dialóg s vybratím zhluku a výslednej témy, do ktorej bude veta presunutá. Kliknutím na ikonu *link* pri názve témy sa zobrazí pohľad na všetky vety odpovedajúce danej téme zhluku.

Zobrazenie výsledkov experimentu zhlukovania viet pre bežného užívateľa je možné napríklad v pohľade 4.4 kliknutím na tlačítko *Cluster Experiment*. Výsledok zhlukovania recenzií produktu je možné vidieť v dolnej časti pohľadu produktu, zobrazenom na obrázku B.12. Po kliknutí na názov zhluku, respektíve na názov témy sa zobrazí pohľad na zoznam viet prislúchajúc danému zhluku alebo téme.



Obr. 4.9: Upravený výstup experimentu zhlukovania negatívnych viet kategórie „holící-strojky“.

Demo

Fakt, že všetky recenzie nie sú analyzované pomocou natrénovaných modelov spôsobuje potrebu ich mať načítané a pripravené v rámci analýzy recenzií. Vedľajšie využitie spočíva v jednoduchom „deme“ modelov, využitie formulárov pre napísanie vety a následná analýza pomocou vybraného modelu. Takéto demo je zobrazené v pohľade B.13. Tento pohľad je implementovaný pomocou komponenty `PosConSentenceDemo`.

V tomto pohľade je možné zadať vetu, z príslušného selektoru vybrať klasifikačný model bipolárneho sentimentu a následne pomocou tlačítka *Check polarity* sa vyhodnotí zvoleným modelom príslušný text. V hornom záložkovom menu sú zobrazené ďalšie dve demá. Jedná sa o predpoveď skóre textu, implementovaného pomocou komponenty `TextRatingDemo` a demo irelevantného modelu, ktorý klasifikuje zadaný text, implementovaného komponentou `IrrelevantDemo`.

4.10 Možné vylepšenia

V tejto sekcii uvediem problémy, ktoré vznikali počas vývoja a navrhne ďalšie možné smerovania práce s cieľom na využitie viacerých modelov.

Problémy

Počas vývoja systému som sa stretol s istými nedokonalosťami indexačného systému elasticsearch. Systém závažne zaťažuje disk. Systém bol nasadení na počítači *athena18*, ktorý patrí výskumnej skupine *KNOT*. V letnom semestri po zapojení všetkých natrénovaných modelov a následných pokusoch o znova indexovanie dát s využitím modelov som sa viac krát ocitol v probléme, že elasticsearch pri čítaní dát z disku mal veľkú odozvu a odmietal indexovať dáta. Následne dáta boli poškodené a musel som celý systém obnovovať z poslednej zálohy, pričom samotná indexácia trvala istý čas. Systém som premiestnil na počítač *pcknot5*, kde systém zatiaľ funguje.

Administratíva

Elasticsearch ponúka architektúru založenú na viacerých uzloch, pričom dáta môžu byť distribuované medzi viacero uzlov vrámci zhluku, pričom sa zaistí redundancia dát v prípade zlyhania jedného uzla. Teraz celý idnexačný systém využíva práve jeden zhluk. S touto myšlienkou súvisí aj využitie administratívnej funkcionality pri pohľade na indexy elasticsearch.

Reindexácia

Aplikácia by mala zobrazovať spracované dáta bez veľkej odozvy na požiadavku. V súčasnom stave nie sú všetky dáta analyzované a tým pádom je potrebné recenzie analyzovať pri dotaze. Túto nevýhodu som sa snažil vyriešiť spomínanou znovu indexáciou recenzií no vyústilo to k presunutí systému na iný počítač.

Integrácia ďalších dátových sád internetových obchodov, ako je napríklad alza. Tento obchod nie je zahrnutý na heuréke kvôli obchodnému sporu, pričom formát webu je obdobný, ako v prípade heuréky. Využitie natrénovaných modelov v analýze iných dátových

sád, napríklad filmová databáza csfd[10]. Zaujímavé by bolo využitie modelov na slovenčinu, respektíve začlenenie slovenskej heuréky.

Modely

Systém využíva modely, ktoré sú načítané v RAM pamäti na strane koncového servera. Jedná sa o modely Bert a SVM spolu s uSIF modelom. Lepším riešením je vytvorenie externej služby, ktorá bude obsahovať načítané modely implementované napríklad pomocou frameworku *Celery*[11]. Takýto framework by bol využitý koncovým serverom, ktorý by mu preposielal vety/text na analýzu, obdobne ako dotazy do databáze.

V prípade zhlukovania viet na základe podobnosti som využil kombináciu váhovej funkcie SIF/uSIF spolu s modelom vektorovej reprezentácie FastText. Za pokus by stálo využitie SIF/uSIF na Glove vektorových reprezentáciách. Ladenie modelu Bert na datasete predpovedania nasledujúcej vety. Takýto dataset v doméne recenzií bude zaujímavé vytvoriť. Väčšina recenzií pozostáva zo strohých vyjadrení, často sú to dvoj alebo troj slovné vety.

Ďalším smerom práce môže byť aj vytvorenie manuálne anotovaného dátového setu, čo sa trošku odkláňa od cieľa práce zhlukovania viet bez prípravy manuálnej prípravy dát. Napríklad, vytvorenie dátového setu vetných trojíc sa javí, ako najlepšia možnosť smerovania práce, spolu s využitím aktuálnych modelov, ako napríklad S-Bert. Obdobne analýza sentimentu na úrovni aspektu je ďalší potenciálny cieľ.

[10] https://www.csfd.cz/
[11] http://www.celeryproject.org/

Kapitola 5

Experimenty a vyhodnotenie systému

Súčasťou práce sú aj vykonané experimenty nad datasetom recenzií a porovnanie s dostupnými publikáciami, zameranými na analýzu sentimentu, pri využití súčasných technológií spolu s analýzou chýb.

Následné rozobratie funkcionality systému týkajúce sa irelevantných recenzií, rozbor bipolárnej klasifikácie nad príkladom recenzie a zhlukovanie viet.

5.1 Porovnanie datasetu

Výsledky bipolárnej klasifikácie sa môžu porovnať s obdobnými publikáciami, v ktorých autori skúmali analýzu sentimentu v českom jazyku. Jednou z nich je celkom aktuálna publikácia [44], v ktorej autori skúmali analýzu sentimentu z datasetov vytvorených a popísaných v publikácii [40]. Autori využili datasety *Facebook* a *Mall*. Dataset *Mall* je v prípade heuréky už obsiahnutý, pretože heuréka agreguje dáta z portálu *mall.cz*.

Autori využili jednoduchú reprezentáciu textu do vektrorového priestoru pomocou vektorizéru *Tf–Idf*, opísaného v sekcii 2.5. V publikácii experimentovali s nájdením najvhodnejších argumentov vektorizéru, ktorými sú napríklad dĺžka n–gramov, použitie stop slov, normalizácia a samotných klasifikačných algoritmov. Následne otestovali všetky možné kombinácie vstupov, kvôli dosiahnutiu čo najlepšieho výsledku.

Tieto datasety a ich výsledky sú porovnané s generovaným datasetom heuréky. Štatistiky datasetov sú zobrazené v tabuľke 5.1, v ktorej je možné vidieť, že oba datasety publikácie [44] sú nevyvážené, zatiaľ čo datasety analýzy výrobkov sú vyvážené a v prípade datasetu *Biela technika* aj niekoľko násobne objemnejšie. Netreba zabudnúť zdôrazniť, že tieto datasety sú generované automaticky zo stiahnutých dát heureky, až na *Mall (vlastný)*, ten je využitý na porovnanie v *state of the art* metódach, pretože autori [44] nezverejnili dataset.

Pri porovnaní bol vygenerovaný dataset *vysávače*, *vysávače-mixéry*, *biela_technika* pre bipolárnu klasifikáciu sentimentu, dĺžka vety sa bola nastavená v intervale $[3, 20]$ slov na jeden záznam (riadok).

V tabuľke 5.2 sú zobrazené výsledky zaužívaných klasifikačných algoritmov ako *SVM*, *Naive Bayes*, ktoré sú implementované pomocou knižnice *scikit–learn*. Analýza výsledkov bola vykonaná pomocou skriptu `algoritmic_survey.py`.

	Dataset [44]		Dataset analýzy výrobkov			
	FB	**Mall**	**Vysávače**	**Vysávače–mixéry**	**Biela_technika**	**Mall (vlastný) [40]**
Records	11K	10K	4 378	6 968	**63 136**	20 780
Tokens	151K	105K	47 282	71 909	**643 355**	510 490
Avg len	13	10	11	10	10	**24**
Classes	2	**3**	2	2	2	2
Negative	4356	1991	2189	3484	**31 568**	10 390
Positive	7274	2587	2189	3484	**31 568**	10 390

Tabuľka 5.1: Štatistiky datasetov v porovnaní.

Manuálne pripravený dataset *Mall* [44] dosahuje najlepších výsledkov pri algoritmoch *nuSVM* a *Naive Bayes*. Pri algoritme *Random Forrest* dosahuje najlepšej presnosti manuálne pripravený dataset *Mall* [44], na druhú stranu sa jedná o nevyvážený dataset, pričom dataset *vysávače–mixéry* dosahuje lepšie F1 skóre. Obdobne sú na tom výsledky algoritmu *logistickej regresie*, kde má lepšie F1 skóre naopak dataset *Biela_technika*. Pri algoritme *SVM* dosahuje najlepšie hodnotenie dataset *Biela_technika*, je potrebné dodať, že pri testovaní úspešnosti klasifikácie pomocou vygenerovaných datasetov sa nevykonávala žiadna optimalizácia parametrov vektorizéru alebo klasifikačných algoritmov. Zaujímavé je zistenie, že najhoršie výsledky dosahuje dataset *Facebook*.

	Facebook		**Mall**		**Vysávače–mixéry**		**Biela_technika**	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
SVM	69.7	63.2	92.1	91.6	92.00	92.0	**92.21**	**92.04**
nuSVM	69.3	64.9	**92.6**	**91.4**	91.39	91.3	91.51	91.38
RF	62.7	44.2	**88.3**	83.8	86.66	**86.0**	85.17	85.00
LR	69.9	62.9	**92.8**	91.3	91.14	91.0	91.80	**92.10**
NB	67.2	57.6	**92.8**	**91.5**	90.07	90.6	90.92	90.95

Tabuľka 5.2: Porovnania datasetov pomocou bipolárnej klasifikácie sentimentu na úrovni sekcie recenzie +/-.

Zaujímavejšie je porovnanie datasetov s využitím *state of the art* modelov reprezentácií textu do vektorového priestoru. Jedným z nich sú modely založené na mechanizme pozornosti, transformer. Ich zástupcom je napríklad model *Bert*. Výsledky bipolárnej klasifikácie sentimentu na úrovni položky +/- recenzie na generovaných datasetoch a dostupného datasetu *Mall* [40] sú zobrazené v tabuľke 5.3. Zaujímavím zistením je fakt, že v rámci jednej kategórie *vysávače* dosahuje klasifikátor lepšej úspešnosti, ako pri zmiešanej kategórii *vysávače-mixery*.

Najlepšie výsledky s presnosťou 94.61% dosahuje generovaný dataset *Biela_technika* oproti datasetu *Mall*. Môže to byť aj tým, že dataset *Biela_technika* je ďaleko objemnejší a aj tým, dataset je tvorený z recenzií anotovaných priamo recenzentami a predpokladu, že do položky plus patria len kladné stránky produktu a naopak.

	Vysávače		**Vysávače-mixéry**		**Biela_technika**		**Mall**[40]	
	Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
Bert	90.30	91.03	82.96	83.19	**94.61**	**94.59**	91.57	91.82

Tabuľka 5.3: Bipolárna klasifikácia sentimentu na úrovni sekcie pre a proti celej recenzie.

Chyby klasifikácie

Pri analýze chýb bipolárnej klasifikácie sentimentu na úrovni sekcie recenzie pre a proti je potrebné zvážiť fakt, že každý človek pristupuje subjektívne a má vlastný názor na dané veci, pričom tento názor môže vyjadriť slobodne. V rámci generovaných datasetov neprevláda mechanizmus zhody anotátorov, ako v prípade publikácie [44].

Najčastejším typom chyby klasifikácie je takzvaný *false positive*, teda klasifikátor klasifikoval vetu, ktorá vyjadrujúce pozitívny sentiment z datasetu, ako negatívnu. Tiež je potrebné zmieniť, že vygenerované vety obsahujú minimálne 3 slová, tým pádom sa zamedzí vetám typu „*Nic.*“ alebo „*Zatím nic.*“. Výsledky analýzy chýb datasetu *biela technika* je možné vidieť na grafe 5.1, kde bolo manuálne analyzovaných 546 zle klasifikovaných recenzií zo sekcie +/-.

Najčastejšia kategória chyby predstavuje *irelevantné recenzie* (25.8%). Jedná sa prevažne o vety, kde recenzent neuviedol názorové slovo, ako *„Okénko do pečícího prostoru.“*, resp. sentiment vety je neutrálny alebo prevládajú vety, ako *„Zatím nemůžu posoudit, mám ji kratkou dobu.“*. Bez uvedenia názorového slova vzťahujúceho sa na danú entitu je ťažké posúdiť polaritu príspevku. Jedná sa o prevažne neutrálne vety.

Nasleduje takzvaná *chyba klasifikácie* (24.3%). Táto kategória obsahuje recenzie, ktorým klasifikátor neklasifikoval správnu polaritu. Táto kategória predstavuje zarážku, ak nebola recenzia anotovaná pomocou inej kategórie. Príkladom sú vety, ako *„Levně vyráběný nekvalitní výrobek.“*, *„Nejsou starosti s baterkami.“* Patria tu aj porovnávacie vety, kde recenzent porovnáva svoj kúpený výrobok s iným, pričom vykazuje negatívny sentiment k jednému z nich a klasifikátor tým pádom hodnotí túto recenziu ako negatívnu, napríklad *„Neuvěřitelně praktické použití 'flexizóny', s mojí starou deskou s kolečky se to nedá vůbec srovnat.“*.

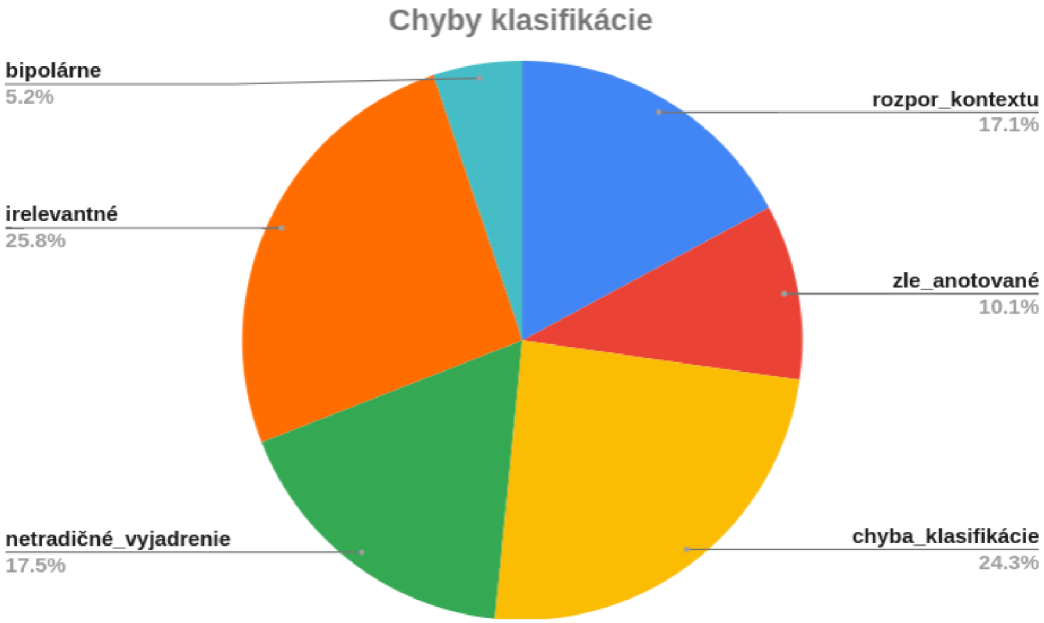
Pokračuje kategória *netradičných vyjadrení* (17.5%), kde recenzenti vyjadrujú pozitívny sentiment pomocou negatívnych slov, napríklad *„**Není** vlhkost v domě.“*, *„Uzavřená nádoba na zrnka kávy, při mletí **nevyskakují** z mlýnku ven.“*. Negatívny sentiment pomocou sarkazmu *„Dveře od mrazničky snadno otevře i naše dvouletá vnučka. chtělo by to nějakou pojistku.“*.

Ďalej kategória *rozpor kontextu* (17.1%), kde recenzent vyjadruje svoj názor, ktorý je príliš špecifický pre daný výrobok a bez väčšieho kontextu sa nedá presne rozoznať polarita vety/recenzie, napríklad *„Tichá signalizace konce programu.“* alebo *„Výdrž baterie do tří let.“*.

Predposledná kategória *zle anotovaných* recenzií (10, 1%), kde medzi pozitívne uvedenými recenziami sa vyskytujú recenzie, ako *„Trosku hlucny, ale asi jak vsechny.“* alebo medzi negatívne sú uvedené vety ako *„Nic je to dobrý pytel.“*

Poslednou kategóriou sú *bipolárne recenzie* (5.2%), v ktorých recenzent uviedol do jednej sekcie +/- všetko, resp. vyjadril v nej zmiešaný postoj. Napríklad *„Az na horsi zobrazeni desetin st. c v objektu vse v pohode.“*

Vygenerované datasety obsahujú vety s aspoň troma slovami, čím filtrujú nezmyselné vety typu *nic*, *ok*. V datasete sa často vyskytujú vety typu *„Daný výrobok máme krátko.“* alebo *„Nic mi nenapadá.“*, ktoré sa vyskytujú prevažne v sekcii mínusov recenzie. Tieto vety spôsobujú šum.



Obr. 5.1: Graf reprezentujúci rozloženie chýb pri klasifikácii bipolárneho sentimentu z datasetu biela technika.

5.2 Sentiment medzi doménami

Ďalším experimentom je natrénovanie modelu na klasifikáciu bipolárneho sentimentu nad istou doménou a overenie správnosti klasifikácie nad ostatnými doménami.

V rámci tohto experimentu boli vygenerované datasety cez všetky domény produktov, s ktorými pracuje systém analýzy recenzii, pričom bola vynechaná doména obchodov. Za klasifikátor bol vybraný model Bert, ktorý bol ladený nad doménou biela technika. Výsledky je možné vidieť v tabuľke 5.4. Model dosahoval úspešnosti klasifikácie bipolárneho sentimentu nad datasetom biela technika s presnosťou 94.61% a f-skóre 94.59, spomínaného v porovnaní tabuľkou 5.3. Všetky vygenerované datasety majú rovnaký počet záznamov pre obe triedy.

Najlepšiu presnosť dosahuje klasifikátor nad doménou *dom a záhrada* a tesne nasledujú domény *oblečenie*, *detský tovar* a *elektronika*. Najnižšiu presnosť dosahuje klasifikátor v rámci domény *jedlo*.

V prípade dostupného datasetu Mall [40] je presnosť len 76.86% a F1 skóre 71.69, pričom prevládajú chyby *false negative*. Následne bol ladený model Bert nad datasetom Mall [40]. Pri klasifikácii bipolárneho sentimentu datasetu *biela technika* dosahoval model presnosti 78.42% a F1 skóre 80.11. V tomto prípade prevládajú chyby false positive.

5.3 Predpovedanie skóre hodnotenia

Predpoveď skóre sentimentu z vygenerovaného datasetu *biela technika*, je možné vidieť v tabuľke 5.5, kde je okrem štatistik zobrazená aj stredná štvorcová chyba (*mean squared error*) pri predpovedi skóre. Bol využitý regresný model Bert. Jednotlivé hodnoty skóre sú nerovnomerne rozdelené, skóre 100% má viacej recenzií ako zbytok dokopy.

dataset	záznamov	Acc	F1
Šport	50 358	91.99	91.90
Elektronika	114 676	92.51	92.48
Filmy, knihy, hry	43 364	89.67	89.49
Dom a záhrada	47 252	**92.67**	**92.66**
Chovateľstvo	43 150	89.90	89.82
Auto-moto	11 956	92.19	92.18
Detský tovar	44 662	92.56	92.53
Kozmetika	77 646	89.50	89.29
Hobby	55 526	90.09	90.02
Jedlo	9 008	87.22	86.46
Stavebniny	5 232	92.12	92.15
Sexuálne pomôcky	9 922	89.62	89.61
Oblečenie	3 604	92.64	92.59
Mall [40]	20 780	76.86	71.69

Tabuľka 5.4: Klasifikácia bipolárneho sentimentu na úrovni sekcie pre a proti pomocou natrénovaného modelu Bert nad doménou biela technika.

Pri testovaní modelu bolo 20% dát každej kategórie skóre rozdelených do testovacej množiny a následne pre každú kategóriu skóre (napríklad pre hodnotu skóre 70%) vypočítaná stredná štvorcová chyba.

skóre v %	záznamy	vety_za_záznam	slov_za_záznam	MSE
0.1	2415	2.01	14.39	0.265
0.2	776	2.27	16.87	0.187
0.3	594	2.84	20.29	0.188
0.4	987	2.75	20.02	0.136
0.5	2644	2.57	18.23	0.115
0.6	3546	2.56	17.72	0.078
0.7	6249	2.46	16.14	0.057
0.8	17286	2.51	16.33	0.029
0.9	28303	2.46	15.37	0.011
1.0	102472	2.098	12.45	0.004

Tabuľka 5.5: Predpovede skóre recenzie pomocou modelu Bert

Chyby regresie

Na prvý pohľad je možné si všimnúť, že s narastajúcim skóre sentimentu sa znižuje aj *mean squared error*.

Skóre 10%,20%,30% obsahuje negatívne recenzie, v ktorých autori vyjadrujú prevažne negatívny sentiment. Model má problém predpovedať nízku hodnotu skóre pri vyjadrení nevýhod výrobku k explicitným aspektom, bez vyjadrenia záporov s kombináciou sarkazmu, čo je v recenziách celkom bežné. Keďže z definície sarkazmus vyjadruje isté pohŕdanie veci, čo je príznak negatívneho sentimentu, ktorý nie je priamo vyjadrený.

Čím väčšie skóre, tým sa ľudia vyjadrujú presnejšie a popisujú relevantné aspekty produktu, neprevláda v nich sarkazmus a model dokáže skóre recenzie presnejšie predpokladať.

Samozrejme každý človek má nastavený svoj vlastný systém hodnotenia a aj naprosto negatívne hodnotenie u neho má skóre väčšie alebo rovné 50%.

5.4 Irelevantné recenzie

Irelevantné vety sú vyhodnotené pomocou natrénovaného klasifikátora, spomínaného v sekcii Klasifikácia 4.5. Ako bolo spomínané, po každej aktualizácii od nasadenia všetkých modelov sa prevádzali aktualizácie, po ktorých bol vygenerovaný súbor irelevantných recenzií, ktoré model ohodnotil značkou *irelevant*. Tento súbor bol manuálne skontrolovaný a prípadné vety, ktoré neboli podľa anotátora správne klasifikované ako irelevantné, boli opravené.

Následne sa tieto vety doplnili do datasetu pre natrénovanie modelu. Štatistiky je možné vidieť v tabuľke 5.6, ktorá predstavuje priebeh aktualizácii s využitím klasifikátora irelevantných recenzií. V posledom stĺpci tabuľky je uvedená dĺžka aktualizácie systému. Záznamy označené * označujú aktualizácie bez využitia modelu na predpoveď hodnotenia recenzie. Záznamy označené znakom X u stĺpcov *Irelevantné vety* a *Prázdne recenzie* označujú prázdne záznamy, v dobe aktualizácie sa tieto štatistiky nezaznamenávali. Po integrácii modelu predpokladajúceho ohodnotenie skóre recenzie je možné vidieť zvýšenie doby aktualizácie, pretože každá recenzia sa musí vyhodnotiť modelom Bert, ktorý je načítaný v RAM pamäti pomocou procesora a nie grafickej karty. Obdobne pred zapojením modelu na filtrovanie recenzií trvala priemerne aktualizácia okolo 5 hodín. Táto hodnota závisí aj na dĺžke odozvy k portálu *heuréka*.

Stĺpec *Chybovosť* označuje podiel nesprávne klasifikovaných viet, teda relevantných k celkovému počtu viet, ktoré klasifikátor klasifikoval ako irelevantné v rámci vygenerovaného súboru. Tento súbor obsahuje vety s aspoň dvoma slovami. Podiel týchto viet a celkového počtu irelevantných viet je zobrazený v stĺpci *Označené vety*. Vety s jedným slovom sú automaticky hodnotené ako irelevantné a nezapisujú sa do výstupného súboru. Celkový podiel irelevantných viet a relevantných viet je zobrazený v stĺpci *Irelevantné vety*. Uvedený je aj podiel prázdnych recenzií v stĺpci *Prázdne recenzie*. Jedná sa o recenzie, ktoré neobsahujú sekcie pre/proti/zhrnutie. Posledný stĺpec obsahuje záznam o dobe priebehu aktualizácie v hodinách.

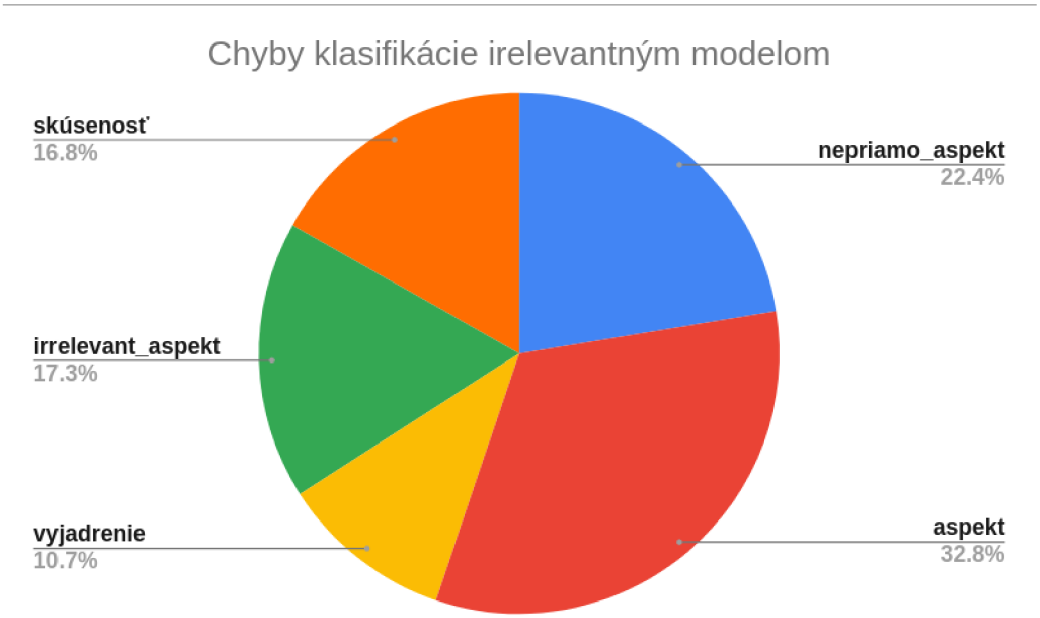
Dátum	**Chybovosť** [%]	**Označené vety** [%]	**Irelevantné vety** [%]	**Prázdne recenzie** [%]	**Čas** [*h*]
8. Apríl	14.65%	12.21%	X	X	8.94*
14. Apríl	9.98%	14.41%	X	X	5.16*
20. Apríl	9.28%	14.56%	40.73%	10.56%	18.83
27. Apríl	5.79%	14.21%	40.44%	10.43%	9.37
4. Máj	5.64%	14.14%	39.83%	10.41%	8.88
11. Máj	3.41%	14.55%	39.56%	10.43%	9.88

Tabuľka 5.6: Výsledky manuálnej kontroly výstupov aktualizácie s využitím klasifikátora irelevantných recenzií.

Ako irelevantné vety hodnotím frázy vyjadrujúce žiadnu skúsenosť s produktom. Jedná sa o vety a ich rôzne modifikácie typu „Produkt máme krátce.“, „Skoro na hodnocení.“, „Zatím jsme nic nenašli.“, „Byl to dárek.“, „Uvidíme jak sa bude dařit“, „Zatím jsem produkt ještě nepoužil.“. Ďalej vety nepopisujúce aspekty produktu a vyjadrujúce osobný pocit z produktu ako: „Jsem s produktem spokojen.“, „Funguje jak má.“, „Výrobek splnil očekávání.“ „Je Ok.“.

Analýza a chyby

Chybovosť klasifikátora sa pohybuje priemerne okolo 8.12%. Rozloženie chýb klasifikácie irelevantného modelu je zobrazené na obrázku 5.2. Graf popisuje päť tried chýb, ktoré boli vytvorené po manuálnom prejdení a následnom anotovaní výstupu aktualizácii. Dokopy bolo manuálne prejdených 1283 viet.



Obr. 5.2: Graf reprezentujúci rozloženie chýb pri klasifikácii irelevantných viet.

Najpočetnejšou kategóriou chyby je kategória *aspekt* s percentuálnym ohodnotením 32.8%. Jedná sa o vety charakterizujúce konkrétny aspekt produktu, jedná sa o priamo chybne klasifikované vety. Medzi takéto vety patria vyjadrenia, ako napríklad „Velmi úzký pas.“, „Kvalitní materiál.“, „Malý, nezabere místo.“.

V poradí druhou je kategória *nepriamo_aspekt* s percentuálnym ohodnotením 22.4%. Jedná sa o vety, ktoré nepriamo pomenúvajú aspekt produktu, ako napríklad „Trochu se rozpadají.“, „Dobře se s ním pracuje.“ „„Léčí úplně všechno.“. Ďalej vety v cudzom jazyku, ako „Hungarian languages.“, „Has 3D heating.“. Táto kategória predstavuje, obdobne ako predchádzajúca, chyby klasfifikácie.

Zaujímavá je kategória *irelevant_aspekt* s percentuálnym zastúpením okolo 17.3%. Jedná sa o výsledok spojenia irelevantných viet a ohodnotenia aspektu produktu, prípadne vyjadrenie osobného názoru na aspekt produktu spojeného s krátkym príbehom. Príkladom môžu byť vety typu „Nejlepší pánská vůně, a že jsem jich už vyzkoušel!“, „Dobrá cena, účinek zatím nemůžu zhodnotit.“, „Neznám žádnou, je skvělý za tu cenu.“.

Nasleduje kategória *skúsenosť* s percentuálnym podielom 16.8%. Vo vetách tejto kategórie recenzenti zdôrazňujú ich dlhodobú skúsenosť s produktom. Príkladom sú vety typu: *Používáme Už 20 let,naprostá spokojenost.*, „S tímto výrobkem jsem nadmíru spokojena, užívám již několik let.“, *Používám už dlouho a bez problémů.*. Časť týchto viet obsahuje slovo *spokojen*, ktoré je spojené s príkladmi irelevantných viet v trénovacom datasete. Obdobme slovo *použít*.

Najmenšie percentuálne ohodnotenie 10.7% dosahuje kategória *vyjadrenie*. Táto kategória slúži aj ako zarážka anotácie, pričom obsahuje všeobecné vyjadrenia autora recenzie, u ktorých nie je poznať k čomu sa vyjadrujú. Jedná sa o nasledujúce vyjadrenia: „Sbírám knihy o Praze.“, „Mám ráda tyčinky.“, „Mám raději velké.“.

5.5 Rozbor príkladu recenzie

Kľúčová funkcionalita systému vyplýva z počiatočnej motivácie. Urýchlenie procesu analýzy recenzií produktu s minimálnym využitím anotovaného datasetu. Filtrovaním irelevantných recenzií alebo aspoň upozornením na ne, sumarizáciu obsahu recenzií pomocou združovania viet kategórie/produktu a následnou podrobnou analýzou recenzie pomocou dostupných klasifikačných, regresívnych modelov, spomínaných v sekcii Natrénované modely 4.6.

V tejto sekcii rozoberiem recenziu vysávača produktu. Zobrazenie náhľadu recenzií produktu pomocou pohľadu 4.6 poskytuje prvotné meta dáta recenzií, ako napríklad výskyt položiek pro/proti/zhrnutie, relevantnosť recenzie pomocou modelu na predpoveď hodnotenia recenzie, aplikovanie dostupných modelov.

Irelevantné recenzie označené pomocou modelu na predpoveď skóre recenzie, s využitím značky zobrazenej na obrázku B.6, môže užívateľ ignorovať prípadne ich bližšie analyzovať. V sekcii Predpovedanie skóre hodnotenia 5.3 som uviedol príčiny chybnej predpovede skóre, pričom z výsledku plynie, že každý človek má vlastný systém hodnotenia, tým pádom sa stáva, že dosť často model ukazuje nižšie hodnotenie ako je uvedené autorom recenzie. Výnimočný prípad je situácia, kedy model vyhodnotí recenziu s vyšším ohodnotením ako predpovedal autor. Jedná sa často o recenzie, plné negatívneho textu, pričom autor im pridelí najmenší možné ohodnotenie 10%, príkladom je recenzie zobrazená na obrázku B.14 a to aj v prípade použitia regresívneho modelu, naučeného na vyrovnanom datasete, ktorý je využitý vo výslednom systéme.

Podrobnú analýzu recenzie je možné vidieť na obrázku 4.7, podobu recenzie na portály heureka je možné vidieť v kapitole *Návrh riešenia* na obrázku 3.2. Táto recenzia bola spracovaná modelom na odfiltrovanie irelevantných viet. Jej výsledok nebolo odstránenie žiadnej vety. Predpokladal som, že model odstráni položku sekcie plus *Výkon ok.*, no váha slova „výkon“ prevyšuje irelevantnú frázu „ok“. Recenzia neobsahuje iné irelevantné vety typu „Nevím o žádné.“, „Splnila naše očekávání.“ a podobne.

V analýze recenzie je možné vidieť predpoveď skóre recenzie, pričom rozdiel v ohodnotení skóre a autorovým ohodnotením je 10%. Pomocou zhlukovacieho experimentu sa extrahovali kľúčové slová vzťahujúce sa na jednotlivé témy, vygenerované pomocou LDA. Tieto slová predstavujú v istej podobe aspekty produktu. Sú zobrazené tučným písmom, teda užívateľ systému môže zbežne vidieť k čomu sa viažu jednotlivé sekcie.

Pri rýchlom pohľade na recenziu si človek všime práve hrubo zvýraznené slová v sekciách pro/proti/zhrnutie. V každej sekcii pro/proti, až na položku „Bezvadná filtrace .“ v sekcii plusov, spomína autor postoj ku niektorému z identifikovaných aspektov produktu pomocou zhlukovacieho experimentu. Jedná sa o pozitívne vyjadrenia ku aspektom *hlučnosť*, *výkon*, *príslušenstvo*, *chod a manipulace*, *kabel*, *hubica/tyč/nástavec (manipulace)* a porovnanie vysávača. Negatívne sa vyjadruje k *cene* všeobecne a ku cene *príslušenstva (sáčky)*.

V autorovom zhodnotení recenzie sú okrem zvýraznenie kľúčových slov označené aj jednotlivé vety pomocou bipolárneho modelu. Je vidieť že autor zhodnocuje prevažne negatívne postoje k aspektom produktu ako *príslušenstvo hubica* a *cena*, pozitívne sa vyjadruje k *vysavaču* ako produktu.

Výsledky klasifikácie bipolárnych modelov pri analýze negatívnych sekcií sú zhodné s výsledkom všeobecného modelu, čo sa týka viet „Vyšší cena.“ a „Dražší original sáčky (ale jsou velké a opravdu filtrují skvěle).“ Tieto vety obsahujú prevažne negatívne frázy, až na druhú vetu, ktorá obsahuje aj pozitívnu časť.

Kladná sekcia obsahuje dokopy osem viet. Veta „Super tichý.“ je klasifikovaná všeobecným modelom pozitívne, obsahuje pozitívne názorové slovo *super* spolu príznačným slovom *tichý*, ktoré vyvodzujú pozitívny sentiment naprieč všetkými modelmi. V prípade vety „Výkon ok.“ sa s výsledkom analýzy bipolárneho sentimentu všeobecného modelu líšia modely domén *oblečenie_a_móda*, *jedlo_a_nápoje* a *sexuálne_pomôcky*. V prípade domén jedla a oblečenia sa táto slovná fráza moc nevyužívajú, pričom je zaujímavý výsledok klasifikátora domény sexuálnych_pomôcok, kde sa tieto frázy vyskytujú častejšie práve v pozitívnych sekciách. Táto situácia môže byť spôsobená tokenizérom wordpiece, ktorý rozdelí slová na čo najmenšie tokeny, pričom práve tieto tokeny prikláňať pozornosť modelu k negatívnej klasifikácii. Obdobne je na tom veta „Bezvadná filtrace .“, kde sa výsledky klasifikácie líšia u doménových modelov *oblečenie_a_móda*, *jedlo_a_nápoje* a zaujímavo aj u modelu *auto-moto*.

Zaujímavejším príkladom je veta „Dlouhý přívodní kabel.“, ktorú vyhodnotili pozitívne len *všeobecný model* a model domény *dom_a_zahrada*. Model domény biela technika vyhodnotil túto vetu ako negatívnu. Aj keď fráza „Dlouhý kabel.“ sa vyskytuje v 86% prípadov v pozitívnej sekcii, presnejšie 117 krát. O slove „kabel“ hovoria recenzenti v 86% prípadoch v negatívnej sekcii, číselne sa slovo vyskytuje 911 krát. Obdobne to bude aj v prípade domén všetkých elektronických zariadení, kde recenzenti spomínajú takýto aspekt, pred pozitívnym hodnotením.

Veta „Výfuk nahoru (nerozfoukává ještě nevysátý prach jako můj předchozí vysavač).“ obsahuje prevažne negatívne názorové slova v porovnaní produktu s predchádzajúcim produktom, ktorý užívateľ používal. Jedná sa o *netradičné vyjadrenie* názoru pomocou negatívnych slov, táto kategória chyby je popísaná v sekcii chyby klasifikácie 5.1.

5.6 Zhlukovanie viet

Sumarizácia obsahu recenzií je ďalšou funkcionalitou výslednej aplikácie. Aplikácia dokáže vykonať zhlukovacie experimenty nad vektorovou reprezentáciou viet podľa zvolených parametrov. Kladie sa dôraz na využitie len vektorovej reprezentácie viet, bez využitia klasifikátora naučeného na manuálne anotovanom datasete. Ako najsľubnejšia kombinácia sa javí využitie *FastText* vektorovej reprezentácie slov v 300 dimenziálnom priestore, pričom fasttext je trénovaný na vstupných dátach. Využitie pred trénovaného modelu neponúkalo priaznivé výsledky. Vektorová reprezentácia viet z fasttext je tvorená pomocou *SIF/uSIF* váhovej schémy s využitím normalizovanej matice vzdialenosti všetkých viet. Ako zhlukovacia metóda sa osvedčil jednoduchý *Kmeans*, počet pozitívnych a negatívnych zhlukov je zhodný s číslom 8. Táto sekcia je zameraná na doménu bieleho tovaru a zhlukovací experiment nad kategóriou *holíci–strojky*.

Zastúpenie viet v pozitívnych zhlukoch je ďaleko väčšie (2791) ako v prípade negatívnych viet (760). Systém dokáže zhlukovať vety s tým že vygeneruje k zhlukom východzie názvy a názvy tém na základe charakteristických slov tém. Pomocou dostupného rozhrania na modifikáciu zhlukov, tém popísaných v sekcii *Klientsky server*, pričom som nevyužil funkcionalitu presunu vety medzi témami/zhlukmi, som identifikoval zhluky, pozitívna časť je zobrazená na obrázku 4.9 s histogramom B.8.

Percentuálne zastúpenie viet odpovedajúcim zhlukom a témam je zobrazené v tabuľke 5.7. Pôvodný počet pozitívnych zhlukov bol redukovaný na šesť. Je potrebné podotknúť, že algoritmus združuje podobné vety na základe podobných slov, pričom je použitá metóda vzdialenostnej matice viet, teda zhlukovací algoritmus združuje vety s najmenšou normalizovanou vzdialenosťou od ostatných viet.

Najpočetnejší je zhluk *Holeni*, pričom obsahuje aj najväčšie zastúpenie relevantných viet s percentuálnou hodnotou 85.52%, ktoré sa viažu ku kategórii aspektov spojených s holením. Zhluk obsahuje šesť tém. Téma *hladky nedrazdi kuži* obsahuje relevantné vety „Nedráždí pokožku.“, *Hladké oholení i na krku a bez podráždění*, téma *mokre suche voda* s vetami typu „možnost holení nasucho i ve vlhku“, „Lze mýt pod vodou.“ alebo téma *vous dlouhy* a vety typu „Poradí si i s dlouhými chlupy.“, „Netahá delší vousy.“.

Zhluk *Vybavení/Design* dosahuje najnižší obsah relevantných viet (56.10%), patria sem aspekty zaoberajúce sa vybavením, výzorom, spracovaním strojčeka. Obsahuje štyri témy, téma *žiletky čistíci stanice* dosahuje najlepšie percentuálne zastúpenie relevantných viet, nachádzajú sa v nej napríklad vety „Žiletky krásně drží.“, „Péči o strojek provádí čistící stanice.“, „čistící stanice s náplní v balení“. Téma *design spracovani* dosahuje len 42.40% obsahu relevantných viet, obsahuje vety typu „Úžasný design“, „Přepravní obal.“ a podobne.

Ako tretí v početnosti je zhluk *Baterie*, ktorý obsahuje 66.36% relevantných viet, zhluk je charakterizovaný aspektami viažúcimi sa k baterii, nabíjaniu, výdrži, dobe nabíjania. Obsahuje tri témy. Téma *nabíjení rýchle trvanie* dosahuje najnižšieho obsahu relevantných viet zhluku, obsahuje vety typu „Rychlé nabití“, „Indikátor nabití baterie“. Téma *cesty používaní síť* dosahuje najlepšiu relevantnú úspešnosť, príkladom sú vety „Výhodou je použití na síť i nabíjecí akumulátor.“, „Prakticky na cesty“.

Nasleduje zhluk *Strojek* s 68.32% obsahom relevantných viet. Zhluk je charakterizovaný skupinou aspektov viažúcich sa k produktu, strojku, značke a skúsenostiam s produktom. Nachádzajú sa v ňom tri témy, pričom téma *výrobek spokojeny dárek* obsahuje najmenší počet relevantných viet v rámci zhluku, patria sem vety typu „Dlouholetá kvalita výrobků této značky.“, „Kvalitní výrobek.“, „Zatím jsem spokojen s výrobkem a dopravou.“. Téma *značka produkt zkušenost* je charakterizovaná vetami typu „velmi dobrá zkušenost s výrobky Braun“, „Braun je velmi dobrá a spolelivá značka jsem sní spokojen.“.

Zhluk *Cena/Kvalita* dosahuje druhej najlepšej úspešnosti v počte relevantných viet (83.60%). Jednotlivé vety sa viažu k aspektom spojených s cenou, kvalitou, spracovaním. Najvyššiu percentuálnu úspešnosť dosahuje téma *oholeni cena kvalita*, ktorá obsahuje vety typu „cena a kvalita paráda“, „dobrá cena“, „kvalita oholení“.

Posledný identifikovaný zhluk predstavuje *Manipulace*. Tento zhluk obsahuje 73.58% relevantných viet, ktoré sa viažu k skupine aspektov okolo slov čistenie, údržba, manipulácia, ruka. Obsahuje tri témy, ako napríklad téma *rychle cisteni/nabijeni/udrzba*, príkladom sú vety „snadné ovládání“, „snadné čištění“, „Jednoduché čištění“.

Analýza a chyby

Analýza negatívnych zhlukov vyzerá obdobne. Zaujímavým zistením je aj identifikácia zhluku *Irrelevant*, charakterizujúci irelevantné recenzie, spomínané v sekcii 5.4. Negatívne vety kopírujú pozitívne zhluky, pričom pohľad na ne je zobrazený v obrázku B.15.

Majoritné vyjadrenia recenzentov smerujú k spôsobom holenia, charakterizovaný zhlukom *Holeni*, pričom tento zhluk sa prekrýva so zhlukom *Cena/Kvalita*, napríklad vo vetách typu „Kvalitné holení.“, pričom práve v tomto príklade sa dôraz kladie na slovo kvalitné a tieto vety sa nachádzajú práve vo zhluku *Cena/Kvalita*. Obdobne vo vetách, kde sa

zhluk	téma	zastúpenie	celkový počet viet
Holeni		**85.52%**	**753**
	holit dobre perfektně	85.50%	**269**
	hladky nedrazdi kuži	69.52%	105
	hladke jednoduche holeni	**90.47%**	84
	tichý chod	71.08%	83
	mokre suche voda	77.94%	136
	vous dlouhy	77.63%	76
Vybavení/Design		56.10%	467
	žiletky čistíci stanice	**71.23%**	73
	pouzdro vzhled	55.56%	**126**
	design spracovani	42.40%	125
	vybavení prijemne ergonomicke	48.95%	143
Baterie		66.36%	440
	nabíjení rýchle trvanie	54.11%	146
	cesty používaní síť	**69.66%**	145
	výdrž doba baterie	67.79%	**149**
Strojek		68.32%	382
	výrobek spokojeny dárek	53.80%	**158**
	výdrž zatím	57.65%	85
	značka produkt zkušenost	**69.06%**	139
Cena/Kvalita		83.60%	378
	oholeni cena kvalita	**85.05%**	**248**
	kvalita zpracovaní	54.46%	73
	kvalita levný perfektný	59.57%	57
Manipulace		73.58%	371
	rychle cisteni/nabijeni/udrzba	**86.05%**	129
	údržba snadna manipulace	54.46%	101
	padne ruka cistenii	59.57%	**141**

Tabuľka 5.7: Výsledky manuálnej kontroly zastúpenia relevantných viet identifikovaných zhlukov a tém pozitívnych viet po manuálnej analýze výsledkov zhlukovania užívateľom (pomenovanie zhlukov, zjednotenie niektorých zhlukov, premiestnenie tém alebo ich spojenie bez využitia funkcionality na premiestnenie vety do iného zhluku alebo témy).

spomína manipulácia pri holení, príkladom je téma *údržba snadna manipulace* zhluku *Manipulace* a vety „velmi dobře holeni a udržba“, „Velmi kvalitní holení“ a podobne.

V každom zhluku sa nachádzajú vety s kľúčovým slovom *holení*. Takéto slovo/slová je možné nájsť v každej kategórii produktov, obdobne pri vysávačoch by to mohlo byť slovo *vysávanie*. Fakt, že podobnosť slov sa počíta na všetky slová, teda aj napríklad prídavné mená, ako napríklad slovo „výborný“, má za následok zhlukovanie slov vzniku zhlukov s podobnými slovami, pričom tieto slová popisujú rôzne aspekty, napríklad vety „Výborná značka.“, „Výborné oholení.“, „výborný stroj.“, „Výborná kvalita za výbornou cenu.“, „Výborné parametry.“ témy *kvalita levný perfektný* zhluku *Cena/Kvalita*.

Existujú aspekty produktu, ktoré sú zdielané naprieč všetkými doménami produktov, príkladom je zhluk *Cena/Kvalita*, čo je zrejmé, recenzenti sa vyjadrujú hlavne k cene produktov spojené s typickým vyjadrením „Pomer cena kvalita“. Obdobne bude existovať zhluk *Design/Vzhľad/Produkt* ako v rozobranom experimente.

Výsledky zhlukovania sú priaznivé v predpokladaných kategóriach aspektov, ako *Holeni*, *Cena/Kvalita*. Pri myšlienke skutočnosti, že v recenziách ľudia nevyužívajú moc pestrí jazyk

a pomenúvajú priamo aspekty sa využitie tejto funkcionality dá použiť, no zásah užívateľa analytika je potrebný k dodaní sémantického významu zhlukov (pomenovanie) a následnú kontrolu. Na detailné vytvorenie zhlukov obsahujúcich priamo len dané vety sa dá použiť funkcionalita presunu viet medzi zhlukmi a témami, ktorá nebola využitá k demonštrácii výsledkov.

Výstup zhlukovania viet po manuálnej úprave môže slúžiť k príprave datasetu pre analýzu sentimentu na úrovni aspektu.

Kapitola 6

Záver

V tejto práci som sa venoval vytvoreniu systému sumarizujúceho postoje zákazníkov k produktom a obchodom. Načrtol som teoretický základ, potrebný pre zostavenie takéhoto systému, jeho návrh, vrátane očakávaných funkcionalít.

Využil som recenzie produktov a obchodov z portálu *heureka.cz*, ktoré sú indexované pomocou indexačného systému elasticsearch. Na základe tohto datasetu boli natrénované modely klasifikácie bipolárneho sentimentu, predpovede ohodnotenia recenzie a klasifikátor relevantnosti recenzie.

Systém dokáže vykonávať pravidelné aktualizácie, pričom využíva natrénované modely na odstránenie irelevantných viet a dokáže upozorniť na chybné položky recenzií a ich ohodnotenie. Počas pravidelných aktualizácii systému sa vytvoril dataset recenzií s celkovým počtom šesť miliónov štyristotisíc recenzií produktov a obchodov. Jedná sa o jeden z najväčších datasetov recenzií pre český jazyk. Výsledný systém je prevádzkovaný na serveri `pcknot5.fit.vutbr.cz`.

Tento systém môže slúžiť ako podklad ku vytvoreniu všeobecného systému analýzy recenzií a môže byť využitý priamo webovým portálom, ako je napríklad *heureka.cz* na sumarizovanie postojov zákazníkov.

Generované datasety práce boli porovnané s obdobnou štúdiou pre český jazyk, ktorej dataset bol manuálne anotovaný v úlohe bipolárnej klasifikácie sentimentu, pričom generovaný dataset dosahoval približne podobné výsledky.

Štúdia irelevantných recenzií popisuje veľký podiel takýchto recenzií na aktualizovaných dátach heuréky, jedná sa o 40% viet recenzií, ktoré sa model snaží odstraňovať pri pravidelných aktualizáciách.

V rámci analýzy recenzie domény vysávačov som načrtol analýzu recenzie pomocou dostupných klasifikátorov sentimentu. Táto analýza nesie so sebou nevýhodu načítania modelov v pamäti a dobu potrebnú na samotnú analýzu. Tieto problémy sa dajú predísť plošnou reindexáciou celého datasetu.

Analýza zhlukovania podobných viet bola vykonaná nad doménou holiacich strojčekov do jednotlivých zhlukov a tém. Napriek využitia algoritmov, ktoré nepotrebujú zásah človeka je nutná účasť užívateľa analytika, kvôli dodaniu sémantického významu zhlukom. Aplikácia ponúka rozhranie na úpravu všetkých komponent.

Analýza recenzií výrobkov predstavuje dôležitú spätnú väzbu hlavne v smere pre obchodníka, ale aj pre dátového analytika. Obchodné portály sa snažia túto analýzu čoraz ďalej automatizovať a ponúkať svojím zákazníkom rôzne výsledky.

Literatúra

[1] Allard, M. *What is a Transformer?* Inside Machine learning, Jul 2019. Dostupné z: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04.

[2] Arora, S., Liang, Y. a Ma, T. In:. Január 2019. 5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017.

[3] Bojanowski, P., Grave, E., Joulin, A. a Mikolov, T. *Enriching Word Vectors with Subword Information.* 2016.

[4] Borchers, O. *Fast sentence embeddings* [https://github.com/oborchers/Fast_Sentence_Embeddings]. GitHub, 2019.

[5] Bowman, S. R., Angeli, G., Potts, C. a Manning, C. D. A large annotated corpus for learning natural language inference. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Lisbon, Portugal: Association for Computational Linguistics, September 2015, s. 632–642. Dostupné z: https://www.aclweb.org/anthology/D15-1075. ISBN 978-1-941643-32-7.

[6] Cer, D., Yang, Y., Kong, S. yi, Hua, N., Limtiaco, N. et al. *Universal Sentence Encoder.* 2018.

[7] Cer, D. M., Diab, M. T., Agirre, E., Lopez-Gazpio, I. a Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation. *CoRR.* 2017, abs/1708.00055. Dostupné z: http://arxiv.org/abs/1708.00055.

[8] Dalal, H. a Gao, Q. Aspect Term Extraction from Customer Reviews using Conditional Random Fields. In: Dalhousie University. *DATA ANALYTICS 2017 : The Sixth International Conference on Data Analytics.* 2017. ISBN 978-1-61208-603-3.

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

[10] Dugar, P. *Transformer-Attention is all you need.* Towards Data Science, Oct 2019. Dostupné z: https://towardsdatascience.com/transformer-attention-is-all-you-need-1e455701fdd9.

[11] Ethayarajh, K. Unsupervised Random Walk Sentence Embeddings: A Strong but Simple Baseline. In: *Proceedings of The Third Workshop on Representation Learning for NLP.* Melbourne, Australia: Association for Computational Linguistics, Júl 2018, s. 91–100. Dostupné z: https://www.aclweb.org/anthology/W18-3012.

[12] Hassan Yousef, A., Medhat, W. a Mohamed, H. Sentiment Analysis Algorithms and Applications: A Survey. *Ain Shams Engineering Journal*. Máj 2014, roč. 5, č. 4, s. 1093 – 1113. ISSN 2090-4479.

[13] Hoang, M., Bihorac, O. A. a Rouces, J. Aspect-Based Sentiment Analysis using BERT. In: Hartmann, M. a Plank, B., ed. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. Turku, Finland: Linköping University Electronic Press, September–október 2019, s. 187–196. Dostupné z: https://www.aclweb.org/anthology/W19-6120. ISBN 978-91-7929-995-8.

[14] keitakurita, A. *Paper Dissected: "Deep Contextualized Word RepresentationsËxplained*. Jun 2018. Dostupné z: https://mlexplained.com/2018/06/15/paper-dissected-deep-contextualized-word-representations-explained/.

[15] keitakurita, A. *Paper Dissected: "Glove: Global Vectors for Word RepresentationËxplained*. May 2018. Dostupné z: https://mlexplained.com/2018/04/29/paper-dissected-glove-global-vectors-for-word-representation-explained/.

[16] Konopik, M., Pražák, O. a Steinberger, D. Czech Dataset for Semantic Similarity and Relatedness. In: Mitkov, R. a Angelova, G., ed. *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., September 2017, s. 401–406. Dostupné z: https://doi.org/10.26615/978-954-452-049-6_053. ISSN 2603-2821.

[17] Kumar, A. a Sebastian, T. Sentiment Analysis: A Perspective on its Past, Present and Future. *International Journal of Intelligent Systems and Applications*. September 2012, roč. 4, č. 10, s. 1–14.

[18] Lin, A. *Elasticsearch Architectural Overview*. Towards Data Science, Feb 2016. Dostupné z: https://buildingvts.com/elasticsearch-architectural-overview-a35d3910e515.

[19] Liu, B. a Zhang, L. A Survey of Opinion Mining and Sentiment Analysis. In: Aggarwal, C. C. a Zhai, C., ed. *Mining Text Data*. Sv. 1. Boston, MA: Springer US, 2012, s. 415–463. Dostupné z: https://doi.org/10.1007/978-1-4614-3223-4_13. ISBN 978-1-4614-3223-4.

[20] Ma, E. *ELMo helps to further improve your word embeddings*. Towards Data Science, Nov 2018. Dostupné z: https://towardsdatascience.com/elmo-helps-to-further-improve-your-word-embeddings-c6ed2c9df95f.

[21] Mahto, D. K. a Singh, L. A dive into Web Scraper world. In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. March 2016, s. 689–693. ISSN 0973-7529.

[22] Mikolov, T., Chen, K., Corrado, G. a Dean, J. *Efficient Estimation of Word Representations in Vector Space*. 2013.

[23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. a Dean, J. *Distributed Representations of Words and Phrases and their Compositionality*. 2013.

[24] Ni, J., Li, J. a McAuley, J. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In: University of California, San Diego. *EMNLP*. November 2019. ISBN 978-1-950737-90-1.

[25] Nigam, V. *Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning*. Towards Data Science, Dec 2018. Dostupné z: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90.

[26] Olah, C. *Understanding LSTM Networks*. Aug 2015. Dostupné z: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[27] Palachy, S. *Document Embedding Techniques*. Towards Data Science, Oct 2019. Dostupné z: https://towardsdatascience.com/document-embedding-techniques-fed3e7a6a25d.

[28] Pennington, J., Socher, R. a Manning, C. Glove: Global Vectors for Word Representation. In: Moschitti, A., Pang, B. a Daelemans, W., ed. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing EMNLP*. Doha, Qatar: Association for Computational Linguistics, Október 2014, s. 1532–1543. Dostupné z: https://www.aclweb.org/anthology/D14-1162. ISBN 978-1-937284-96-1.

[29] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C. et al. *Deep contextualized word representations*. 2018.

[30] Purnachandra Rao, B. a Nagamalleswara Rao, N. HDFS Logfile Analysis Using ElasticSearch, LogStash and Kibana. In: Krishna, A., Srikantaiah, K. a Naveena, C., ed. *Integrated Intelligent Computing, Communication and Security*. Sv. 771. Singapore: Springer Singapore, 2019, s. 185–191. Dostupné z: https://doi.org/10.1007/978-981-10-8797-4_20. ISBN 978-981-10-8797-4.

[31] Reimers, N. a Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019.

[32] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*. 1. vyd. 2015, roč. 61, č. 1, s. 85 – 117. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0893608014002135. ISSN 0893-6080.

[33] Sieg, A. *FROM Pre-trained Word Embeddings TO Pre-trained Language Models-Focus on BERT*. Medium, Nov 2019. Dostupné z: https://medium.com/@adriensieg/from-pre-trained-word-embeddings-to-pre-trained-language-models-focus-on-bert-343815627598.

[34] Sieg, A. *Text Similarities : Estimate the degree of similarity between two texts*. Nov 2019. Dostupné z: https://medium.com/@adriensieg/text-similarities-da019229c894.

[35] Silva De Moura, E. Text Indexing Techniques. In: Liu, L. a Özsu, M. T., ed. *Encyclopedia of Database Systems*. Sv. 1. New York, NY: Springer New York, 2018, s. 4084–4088. Dostupné z: https://doi.org/10.1007/978-1-4614-8265-9_1135. ISBN 978-1-4614-8265-9.

[36] Steinberger, J., Brychcín, T. a Konkol, M. Aspect-Level Sentiment Analysis in Czech. In: *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Baltimore, Maryland: Association for Computational Linguistics, Jún 2014, s. 24–30. Dostupné z: https://www.aclweb.org/anthology/W14-2605. ISBN 978-1-941643-11-2.

[37] Sun, C., Qiu, X., Xu, Y. a Huang, X. *How to Fine-Tune BERT for Text Classification?* 2019.

[38] Thomas, D. M. a Mathur, S. Data Analysis by Web Scraping using Python. In: *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. June 2019, s. 450–454. ISSN null.

[39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need*. 2017.

[40] Veselovská, K. *Sentiment analysis in Czech*. Praha, Czechia: ÚFAL, 2017. Studies in Computational and Theoretical Linguistics, sv. 16. ISBN 978-80-88132-03-5.

[41] Williams, A., Nangia, N. a Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jún 2018, s. 1112–1122. Dostupné z: https://www.aclweb.org/anthology/N18-1101. ISBN 978-1-948087-27-8.

[42] Wood, J., Tan, P., Wang, W. a Arnold, C. *Source-LDA: Enhancing probabilistic topic models using prior knowledge sources*. 2016.

[43] Zhang, Y. *Cosine similarity = dot product for normalized vectors*. Jun 2018. Dostupné z: https://medium.com/@zhang_yang/cosine-similarity-dot-product-for-normalized-vectors-c07bdb61c9d1.

[44] Çano, E. a Bojar, O. Sentiment Analysis of Czech Texts: An Algorithmic Survey. *Proceedings of the 11th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications. 2019. Dostupné z: http://dx.doi.org/10.5220/0007695709730979.

Príloha A

Obsah pamäťového média

Priložené pamäťové médium obsahuje zdrojový kód a materiály, ktoré boli v rámci diplomovej práce vytvorené. Nižšie uvedená štruktúra uvádza dôležité adresáre a súbory nachádzajúce sa v pamäťovom médiu.

```
/
├── review_analysis – systém analýzy recenzií
├── review_analysis-backend – koncový server
├── review_analysis-frontend – klientsky server
├── thesis – zdrojové súbory technickej správy pre LaTeX
├── elasticsearch--backup.tar.gz – záloha dat elasticsearch pripravená na nasadenie
├── poster.pdf – plagát
├── readme.txt – popisuje obsah nosiča a spôsob nasadenia systému
├── thesis.pdf – technická správa vo formáte PDF
└── thesis_print.pdf – technická správa vo formáte PDF určená pre tlač
```

Adresár projektu review_analysis vyzerá následovne:

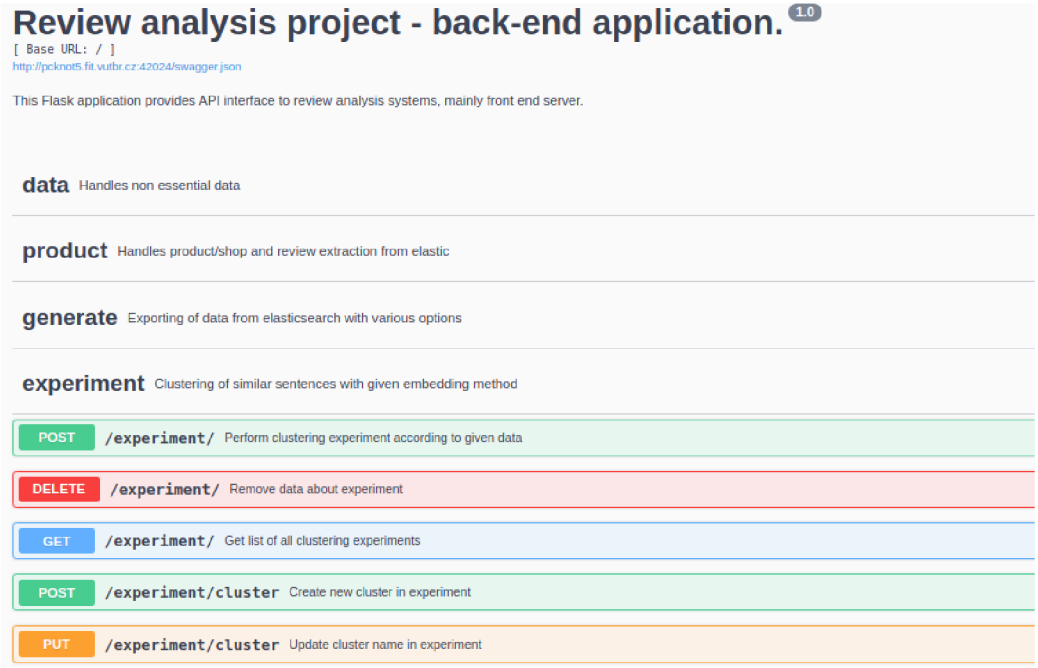
- /
  - `review_analysis` – systém analýzy recenzií
    - `classification` – moduly pre klasifikáciu a trénovanie modelov
    - `heureka_models` – modely využité pri aktualizácii recenzií
    - `experiments` – skripty použité pri vyhodnotení systému
    - `utils` – moduly zaisťujúce utility
    - `README.md` – základné informácie a návod na použitie
    - `heureka_crawler.py` –implementácie pavúka
    - `heureka_index.py` – skript na indexáciu produktov heuréky
    - `crawler.sh` – skript využívaný pri automatickej aktualizácii
    - `init_elastic.py` – inicializácia elastic indexov
    - `requirements.txt` – závislosti na balíčkoch

Samotné adresáre *review_analysis–backend* a *review_analysis–frontend* sú popísané následovne.

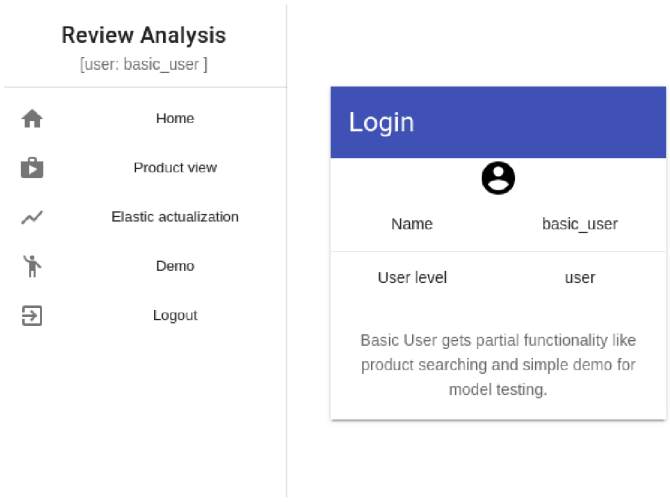
- `/`
  - `review_analysis-backend` – koncový server
    - `app` – zdrojové kódy
      - `controllers` – kontroléry pre routy API
      - `models.py` – model užívateľa
      - `routes.py` – definícia koncových bodov
    - `app.py` – spúšťací skript pre flask
    - `README.md` – základné informácie a návod na použitie
  - `review_analysis-frontend` – klientsky server
    - `src` – zdrojové kódy
      - `assets` – obrázky
      - `components` – implementované komponenty
      - `router` – služby pre komponenty
      - `App.vue` – koreňová komponenta
      - `main.js` – inicializácia koreňovej komponenty
    - `package.json` – závislosti javascriptových knižníc
    - `README.md` – základné informácie a návod na použitie

Príloha B

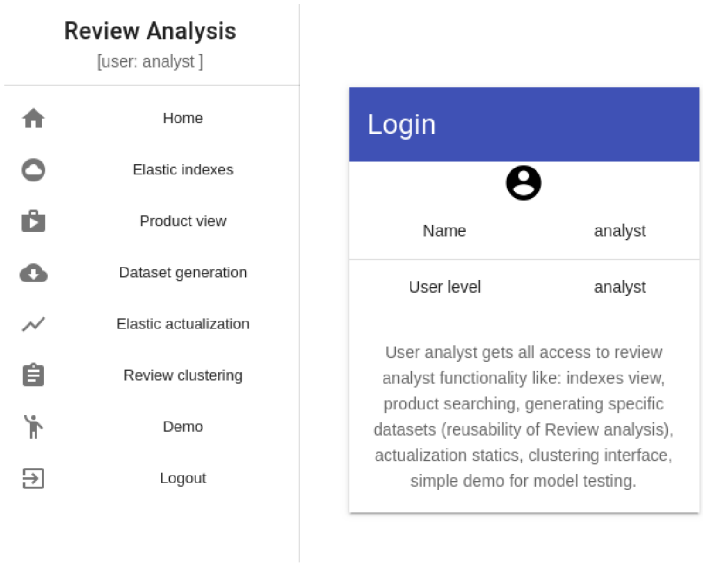
Obrázky



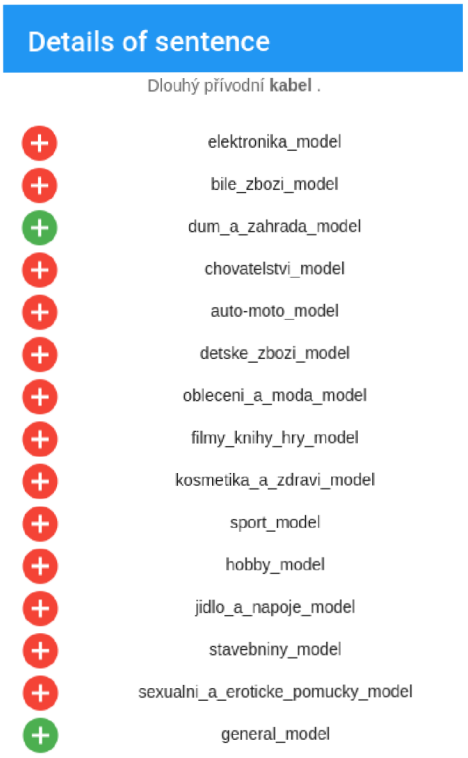
Obr. B.1: Pohľad na zobrazenie generovanej dokumentácie API pomocou nástroja swagger.



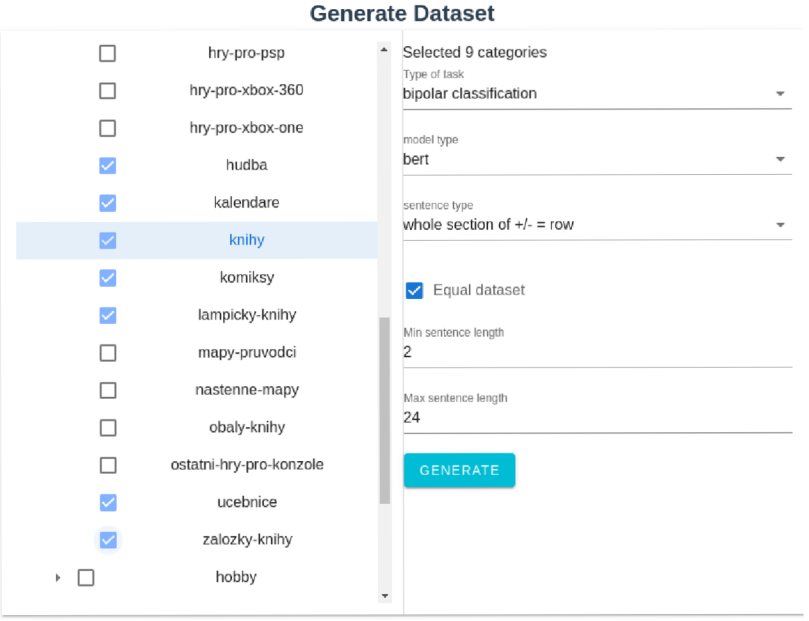
Obr. B.2: Pohľad na domovskú stránku bežného užívateľa.



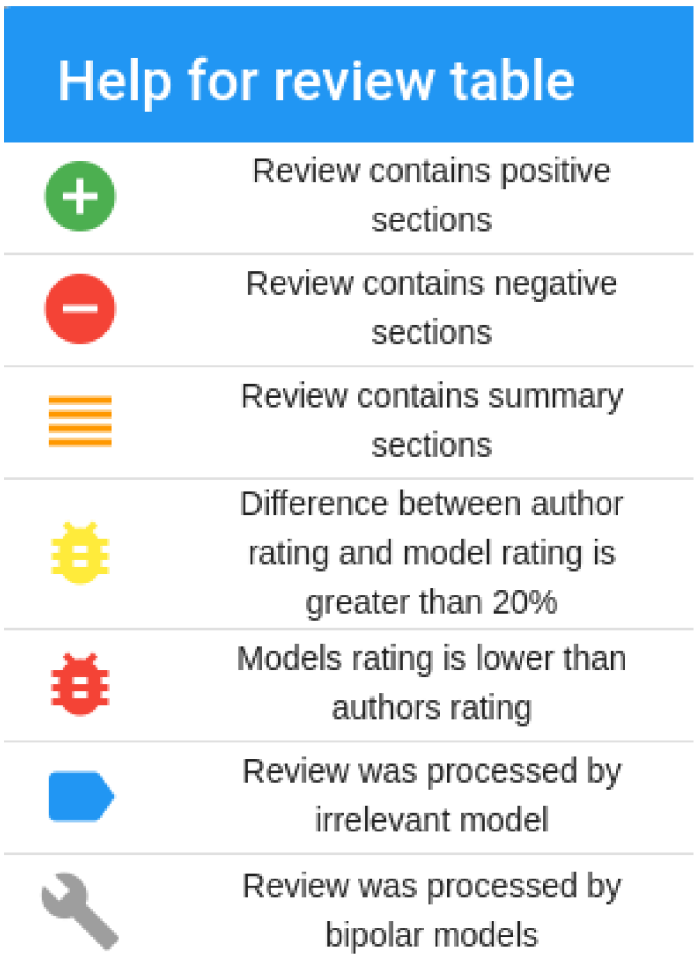
Obr. B.3: Pohľad na domovskú stránku pokročilého užívateľa.



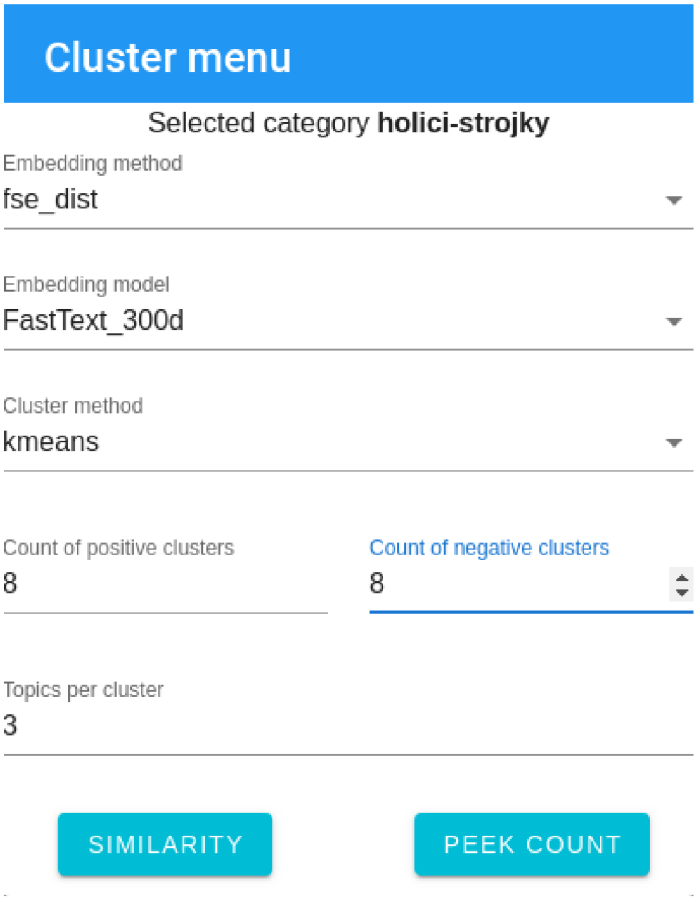
Obr. B.4: Pohľad zobrazenia analýzy sekcie pro recenzie produktu *Rowenta RO6477EA*.



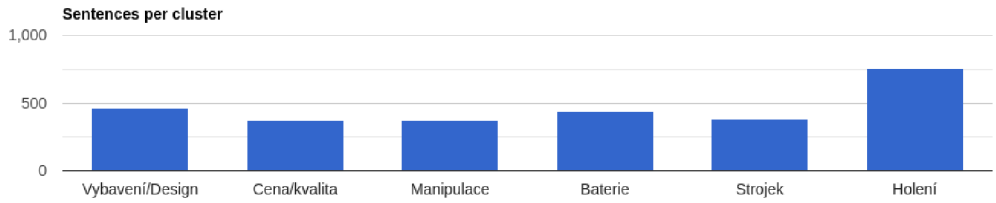
Obr. B.5: Pohľad zobrazenia generovania datasetu.



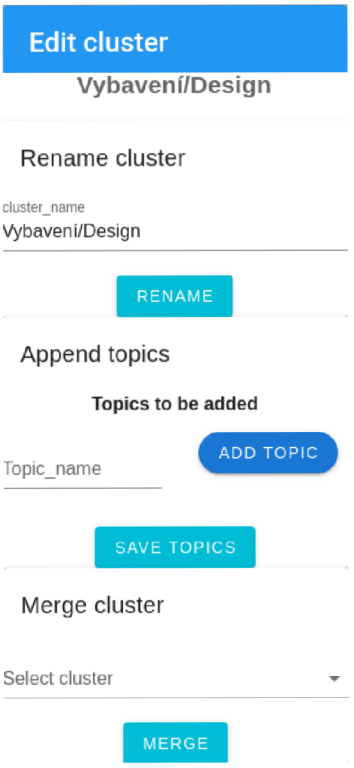
Obr. B.6: Pohľad zobrazenia nápovede, viažúci sa k tabuľke recenzií.



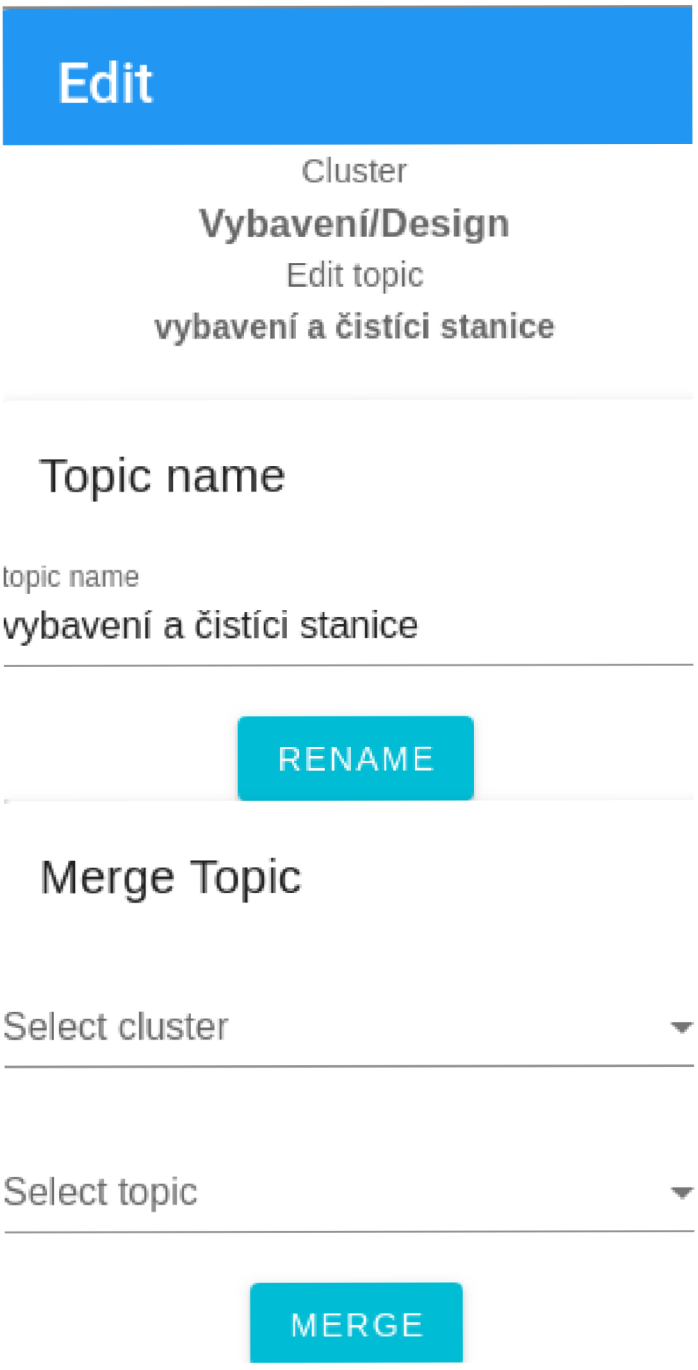
Obr. B.7: Parametre zhlukovacieho experimentu kategórie „holíci-strojky“.



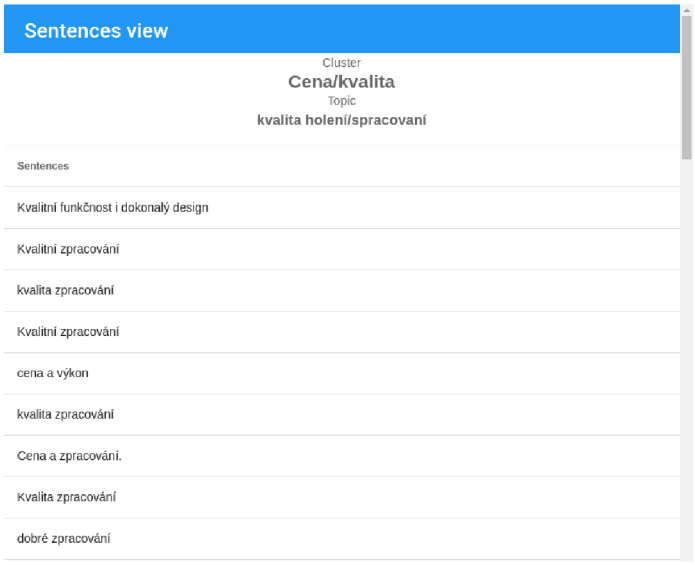
Obr. B.8: Histogram zastúpenia viet medzi jednotlivými pozitívnymi zhlukmi kategórie „holíci–strojky“.



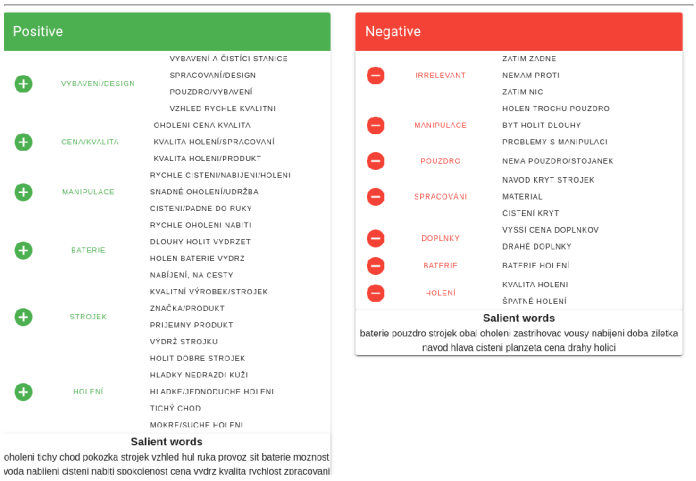
Obr. B.9: Pohľad na editáciu zhluku kategórie „holíci–strojky“.



Obr. B.10: Pohľad na editáciu témy zhluku kategórie „holíci–strojky“.

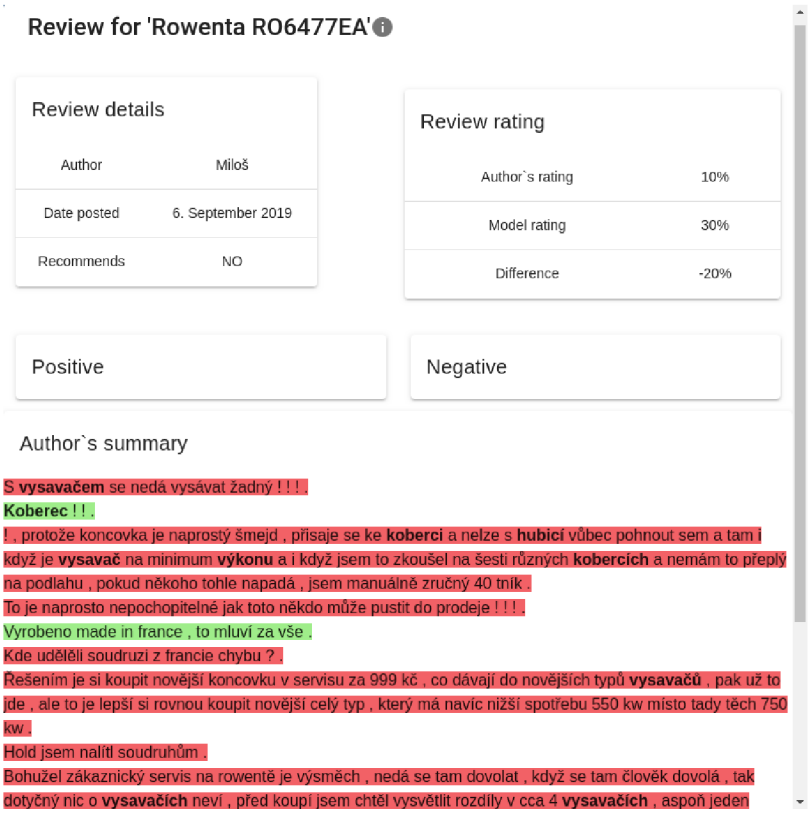


Obr. B.11: Pohľad na rozhranie pre presun vety zhluku „holíci–strojky“.

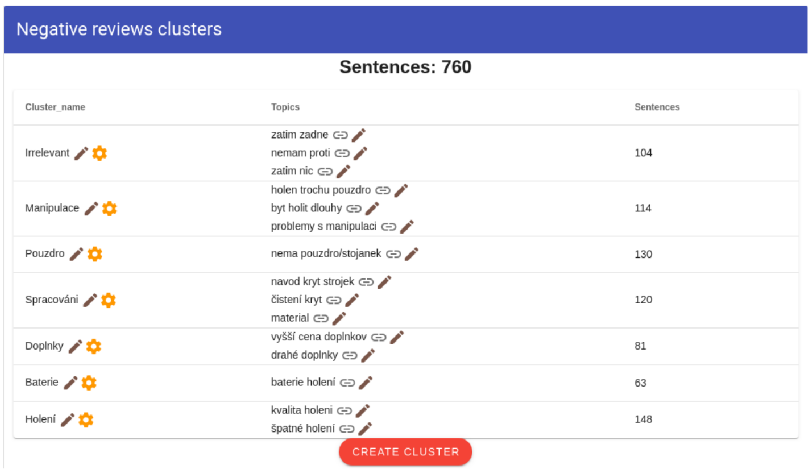


Obr. B.12: Pohľad na výsledok zhlukovania viet „holíci-strojky“ pre bežného užívateľa.

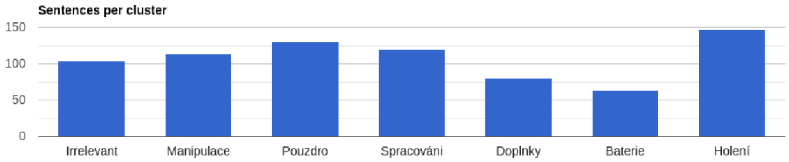
Obr. B.13: Pohľad na demo bipolárneho modelu.



Obr. B.14: Pohľad na recenziu, kde regresný model predpokladal vyššie ohodnotenie ako autor recenzie.



Obr. B.15: Upravený výstup experimentu zhlukovania negatívnych viet kategórie „holící-strojky“.



Obr. B.16: Histogram zastúpenia viet medzi jednotlivými negatívnymi zhlukmi kategórie „holíci–strojky“.