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 POSTOJŮ ČESKÝCH UŽIVATELŮ K OBCHODNÍM ŘETĚZCŮM NA ZÁKLADĚ DAT ZE SOCIÁLNÍCH SÍTÍ A WEBOVÝCH DISKUSÍÍ

SENTIMENT ANALYSIS OF CZECH SOCIAL NETWORKS AND WEB DISCUSSIONS ON RETAIL CHAINS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. MICHAL BOLJEŠIK
AUTHOR

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

BRNO 2017

Zadání diplomové práce/18873/2016/xbolje00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Bolješik Michal, Bc.**

Obor: Informační systémy

Téma: **Analýza postojů českých uživatelů k obchodním řetězcům na základě dat ze sociálních sítí a webových diskusí**
Sentiment Analysis of Czech Social Networks and Web Discussions on Retail Chains

Kategorie: Algoritmy a datové struktury

Pokyny:

1. Prostudujte rozhraní služby Facebook a dalších sociálních sítí
2. Navrhněte a implementujte systém, který dokáže pravidelně získávat, indexovat a analyzovat stahovaná data
3. Vytvořte systém pro automatickou klasifikaci shromažďovaných dat, analýzu trendů a vizualizaci výsledků
4. Demonstrujte vytvořený systém na vhodně zvolených příkladech.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle dohody s vedoucím

Při obhajobě semestrální části projektu je požadováno:

- Funkční prototyp

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cieľom tejto práce je navrhnúť a vytvoriť systém analyzujúci dáta z webu, ktoré sa týkajú českých potravinových obchodných reťazcov. Implementovaný systém umožňuje automatické sťahovanie takýchto dát, analýzu ich sentimentu, prípadnú extrakciu lokalít a názvov reťazcov z dát a následné indexovanie dát. Súčasťou systému je aj webové rozhranie zobrazujúce výsledky vykonaných analýz. Prvá časť práce sa venuje rozboru extrakcie dát z webu, analýze sentimentu a indexovaniu dokumentov. Nasleduje popis návrhu systému a popis jeho implementácie. Posledná časť práce obsahuje vyhodnotenie implementovaného systému.

Abstract

The goal of this thesis is to design and implement a system that analyses data from the web mentioning Czech grocery chain stores. Implemented system is able to download such data automatically, perform sentiment analysis of the data, extract locations and chain stores' names from the data and index the data. The system also includes a user interface showing results of the analyses. The first part of the thesis surveys the state of the art in collecting data from web, sentiment analysis and indexing documents. A description of the discussed system's design and its implementation follows. The last part of the thesis evaluates implemented system.

Kľúčové slová

extrakcia dát z webu, analýza sentimentu, indexovanie, české potravinové obchodné reťazce, Python

Keywords

web scraping, sentiment analysis, indexing, Czech grocery chain stores, Python

Citace

BOLJEŠIK, Michal. *Analýza postojů českých uživatelů k obchodním řetězcům na základě dat ze sociálních sítí a webových diskusí*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Pavel Smrž, Ph.D.

Analýza postojů českých uživatelů k obchodním řetězcům na základě dat ze sociálních sítí a webových diskusí

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Doc. RNDr. Pavla Smrže, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

……
Bc. Michal Bolješik
24. mája 2017

Poďakovanie

Rád by som sa poďakoval vedúcemu práce, pánovi Doc. RNDr. Pavlu Smržovi, Ph.D., za jeho odbornú pomoc, užitočné rady a vedenie pri tvorbe tejto diplomovej práce.

Obsah

1 Úvod

Názor je jedným z faktorov, ktoré ovplyvňujú správanie človeka. A to vo všetkých oblastiach života. Naše vnímanie sveta či rozhodovanie je do istej miery podmienené názormi iných, pretože aj tie nám pomáhajú pri tvorbe vlastného pohľadu na veci, ľudí alebo udalosti. To môže platiť nie len pre postoj konkrétneho človeka, ale i pre postoj organizovanej alebo neorganizovanej skupiny ľudí.

Po prvýkrát v ľudskej histórii sa stretávame so situáciou, v ktorej je v digitálnej podobe dostupné obrovské množstvo dát so subjektívnym obsahom. Je to výsledok rozmachu sociálnych sietí, diskusných fór, osobných blogov alebo recenzií produktov a služieb. Tento jav využívajú vo svoj prospech firmy a organizácie. Skúmajú názory a spätné väzby spotrebiteľov, aby na ich základe vylepšili alebo vytvorili rôzne služby, produkty alebo reklamnú kampaň.

Cieľom tejto práce je navrhnúť a implementovať systém, ktorý by bol osožný ako firmám, tak aj spotrebiteľom, a to v oblasti českých potravinových reťazcov. Automatizoval by činnosti, ktorých manuálne vykonanie by bolo zložité, prípadne by trvalo neprijateľne dlho. Konkrétne ide o sťahovanie dát z vybraných zdrojov, o analýzu ich obsahu a indexovanie. Výsledky analýz by potom boli užívateľovi poskytnuté pomocou vhodnej vizualizácie.

Text práce pozostáva z nasledujúcich logických celkov. Kapitola 2 ponúka teoretický pohľad na zhromažďovanie dát z webu, na analýzu sentimentu v dokumentoch na rôznych úrovniach a na možné prístupy k indexovaniu dokumentov. V kapitole 3 je uvedený návrh vyvíjaného systému, na ktorý nadväzuje popisom jeho implementácie kapitola 4. Vyhodnotenie vybraných častí systému je možné nájsť v kapitole 5. V poslednej časti tejto práce, v kapitole 6, sú zhrnuté dosiahnuté výsledky a spomenuté sú i možné rozšírenia implementovaného systému.

V rámci semestrálneho projektu bol implementovaný modul pre sťahovanie dát zo sociálnej siete *Facebook* (viď podkapitolu 4.1.4). Taktiež boli vykonané kvalitatívne analýzy dát zhromaždených pomocou tohto modulu, ktoré sú základom pre voľbu analyzovaných aspektov (viď podkapitolu 4.2.5).

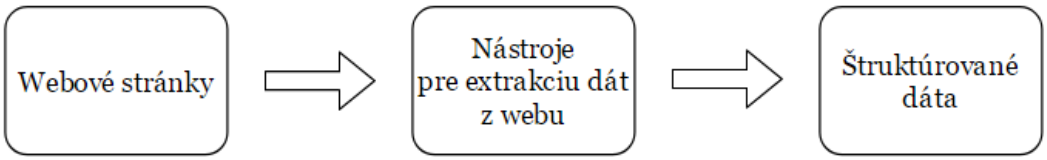
2 Rozbor riešenej problematiky

Táto kapitola ponúka rozbor oblastí, s ktorými prídeme pri návrhu systému a jeho implementácii do styku najčastejšie. Popísaná je tu extrakcia dát z webu, predstavená je problematika analýzy sentimentu, a to na rôznych úrovniach, a záver kapitoly je venovaný indexovaniu dokumentov spolu s popisom jeho jednotlivých metód.

2.1 Extrakcia dát z webu

Základným stavebným kameňom niektorých činností (napríklad výskumov názorov) sú dáta. Tie sú prístupné aj prostredníctvom obsahu webových stránok. Umiestnenia dát na rôznych webových stránkach však môžu byť rôzne. Manuálne spracovanie takýchto dát (kopírovanie a ukladanie, napríklad na lokálny disk) je v prípade väčšieho množstva dát zdĺhavým procesom.

Web scraping je podľa [1] technika extrahujúca dáta z rôznych zdrojov (webových stránok) do súborov, prípadne databáz tak, aby ich bolo jednoduché analyzovať, prípadne i vizualizovať. Proces teda zahŕňa aspoň čiastočnú transformáciu neštruktúrovaných dát (ktoré sa nachádzajú na webových stránkach) na dáta štruktúrované (tie, ktoré budú uložené v súbore alebo v databáze). Takéto extrahovanie určitého typu informácií (napríklad názorov na reklamu) môže byť potom podkladom napríklad pre systémy na podporu rozhodovania. Synonymami pre túto činnosť sú tiež pojmy *web data extraction*, *web data scraping*, *web harvesting* alebo *screen scraping*. Proces extrakcie dát z webu ilustruje Obrázok 2.1.



Obrázok 2.1: Proces extrakcie dát z webu.

2.1.1 Existujúce techniky

Techník pre extrakciu dát z webu existuje podľa [1] hneď niekoľko. Ich vymenovanie a popis ponúkajú nasledujúce odseky.

Manuálne kopírovanie dát z webových stránok (technika tiež nazývaná *Copy and Paste*) je len zriedkakedy najlepším možným riešením. Jej náchylnosť na chyby je spôsobená faktom, že k dátam pristupuje sám človek. Ide tiež o (pre človeka) stereotypnú a únavnú činnosť, obzvlášť pri veľkom množstve spracovávaných dát.

Extrahovanie dát pomocou regulárnych výrazov je jednoduchá a silná technika. Založená je na vyhľadávaní špecifických vzorov (regulárnych výrazov) v spracovávanom obsahu.

Obsah samotných stránok je získavaný prostredníctvom zasielania *HTTP* dotazov (*HTTP requests*) na vzdialený server (s prípadným využitím *socketov*). Týmto spôsobom je možné získať obsah statických i dynamických webových stránok. Dotazovacie jazyky pre semištruktúrované dáta (napr. *XQuery* alebo *Hyper Text Query Language*) sú potom základom *extrahovania dát pomocou parsovania HTML kódu*. Použité sú, ako už samotný názov tejto techniky napovedá, pre parsovanie obsahu webových stránok a následné získanie, prípadne transformáciu dôležitých informácií.

Techniku *parsovania Document Object Modelu* (ďalej už len *DOM*) reprezentujú nástroje so vstavanými (plnohodnotnými) webovými prehliadačmi. Práve vstavané prehliadače vedia zabezpečiť získanie dynamického obsahu webových stránok generovaného skriptami na strane klienta. Pomocou nich je tiež vykonané parsovanie *HTML* kódu webovej stránky na príslušný *DOM* strom.

Využitie špecializovaného *softvéru pre extrakciu dát z webu* môže byť ďalšou alternatívou. Takýto softvér sa pokúša automaticky rozpoznať dátové štruktúry na webovej stránke , prípadne obsahuje rozhranie, ktoré odstraňuje nutnosť manuálne písať kód pre extrahovanie dát. Poskytnuté tiež môžu byť funkcie na extrahovanie a transformovanie dát alebo databázové rozhranie zabezpečujúce lokálne uloženie extrahovaných dát.

Vertical aggregation platforms je anglický výraz pre nástroje vytvárajúce a monitorujúce činnosť tzv. *botov*. Táto činnosť nezahŕňa žiadnu interakciu s človekom. Robustnosť takýchto platforiem je úmerná kvalite informácií, ktoré dokážu získať, a taktiež ich škálovateľnosti.

O *technike rozpoznávania sémantických anotácií* hovoríme v prípadoch, v ktorých analyzované webové stránky obsahujú metadáta alebo sémantické značkovanie a anotácie. Tie môžu pomôcť lokalizovať konkrétne dáta. Ak sú takéto anotácie priamo obsiahnuté v *HTML* kóde webových stránok (napríklad anotácie *microformats*, viď [2]), ide o špeciálny prípad techniky *parsovania DOM*. V ostatných prípadoch sú tieto anotácie uchovávané v samostatnej vrstve oddelenej od webových stránok. Práve pomocou tejto vrstvy môžu potom nástroje pre extrahovanie dát z webu získať dátovú schému či prípadné inštrukcie ešte pred extrahovaním dát zo samotných stránok.

Existujú snahy využiť strojové učenie a počítačové videnie pre extrahovanie informácií z webových stránok. Cieľom je dosiahnuť stav, aby takto vyvinuté nástroje boli schopné identifikovať a extrahovať informácie z webových stránok pomocou vizuálneho interpretovania stránok tak, ako by to robil človek. V tomto prípade hovoríme o *analyzátoroch webových stránok využívajúcich počítačové videnie*.

2.2 Analýza postojov

Cieľom analýzy postojov (*sentiment analysis*) je odhaľovať emočné zafarbenie názorov alebo hodnotení (vytvorených ľuďmi) týkajúcich sa produktov, služieb, firiem, čí mnohých ďalších vecí. Inými zaužívanými názvami pre túto činnosť sú, napríklad, dolovanie názorov (*opinion mining*), analýza subjektivity (*subjectivity analysis*), ale aj mnohé iné (viď [3]). Tento termín sa často vyskytuje v prácach z akademickej pôdy, no čoraz viac sa začína objavovať i v podnikateľskej sfére. Dôvodom je, že výskumy na túto tému majú, okrem dopadu na oblasť spracovania prirodzeného jazyka (*natural language processing*), dopad i na ekonomiku či politické a sociálne vedy.

Analýza ľudských vyjadrení môže prebiehať na rôznych úrovniach (s rôznymi stupňami granularity). Výskumy sa ale upriamujú najmä na tri takéto úrovne. Sú nimi analýza sentimentu na úrovni dokumentu (*document-level sentiment analysis*), analýza sentimentu na úrovni vety (*sentence-level sentiment analysis*) a analýza sentimentu na úrovni aspektu (*aspect-based sentiment analysis*). Prehliadnuť by sme tiež nemali fakt, že analyzované vyjadrenia môžu byť dvojaké. V literatúre sú nazvané ako obyčajné (*regular opinions*) a porovnávacie (*comparative opinions*).

Predtým, ako si bližšie popíšeme jednotlivé typy analýz a názorov, formálne si definujme názor tak, ako je definovaný v [4]. Ide o päticu $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$, v ktorej je e_i je názov entity, a_{ij} aspekt entity e_i, oo_{ijkl} polarita názoru na aspekt a_{ij} entity e_i (môže byť pozitívna, negatívna alebo neutrálna, prípadne vyjadrená pomocou jednej z úrovní intenzity), h_k autor názoru a t_l čas, v ktorom bol názor vyslovený.

Pre extrakciu entity e_i (iba v prípade, že ide o vlastný názov, napríklad názov produktu alebo spoločnosti), autora názoru h_k a času vytvorenia názoru t_l je zaužívané spoločné pomenovanie, a tým je rozpoznávanie pomenovaných entít (*Named Entity Recognition*). Keďže v dnešnej dobe sú najdostupnejšími testovacími dátami najmä príspevky zo sociálnych sietí, rôznych fór alebo diskusií, extrakcia informácií o autorovi príspevku h_k a času jeho uverejnenia t_l je relatívne jednoduchá. Tieto informácie bývajú totiž pri takýchto príspevkoch zverejnené (i keď v prípade informácie o autorovi príspevku si nikdy nemôžeme byť istí jej pravdivosťou). O trochu náročnejšie je to pri zisťovaní entity e_i. Pre názov značky alebo produktu totiž môže existovať viacero pomenovaní (pre značku *Motorola* to môžu byť v anglických recenziách ekvivalenty *Moto* alebo *Mot*).

2.2.1 Analýza sentimentu na úrovni dokumentu

Tento typ analýzy sentimentu vníma názor ako dokument (množinu viet) a určuje, či tento dokument vyjadruje pozitívny alebo negatívny postoj. Dôležitým predpokladom pritom je, aby každý analyzovaný dokument vyjadroval stanovisko k práve jednej entite. Túto techniku teda nemožno aplikovať na dokumenty hodnotiace, prípadne porovnávajúce viacero entít. Jej nevýhodou tiež je, že neposkytuje informácie o tom, čo bolo v dokumente hodnotené kladne a čo záporne.

Uveďme si príklad. Uvažujme recenziu produktu. Analýza sentimentu na úrovni dokumentu je schopná určiť celkový (kladný alebo záporný) postoj k hodnotenému produktu.

Z formálneho hľadiska si môžeme označiť dokument ako *d* a entitu, ktorú *d* hodnotí, ako e_i. Úlohou analýzy sentimentu na úrovni dokumentu je stanoviť orientáciu názoru oo_{ijkl} na entitu e_i, pričom hodnotená je entita ako celok. To znamená, že orientácia názoru oo_{ijkl} sa bude vzťahovať k špeciálnemu aspektu a_{ij} s hodnotou *GENERAL*. Pätica reprezentujúca názor bude teda v tomto prípade $(e_i, GENERAL, oo_{ijkl}, h_k, t_l)$. Môžeme predpokladať, že názov entity e_i, autor dokumentu h_k a čas jeho vytvorenia t_l sú buď hodnoty známe alebo irelevantné. Ako už bolo naznačené vyššie, je vhodné, aby dokument *d* hodnotil iba jednu entitu e_i a aby toto hodnotenie pochádzalo od jediného autora h_k.

Väčšina existujúcich techník vykonávajúca túto analýzu je založená na strojovom učení s učiteľom (*supervised machine learning*), pričom analyzovaný názor môže byť zaradený do jednej z troch tried (pozitívne, negatívne a neutrálne názory). Použitá môže byť ktorákoľvek z existujúcich metód strojového učenia s učiteľom (napr. naivná Bayesovská metóda alebo metóda *SVM*; *SVM* je skratka pre *Support vector machines*) napríklad s unigramami (*a bag of individual words*) ako príznakmi. Najdôležitejšou úlohou je, ako pri väčšine aplikácií strojového učenia, vytvoriť čo najväčšiu množinu označených dát a určiť vhodnú množinu príznakov.

Jedným z príkladov niektorých existujúcich príznakov môžu byť *termíny a ich frekvencia*. Ide o slová, prípadne *n-gramy* slov a ich počty výskytov. V niektorých prípadoch môžu byť uvažované i pozície týchto slov. Aplikovaná tiež môže byť metóda *tf-idf*, no i bez jej použitia sa tieto príznaky ukázali ako celkom efektívne. Ďalším vhodným príznakom môže byť *slovný druh*. Výskumy ukázali, že dôležitým indikátorom názoru sú prídavné mená. Tie by teda mohli byť samé o sebe považované za príznak. Rovnako tak je efektívne uvažovať za príznak *frázy a slová*, ktoré sú často používané na vyjadrenie kladného či záporného postoja. Tými kladnými sú, napríklad, *dobrý*, *výborný*, *vynikajúci*, zatiaľ čo slová ako *zlý*, *škaredý* alebo *otrasný* indikujú záporný význam. Mnohé z nich sú prídavné mená a príslovky, no podstatné mená (*odpad*) a slovesá (*mať rád*, *nenávidieť*) môžu tiež niesť informáciu o zafarbení názoru. *Negácie* sú ďalším z príznakov. Dôležité sú z dôvodu toho, že ich prítomnosť často znamená zmenu orientácie názoru (*farba nie je pekná*), no musí sa na ne prihliadať s opatrnosťou., pretože nie vždy musia znamenať negáciu (*nie len ... ale aj*). V niekoľkých výskumoch bola tiež použitá syntaktická závislosť. Tá využíva závislosti medzi slovami generovanými z derivačného stromu (*parse tree*).

Popri používaní štandardných metód strojového učenia tiež prebehlo niekoľko výskumov za použitia techník upravených práve pre účely analýzy sentimentu. Ide, napríklad, o funkciu počítajúcu skóre (*score function*) založenú na slovách z pozitívnych a negatívnych recenzií alebo o zlepšenie presnosti klasifikácie pomocou priradenia váh atribútom (*feature weighting*).

Ručné označovanie tréningových dát môže byť časovo náročná operácia. K jej zjednodušeniu môže viesť využitie vyššie spomenutých fráz a slov špecifických pre vyjadrenie názorov. Práve tento

postup bol v jednom z výskumov, ako uvádza [4], použitý na označenie časti dát a na ich základe došlo k vytvoreniu nového klasifikátoru. Využívanie takýchto slov tiež môže viesť k zlepšeniu presnosti klasifikácie polarity.

Výskumy tiež prebehli v oblasti určovania skóre recenzií (napr. pomocou stupnice od jednej do päť). Riešený problém je v tomto prípade definovaný ako regresia, keďže sa na hodnoty skóre môžeme pozerať ako na ordinálne hodnoty.

Iným smerom sa zasa uberajú výskumy v oblasti zvanej *transfer learning* (ide o adaptáciu na jednotlivé domény - *domain adaptation*). Bolo totiž zistené (viď [4]), že úspešnosť klasifikácie polarity sa viaže na oblasť, z ktorej pochádzajú tréningové dáta. Výkon klasifikátoru vytvoreného nad dokumentmi s názormi z jednej oblasti nie je príliš dobrý, ak vykonáva analýzu dokumentov s názormi z inej oblasti. Vysvetlením môže byť, že na vyjadrenie názoru sú v rôznych oblastiach použité rôzne slová a jazykové konštrukcie. Výstižným príkladom je, keď rovnaké slovo nadobúda v jednej oblasti kladný význam a v inej záporný (môže isť, napríklad, o *nepredvídateľnú zápletku* z recenzie filmu a *nepredvídateľné riadenie* z recenzie auta, ako uvádza [5]). Jedným z riešení (postup jedného z výskumov) tohto problému by mohlo byť použitie označených dát z jednej oblasti, neoznačených dát z cieľovej oblasti a všeobecných fráz a slov vyjadrujúcich názor ako príznaku strojového učenia (viď [4]).

Pre analýzu sentimentu na úrovni dokumentu je tiež možné použiť i niektoré z metód strojového učenia bez učiteľa. Vyššie spomínané frázy a slová indikujúce názor hrajú v tomto prípade podstatnú úlohu. Je totiž prirodzené založiť strojové učenie bez učiteľa práve na nich. Jedna z metód v rámci výskumov používa známe slová vyjadrujúce polaritu, iná zasa definuje frázy, prostredníctvom ktorých bývajú väčšinou názory vyjadrované (viď [4]).

Algoritmus tej druhej menovanej by sa dal v skratke popísať nasledovne. V prvom kroku sa extrahujú frázy obsahujúce prídavné mená alebo príslovky, keďže práve tieto dva slovné druhy sú dobrými indikátormi názoru. V niektorých prípadoch prichádza do úvahy, že na zistenie polarity postačuje prídavné meno samotné, no v iných prípadoch nemusí byť zrejmý kontext, v ktorom sa toto prídavné meno nachádza (napr. prídavné meno *nepredvídateľný* v príkladoch uvedených vyššie). Práve preto sa extrahujú dvojice po sebe idúcich slov (na základe vopred definovaných vzorov obsahujúcich slovné druhy slov), z ktorých jedno je prídavné meno, prípadne príslovka.

Druhým krokom je ohodnotenie orientácie názoru extrahovaných fráz. Tento akt je založený na výpočte asociácie frázy s pozitívnym referenčným slovom *excellent* (v preklade *výborný*; ide o výskum s anglickými dátami) a s negatívnym referenčným slovom *poor* (*biedny*). Využitá je pri tom miera asociácie (*point-wise mutual information)*. Bližšie detaily tohto výpočtu opisuje [4].

V treťom kroku tento algoritmus počíta priemer orientácií názorov všetkých extrahovaných fráz v analyzovanom dokumente. Ak je výsledný priemer kladný, v dokumente prevláda pozitívne hodnotenie opisovanej entity, ak je naopak priemer záporný, hodnotenie je negatívne.

V praxi sa hodnoty úspešnosti klasifikácie pomocou tohto algoritmu odlišujú na základe domén hodnotených vzoriek dát. Pri analýze recenzií áut bola dosiahnutá správnosť 84%, pri analýze filmových recenzií zasa 66%.

2.2.2 Analýza sentimentu na úrovni vety

Analýza sentimentu na úrovni vety analyzuje každú jednu vetu z dokumentu a rozhoduje, či veta vyjadruje pozitívny, negatívny alebo neutrálny názor. Tento typ analýzy úzko súvisí s klasifikáciou subjektivity (*subjectivity classification*). Tá rozlišuje vety obsahujúce fakty od viet obsahujúcich subjektívne pohľady a názory. Tu treba poznamenať, že subjektivita a polarita sú významovo dva odlišné pojmy.

Poďme si analýzu sentimentu na úrovni vety definovať formálnejšie. Uvažujúc akúkoľvek vetu, táto analýza pozostáva z dvoch úloh. Prvou je vyššie spomínaná klasifikácia subjektivity tejto vety (určí sa, či je veta subjektívna alebo objektívna). V prípade, že ide o subjektívnu vetu, je nad ňou vykonaná druhá úloha - klasifikácia sentimentu na úrovni vety (určenie, či veta vyjadruje pozitívny, negatívny alebo neutrálny názor). V tomto prípade nemá definícia problému pomocou pätice $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ žiaden význam, pretože analýza sentimentu na úrovni vety býva často iba akýmsi medzikrokom. Väčšinou je totiž vhodné poznať cieľ, prípadne ciele, ku ktorým sa veta vyjadruje (entita alebo jej aspekt/y), pretože samotná informácia, že veta má pozitívnu alebo negatívnu polaritu, bez údaju o predmete nie je príliš použiteľná. Použiteľnou sa teda stáva v momente, keď sa viaže k širšiemu kontextu. Prvý krok vyfiltruje vety obsahujúce subjektívny názor. Z kontextu vieme určiť, k akým aspektom alebo ich atribútom sa jednotlivé vyfiltrované vety vzťahujú, takže druhý krok nám poskytuje informáciu, v akom svetle sa na ne konkrétna veta pozerá.

Niektoré výskumy na túto tému (popísané v [4]) sa zaoberajú oboma problémami (klasifikácia subjektivity, klasifikácia sentimentu na úrovni vety), niektoré iba jedným z nich. Nič to ale nemení na fakte, že v oboch prípadoch ide o úlohu z oblasti klasifikácie, čo znamená, že aplikovateľné sú klasické metódy strojového učenia s učiteľom. Väčšina týchto výskumov tiež pracuje s predpokladom, že akákoľvek analyzovaná veta obsahuje práve jeden názor (vyjadrenie k práve jednej entite alebo k jej aspektu) od práve jednej osoby (napríklad *„Kvalita fotiek vytvorených týmto fotoaparátom je úžasná.“*).

Prejdime k samotným postupom analýzy použitých v týchto výskumoch. V jednom z nich je pre klasifikáciu subjektivity viet použitá metóda strojového učenia s učiteľom. Pre následnú klasifikáciu sentimentu subjektívnych viet je zasa použitá metóda podobná tej, ktorá je opísaná v podkapitole 2.2.1 v rámci klasifikácie pomocou strojového učenia bez učiteľa. Rozdielmi sú použitie viacerých tzv. *seed words* a iný tvar použitých matematických funkcií. Iný výskum pracuje i s akostnými prídavnými menami (také, ktoré sa dajú stupňovať), ďalší zasa využíva

kombináciu metód strojového učenia s učiteľom a bez učiteľa. V jednom tiež boli vytvorené modely pre identifikovanie konkrétnych druhov názorov.

Ako bolo spomenuté vyššie, analýza sentimentu na úrovni vety nie je úplne vhodná pre vety obsahujúce vyjadrenia k viacerým entitám alebo ich aspektom. Ideálnymi kandidátmi sú preto jednoduché vety. Aj tie ale môžu obsahovať viacero názorov. Rovnako tak zložené vety, resp. ich vedľajšie vety, ktoré je dôležité identifikovať. Tiež je podstatné určiť intenzitu názorov, ktoré obsahujú. To bolo vykonané v jednom z výskumov a konkrétnym názorom bola priradená jedna zo štyroch hodnôt intenzity (vysoká, stredná, nízka a neutrálna; neutrálna signalizuje absenciu názoru, prípadne nesubjektívne vyjadrenie). Práve tento fakt naznačuje, že opisované riešenie v sebe zahŕňa i klasifikáciu subjektivity. Rovnaký problém rieši i iný výskum (pomocou strojového učenia s učiteľom) a do úvahy berie i slová, ktorých výskyt môže zmeniť sentiment (napr. negácie – *nie*, *nikdy*).

Ďalšie výskumy boli podniknuté s testovacími dátami z konverzačných vlákien (napr. z fór alebo e-mailov). Pre takéto dáta je špecifické, že príspevky z nich obsahujú názory nie len k téme, ale tiež interakciu medzi ich autormi. V spomínaných výskumoch bola preto vytvorená metóda extrahujúca iba také vety, ktoré obsahujú názory ľudí na entity alebo ich aspekty.

2.2.3 Analýza sentimentu na úrovni aspektu

Analýza sentimentu na úrovni dokumentu i vety nie je schopná odhaliť, čo konkrétne ľudia vo svojich názoroch vyzdvihujú a čo zasa hodnotia záporne. Práve takýto pohľad na analyzované dáta ponúka analýza sentimentu na úrovni aspektu (v angličtine sa pre ňu okrem názvu *aspect-based sentiment analysis* používa i označenie *feature-based sentiment analysis*). Táto analýza sa nezaoberá stavebnými konštrukciami jazyka (odsekmi, vetami), ale zameriava sa priamo na názor ako taký. Je založená na princípe, že názor pozostáva z cieľu a z jeho sentimentu (kladného alebo záporného). Podstatnou vecou je tiež identifikácia samotného cieľu, pretože názor vyjadrujúci sa k neznámemu cieľu je v praxi väčšinou nepoužiteľný. Týmto cieľom môže byť entita alebo jeden z atribútov entity. Úlohou tejto analýzy je určiť sentiment pre každý takýto cieľ.

Poďme si uviesť príklad. Majme vetu *„Kvalita hovoru smartfónu je dobrá, ale výdrž jeho batérie je slabá“*. Táto veta sa obsahuje vyjadrenie k entite *smartfón*, konkrétne k jej aspektom *kvalita hovoru* (hodnotený kladne) a *výdrž batérie* (hodnotený záporne). Hodnotené aspekty entity sú zároveň i vyššie zmieňovanými cieľmi analýzy. Ako teda vidíme, jej výstupom môže byť štruktúrovaný sumár názorov o entitách a ich aspektoch (a jej činnosť môže byť popísaná ako prevod neštruktúrovaného textu na štruktúrované dáta). Tento typ analýzy je oveľa komplikovanejší ako analýza sentimentu na úrovni dokumentu a analýza sentimentu na úrovni vety.

Z formálneho hľadiska je úlohou analýzy sentimentu na úrovni aspektu odhaliť každú päticu $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ v dokumente d. To znamená, že v rámci tejto analýzy je potrebné vykonať päť

úloh. Extrakcia entity e_i, autora h_k a času vytvorenia t_l dokumentu d je popísaná v úvode celej tejto podkapitoly o analýze sentimentu (podkapitola 2.2).

Jednou z úloh je extrakcia aspektov (*aspect extraction*, *aspect expression extraction*). Ide o zistenie takých atribútov entity, ktoré sú v rámci analyzovaného dokumentu hodnotené. Dôležité pri tom je, aby sme vedeli, ku ktorej entite sa extrahované aspekty vzťahujú. Jedna z techník použitých na extrakciu aspektov (popísaná v [4]) využíva metódy strojového učenia bez učiteľa. Pozostáva z dvoch krokov. V tom prvom sa vyhľadajú najfrekventovanejšie podstatné mená a frázy obsahujúce podstatné mená, a to pomocou tzv. *POS taggeru* (zistením príslušných slovných druhov jednotlivých slov). V druhom kroku sa zasa vyhľadávajú najfrekventovanejšie aspekty (s využitím vzťahov medzi aspektmi a slovami a frázami indikujúcimi názor). Iné techniky extrakcie aspektov pracujú s metódami strojového učenia s učiteľom (prípadne s kombinovanými), so zhlukovaním a s odhaľovaním tém (tzv. *topic modelling*). Použité tiež môžu byť techniky extrakcie informácií - skrytý Markovov model, vyhľadávanie sekvenčných vzorov (*sequential rule mining*) a technika *conditional random fields*.

Ďalšou úlohou je klasifikácia sentimentu aspektu. Ide o určenie polarít názorov na rôzne aspekty. Pri tejto úlohe môžu byť užitočné metódy používané v rámci vyššie popisovanej analýzy sentimentu na úrovni vety a analýzy vedľajšej vety. Tieto metódy však nie sú schopné identifikovať zmiešané názory a nahradiť analýzu na úrovni slov a fráz. V jednom z výskumov bolo ukázané (ako uvádza [4]), že dobrú efektivitu vykazuje prístup používajúci lexikón (zoznam slov a fráz indikujúcich názor) a množinu pravidiel na určenie orientácie jednotlivých názorov. Do úvahy sú v tomto prípade tiež brané slová meniace orientáciu sentimentu („*nie*“, „*nikdy*“) a vedľajšie vety začínajúce slovom *„ale“*. Metóda pozostáva zo štyroch krokov. Prvým je označenie slov a fráz indikujúcich názor. Druhým a tretím je spracovanie slov meniacich orientáciu sentimentu, resp. spracovanie vedľajších viet. Posledným krokom je výpočet výslednej orientácie každého aspektu. Efektivita tohto algoritmu je vo veľkom počte prípadov celkom dobrá, no nájdu sa aj také, v ktorých je jej hodnota horšia. Ide najmä o dáta s menšou frekvenciou výskytov slov a fráz indikujúcich názor a s vyššou frekvenciou výrazov reprezentujúcich samotnú hodnotu sentimentu.

V [6] sa síce neuvažuje extrakcia entity, autora a času vytvorenia, no k úlohám pridáva ďalšiu, a tou je agregovanie aspektov, ktoré sú významovo podobné. Problém je teda rozdelený do celkovo troch úloh – extrakcia aspektov, ich agregovanie a napokon analýza sentimentu pre jednotlivé aspekty.

Čo sa týka konkrétnych metód strojového učenia použitých pri implementácií analýzy sentimentu na úrovni aspektu, v [7] je konštatované, že najviac víťazných riešení súťaže *SemEval* (viď [8]) pracuje s metódou *SVM* (rovnako ako napr. riešenia [9] a [10]). Samotná [7] ale pracuje s neurónovými sieťami a tvrdí, že výsledky tohto experimentu sú porovnateľné a v niektorých prípadoch dokonca lepšie ako najlepšie výsledky spomínanej súťaže. Zaujímavosťou v tomto prípade

tiež je, že model je vytvorený na základe tréningových dát týkajúcich sa recenzií notebookov, no vykazuje dobré výsledky i v oblasti recenzií hotelov.

2.2.4 Porovnávacie názory

Pri analýze sentimentu na akejkoľvek úrovni je dôležité vedieť, s akým druhom názorov pracujeme. V [4] sú definované dva druhy. Tým prvým sú obyčajné (*regular*) názory (predchádzajúce podkapitoly uvažovali analýzu tohto typu názorov). Nachádza sa v nich vyjadrenie na práve jednu entitu alebo jej aspekt. Príkladom môže byť veta *„Kvalita mäsa je vynikajúca“*. V nej je aspektom *kvalita mäsa* a ten je hodnotený pozitívne. Druhým definovaným druhom názorov sú porovnávacie (*comparative*) názory. Ako už samotný názov napovedá, ide o názory, ktoré porovnávajú viacero entít na základe ich spoločného aspektu. Majme, napríklad, vetu *„Kvalita mäsa je lepšia ako kvalita ovocia“*. Tá porovnáva entitu *mäso* a entitu *ovocie* na základe aspektu *kvalita*, pričom preferencie sú na strane mäsa.

V tejto podkapitole sa ďalej budeme venovať iba porovnávacím názorom. Od tých obyčajných sú odlišné nie len po sémantickej stránke, ale i po tej syntaktickej. Vo väčšine prípadov sa v nich vyskytujú prídavné mená a príslovky v komparatíve (v druhom stupni) alebo v superlatíve (v treťom stupni). Názory obsahujúce komparatív vyjadrujú, že jedna z entít má lepšie alebo horšie vlastnosti ako iná, berúc do úvahy konkrétny atribút. Názory so superlatívmi zasa tvrdia, že jedna z entít má najvyššie alebo najnižšie kvality daného atribútu spomedzi jej podobných entít. Do úvahy tiež treba brať mnoho iných slov, ktorými sa dá vyjadriť porovnanie (napr. *preferovať* či *uprednostňovať*).

Porovnávanie je možné popísať reláciami. V tomto prípade existujú štyri typy relácií vyjadrujúcich porovnávanie. Prvou z nich je komparatívna relácia, ktorá obsahuje akési usporiadanie. To môže byť rastúce (napr. *je rýchlejší ako*) alebo klesajúce. (napr. *je pomalší ako*). Väčšinou ide o preferencie. Druhou je relácia rovnosti (napr. *je rovnaký*), treťou superlatívna relácia, ktorá vyzdvihuje jednu z entít nad ostatné (napr. *je najrýchlejší*). Štvrtý typ relácie neobsahuje, na rozdiel od prvých troch typov, stupňované prídavné mená alebo príslovky. Môže poukazovať na zhody alebo odlišnosti entít (napr. *„Chuť koly je iná ako chuť džúsu“*), na fakty, že entity majú odlišné (ale súvisiace) aspekty (napr. *„Stolové počítače používajú externé reproduktory, zatiaľ čo notebooky používajú vlastné zabudované interné reproduktory“*) alebo na chýbajúce aspekty jednej z entít (napr. *„iPhone 6 obsahuje konektor pre slúchadlá, ale iPhone 7 už tento konektor neobsahuje“*).

Z formálneho hľadiska (uvedeného v [4]) môžeme porovnávacie názory popísať ako množinu šestíc (E_1, E_2, A, PE, h, t). E_1 a E_2 sú množiny entít, ktoré sú porovnávané na základe ich spoločného aspektu A. $PE \in \{E_1, E_2\}$ je množina entít, ktorú uprednostňuje autor názoru h a t je čas, v ktorom bol názor vyslovený. Dolovanie porovnávacích názorov z kolekcie dokumentov d potom môžeme definovať, ako získanie všetkých takýchto šestíc z kolekcie dokumentov d.

Uveďme si príklad. Uvažujme vetu *„Objektív Canonu je lepší ako objektív Nikonu alebo Sony“*, ktorú napísal Viktor v roku 2016. Ide o porovnávací názor, ktorý môžeme vyjadriť šesticou *({Canon}, {Nikon, Sony}, {objektív}, {Canon}, Viktor, 2016)*.

Extrakcie entít, ich aspektov, autorov názorov a časov ich vytvorenia sú v prípade porovnávacích názorov rovnaké úlohy ako pri obyčajných názoroch. Aplikovateľné sú teda postupy popísane v predchádzajúcich podkapitolách.

2.3 Indexovanie dát

Princíp indexovania spočíva v spracovaní textu za účelom umožnenia rýchleho vyhľadávania nad jeho obsahom. Počas samotného procesu dochádza k zhromaždeniu textov, k rozdeleniu jednotlivých textov (parsovaniu) a k ich uloženiu takým spôsobom, aby bolo neskoršie vyhľadávanie rýchle a presné. Podľa [11] môže byť táto operácia vykonaná nielen nad textami prirodzeného jazyka, ale i na akejkoľvek inej textovej informácií, či už je to zdrojový kód programovacieho jazyka, databáza DNA, prípadne textové dáta uchovávané v tradičných databázových systémoch. Pokusy o indexovanie elektronických dokumentov možno nájsť v literatúre od čias začiatku existencie výpočtových systémov. V 50. rokoch 20. storočia už dokonca existovali systémy schopné indexovať a vyhľadávať text.

V dnešnej dobe je indexovanie dát neoddeliteľnou súčasťou systémov pre získavanie informácií (tzv. *retrieval systems*). Od takýchto systémov je vyžadované vyhľadanie relevantného výsledku k zadanému dotazu v priebehu niekoľkých sekúnd, prípadne desatín sekúnd. Vzhľadom na fakt, že veľkosti kolekcií dokumentov systémov pre získavanie informácií neustále narastajú, dosiahnutie takéhoto času bez použitia indexu by bolo, ako tiež vraví [12], prakticky nemožné. Indexovanie tiež zohráva významnú rolu v algoritmoch používaných v oblasti správy dát (konkrétne pri integrácií a disambiguácií dát).

Vychádzajúc z informácií z [13], systémy vykonávajúce indexovanie využívajú redukciu dimenzií, kompresiu a iné techniky, aby tak redukovali veľkosť pamäte potrebnej pre uloženie indexu. Takisto sú vykonávané výskumy vzťahujúce sa k tvorbe a údržbe dynamických a distribuovaných indexov. Dôvodom je už spomínaný rast objemu dát a taktiež potreba decentralizovaného vyhľadávania.

Bez povšimnutia nemôžeme nechať informáciu, že [14], [15] a [16] sa zmieňujú o manuálnom indexovaní (*manual indexing, human indexing*). Zo samotného pojmu možno vyčítať, že ide o tvorbu indexu človekom. Výhodou tohto prístupu je presnosť výsledného indexu, no samotný proces jeho tvorby je časovo veľmi náročný. Opačným prístupom je automatické indexovanie textu. V porovnaní s manuálnym indexovaním je tento typ indexovania oveľa rýchlejší a menej náchylný na chyby, pričom jeho využitie je populárne pri práci s veľkými kolekciami dokumentov. Výskumy tiež ukázali, že efektivita výsledkov vyhľadávania je pre index vytvorený manuálne a index vytvorený

automaticky porovnateľná. A to napriek tomu, že automatické indexovanie je vykonávané bez akejkoľvek vedomosti o texte, ktorý tejto činnosti podlieha.

V ďalšom obsahu tejto podkapitoly o indexovaní sa budeme zaoberať iba automatickým indexovaním. Detailnejšie popísané budú niektoré techniky spadajúce do tejto oblasti.

2.3.1 Invertovaný index

Technika s názvom invertovaný index, pre ktorú existuje v angličtine viacero pomenovaní (*inverted index*, *inverted file*, *inverted file index*), je momentálne najpopulárnejšou indexovacou technikou, a to kvôli jej jednoduchosti a efektívnosti. Používaná je v mnohých aplikáciách. Ponúka výbornú podporu pre metódy získavania textu (*text retrieval methods*) založených na princípe množiny slov – *bag of words* (napr. pre metódu vektorového modelu *tf-idf*, modelu *BM25* alebo pravdepodobnostného jazykového modelu).

Táto technika má, podľa [17], dve základné stavebné jednotky. Tou prvou je slovník slov (*dictionary of terms*). Tiež sa môže vyskytovať pod názvami slovná zásoba (*vocabulary*), prípadne lexikón (*lexicon*). Ku každému slovu z tohto slovníku existuje zoznam (druhá stavebná jednotka) nazývaný zoznam výskytov (*postings list*) alebo invertovaný zoznam (*inverted list*). Ako už samotný názov napovedá, takýto zoznam reprezentuje výskyty daného slova v jednotlivých dokumentoch, pričom konkrétna položka zoznamu (výskyt) môže byť, napríklad, číslo konkrétneho dokumentu. Je nutné podotknúť, že kvôli efektivite techniky je užitočné udržovať zoznamy výskytov zoradené podľa vhodného parametra (napr. podľa čísla dokumentu).

Poďme sa pozrieť na proces budovania invertovaného indexu. Predpokladajme, že máme k dispozícii kolekciu textov, nad ktorými chceme samotný index vytvoriť. Prvým krokom je rozdelenie textov (tokenizácia) na slová, prípadne čísla. Ide o rozdelenie na tzv. tokeny. Z jednotlivých textov sa tak stanú zoznamy takýchto tokenov. Za tokeny sa nepovažujú interpunkčné znamienka („.“, „,“, „!“, ...), takže tieto časti pôvodného textu sa už v zoznamoch tokenov nenachádzajú. Po rozdelení textov na tokeny nasleduje lingvistické predspracovanie. V jeho priebehu sú zo zoznamov tokenov odstránené tzv. stop slová (*stop words*; často sa vyskytujúce slová, napr. spojky, predložky či niektoré slovesá a zámená), ktoré nemajú až taký veľký podiel na informácii, ktorú dokument nesie. Zostávajúce tokeny sú normalizované na kmeňový základ slova (kmeňový základ slov „učiť“, „učiteľ“ a „učenie“ je „uč“). Takáto normalizácia sa nazýva stemovanie (ako uvádza [11], najpopulárnejší algoritmus vykonávajúci stemovanie je pre anglický jazyk *algoritmus Martina Portera*). Po stemovaní nasleduje vytvorenie slovníku slov a zoznamov výskytov. Položkami slovníka sú tokeny, ktoré sú výsledkom lingvistického predspracovania. Slovník tiež môže obsahovať štatistiku o počte výskytov daného základu slova v konkrétnom dokumente.

Ukážme si vyššie uvedený postup na konkrétnom príklade. Uvažujme malú kolekciu troch dokumentov. Prvý dokument obsahuje text *„Toto je dokument číslo jeden.“*, text druhého dokumentu

je *„Toto je dokument číslo dva, áno dva.“* a obsahom toho tretieho je text *„Toto je dokument číslo tri, áno tri.“*. Tokenizácia rozdelí prvý dokument na tokeny toto, je, dokument, číslo, jeden, druhý dokument na tokeny toto, je, dokument, číslo, dva, áno, dva, no a ten tretí na tokeny toto, je, dokument, číslo, tri, áno, tri. Tokeny toto a je možno považovať za stop slová. Lingvistické predspracovanie by ich teda zo zoznamu tokenov odstránilo a zostali by tokeny dokument, číslo, jeden, dva, tri a áno. Stemovanie môžeme pre jednoduchosť vynechať. Nezmenil by sa ním počet tokenov, iba ich tvar. Výsledný slovník slov a zoznamy výskytov, ktoré napokon vzniknú, uvádza Obrázok 2.2.

Obrázok 2.2: Slovník slov a zoznamy výskytov (obsahujúce čísla dokumentov) pre uvažovanú kolekciu dokumentov.

Invertovaný index poskytuje, ako tiež uvádza [18], jednoduchý spôsob, ako nájsť odpoveď na dotazy. Povedzme, že nás zaujímajú dokumenty súvisiace s konkrétnou množinou slov. V tom prípade si pre každé slovo z tejto množiny musíme zistiť zoznam výskytov pre toto slovo. Po vykonaní prieniku nad získanými zoznamami výskytov dostaneme množinu čísiel dokumentov relevantných k hľadanej množine slov. To bol príklad konjunktívneho dotazu. Pri dotaze disjunktívnom stačí, podľa [19], vykonať namiesto prieniku zoznamov výskytov ich zjednotenie.

Na záver tejto podkapitoly si uveďme klady a zápory opisovanej indexovacej techniky. Medzi pozitíva patria časové i priestorové vlastnosti. Kvôli zlepšeniu efektivity sa odporúča ukladať slovník slov do dátových štruktúr akými sú B+ strom alebo hashovacia tabuľka. Problém

pri invertovanom indexe môže nastať pri výskyte synonýmie (jav, pri ktorom má viacero slov rovnaký význam) alebo polysémie (jav, pri ktorom má konkrétne slovo viac významov).

2.3.2 Index pracujúci s dvojicami slov

Mnoho technických termínov, názvov organizácií alebo produktov (napríklad *The New York Times*) je zložených z viacerých slov. Pri použití tzv. frázových dotazov (*phrase queries*) na vyhľadanie dokumentov obsahujúcich takéto termíny či názvy už ale koncept zoznamu výskytov v zmysle zoznamov dokumentov, v ktorých sa nachádzajú jednotlivé slová, nepostačuje.

Jedným z možných prístupov, ako riešiť takéto situácie, je považovať za frázu každý pár po sebe nasledujúcich slov v rámci dokumentu. Takéto páry sa podľa [17] označujú ako tzv. *biwords*. Index nimi pracujúcimi sa potom v angličtine nazýva *biword index*.

Majme, napríklad, text *„Billa Bonus club“*. Dvojice slov k nemu prislúchajúce sú potom billa bonus a bonus club.

V rámci popisovaného prístupu je každý takýto pár považovaný za samostatnú jednotku (*vocabulary term*). Spracovanie dvojslovných frázových dotazov je v tomto prípade jednoznačnou záležitosťou – ide o vyhľadávanie príslušnej dvojice slov obsahujúcej hľadané slová v správnom poradí. Frázové dotazy obsahujúce viac ako dve slová ale treba rozčleniť. Pre príklad uvažujme dotaz *„Vysoké učení technické“*. Ten by bol rozčlenený na dva dotazy na dvojice slov spojené booleovským operátorom *AND* – *„vysoké učení“ AND „učení technické“*.

Na prvý pohľad sa zdá, že v praxi je táto technika úspešná. Môžu ale nastať i prípady, v ktorých dotaz úspešný nie je a ako výsledok vráti dokument, ktorý neobsahuje pôvodne hľadanú frázu.

Medzi často vyhľadávanými frázami sa môžu nachádzať i také, ktoré obsahujú podstatné mená. Tie sú ale vo frázach väčšinou fyzicky oddelené spojkami, predložkami atď. Príkladom nech je fráza *„kniha o varení pre začiatočníkov“*. Do modelu pracujúcom na báze indexovania dvojíc slov sa tento fakt premietne nasledovne. Fráza je rozdelená na tokeny a každému tokenu je priradený príslušný slovný druh. Tokeny sú následne rozdelené do dvoch tried – jedna obsahuje podstatné mená (označovať ju budeme *N*), zatiaľ čo spomínané predložky, spojky a ostatné slovné druhy zaradíme do tej druhej (označíme ju *X*). Každý reťazec slov majúci vzor *NX*N* potom môžeme považovať za rozšírenú dvojicu slov, ktorý bude uložený ako samostatná jednotka (fráza *„kniha o varení“* by mala vzor *NXN*). Spracovanie dotazu pracujúceho s rozšírenými dvojicami slov má podobný priebeh – text treba rozdeliť na tokeny, pre každý z nich treba určiť slovný druh, na základe slovného druhu ho zaradiť do triedy *N* alebo *X* a na základe toho treba napokon dotaz rozdeliť na samostatné rozšírené dvojice slov, ktoré budú vyhľadávané.

2.3.3 Frázový index

Koncept indexovania dvojíc slov môže byť, ako uvádza [17], rozšírený do dlhších indexovaných sekvencií slov. Ak index obsahuje takéto sekvencie s rôznymi dĺžkami, označovaný je ako frázový index (*phrase index*). Nevýhodou ale je, že vyhľadávanie jednoslovných dotazov nie je úplne efektívne (toto slovo treba vyhľadávať vo všetkých uložených položkách slovníku). Je teda vhodné mať k dispozícii aj index jednoslovných položiek slovníku. Výhodou naopak je zníženie pravdepodobnosti nájdenia irelevantných dokumentov pre frázy obsahujúce 3 a viac slov. Ukladanie dlhších fráz zasa môže mať za následok značné zväčšenie veľkosti slovnej zásoby.

2.3.4 Pozičný index

Index založený na dvojiciach slov a jeho všeobecný variant – frázový index – nie sú v praxi až tak často používané (viď [17]). Prednosť pred nimi dostáva pozičný index (*positional index*). Ten pre každé slovo zo slovnej zásoby uchováva zoznam výskytov vo všeobecnom tvare: *číslo_dokumentu: [pozícia_1, pozícia_2, ...]*, kde *pozícia_n* nesie hodnotu indexu daného tokenu v danom dokumente. Súčasťou každého zoznamu výskytov je väčšinou i údaj o frekvencii daného slova.

Spracovanie frázového dotazu znamená prístup k záznamom jednotlivých tokenov v invertovanom indexe. Okrem kontroly prítomnosti slov v dokumente je tiež potrebné kontrolovať, či ich pozície v tomto dokumente reflektujú poradie, v ktorom sú uvedené vo frázovom dotaze. Tento prístup preto vyžaduje znalosť ofsetov medzi slovami.

Uveďme si príklad pozičného indexu (prevzatý zo [17]). Majme dva zoznamy výskytov (pre jednoduchosť bez údajov o frekvenciách výskytov jednotlivých slov) – jeden pre slovo *to*, druhý pre slovo *be*:

to:[
  1: [20, 30];
  2: [8, 29, 33];
  3: [16, 44];
…
]

be:[
  1: [5, 18];
  2: [30, 34, 49];
  3: [9, 25];
…
]

Z nich je možné vyčítať, že slovo *to* sa nachádza v dokumente *1* na pozíciách *20* a *30*, v dokumente *2* na pozíciách *8*, *29* a *33* atď. Ďalej predpokladajme frázový dotaz *„to be or not to be“*. V rámci spracovania tohto dotazu je treba skontrolovať zoznamy výskytov slov *to*, *be*, *or* a *not*. My budeme kontrolovať iba zoznamy výskytov slov *to* a *be* – kontrola tých ostatných bude prebiehať obdobne. Najprv teda nájdime dokumenty, ktoré obsahujú obe slová. Sú to dokumenty označené číslami *1*, *2* a *3*. Ďalším krokom je kontrola pozícií. Pozícia slova *be* by mala byť v porovnaní s pozíciou slova *to* vyššia o hodnotu jedna. A to v dvoch prípadoch. V tom druhom by malo ísť

o pozície zvýšené o hodnotu *4* (v porovnaní s prípadným prvým výskytom). Vidíme, že týmto obmedzeniam vyhovujú pozície *29* (*to*), *30* (*be*), *33* (*to*) a *34* (*be*) v dokumente číslo *2*.

Pozičný index tiež môže slúžiť napríklad na vyhľadávanie slova v rozpätí *k* slov od iného slova (pomocou indexu používajúceho dvojice slov by toto možné nebolo). Jeho použitie však značne zvyšuje veľkosť zoznamov výskytov, a to i v prípade optimalizácie týchto štruktúr. Taktiež je zväčšená zložitosť operácie prieniku zoznamov výskytov (počet kontrolovaných položiek už nie je ohraničený počtom dokumentov, ale počtom tokenov v kolekcii dokumentov).

Za zmienku tiež stojí fakt, že kombinácia pozičného indexu s indexom využívajúcim dvojice slov, prípadne frázovým indexom, môže byť prínosom. Pri frekventovaných dotazoch je totiž neefektívne stále dookola kontrolovať zoznamy slov pozičného indexu. Kombinácia dvoch popísaných prístupov používa pre určité frázové dotazy frázový index a pre ostatné zasa pozičný index. Ideálnou voľbou je zahrnúť časté dotazy práve do frázového indexu. Ak však frázový index obsahuje frázy, ktoré sú tvorené často používanými slovami, no ktorých samotný výskyt ako taký je už menej častý (napríklad slovné spojenie *The Who*), zrýchlenie vyhľadania relevantných výsledkov je značné.

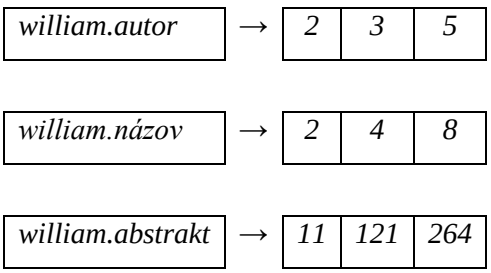
2.3.5 Parametrický index

Doposiaľ sme sa na dokumenty pozerali ako na sekvenciu slov. Väčšina dokumentov ale v skutočnosti obsahuje i ďalšiu štruktúru, tzv. metadáta. Prostredníctvom takejto štruktúry sú uchované informácie o danom dokumente, ako napríklad meno jeho autora, jeho názov alebo dátum vydania. O týchto informáciách hovoríme vo všeobecnosti ako o poliach (*fields*) metadát, ktoré môžu nadobúdať konečný počet hodnôt. Pre každé takéto pole (napríklad dátum vydania dokumentu) je potom vytvorený parametrický index (*parametric index*). Prostredníctvom neho sme schopní vyhľadať dokumenty, v ktorých je hodnota konkrétneho poľa zhodná s požadovanou hodnotou v dotaze (napríklad s hľadaným dátumom vydania dokumentu). Hodnoty niektorých polí môžu byť usporiadané (ako napríklad už spomínaný dátum vydania dokumentu). Na základe tohto faktu potom prichádzajú do úvahy i dotazy podporujúce vyhľadávanie na základe výskytu hodnoty poľa dokumentu v zadanom rozsahu hodnôt.

2.3.6 Zónový index

Podobným konceptom, akým sú polia spomínané v predchádzajúcej podkapitole, sú i tzv. zóny. Rozdiel medzi poľami a zónami je však ten, že zóny môžu obsahovať neobmedzene dlhý ľubovoľný text (zatiaľ čo polia môžu nadobudnúť iba konečný počet hodnôt). Za zónu býva často považovaný napríklad abstrakt dokumentu. Je vhodné, aby bol pre každú zónu dokumentu vytvorený samostatný invertovaný index. Podporované potom môžu byť dotazy typu *„vyhľadaj dokument obsahujúci slovo*

`william` v mene autora, slovo `romeo` v názve a slovo `julia` v samotnom tele dokumentu“. Príklad takéhoto zónového indexu (tiež označovaného ako *zone index*) ilustruje Obrázok 2.3.



Obrázok 2.3: Príklad zónového indexu.

Ak je dôraz kladený na veľkosť zónového indexu, existuje metóda, ktorou je možné jeho veľkosť redukovať. Dosiahneme to tak, že názov zóny, ktorá obsahuje dané slovo, uvedieme do zoznamu výskytov.

Takáto redukcia má tiež ďalší podstatný význam – zefektívni sa ňou výpočet hodnoty váženého skóre jednotlivých zón (*weighted zone scoring*). Uvažujme dotaz *q* a dokument *d*. Táto metóda priraďuje páru *(q, d)* skóre z intervalu *<0, 1>*, a to pomocou výpočtu skóre jednotlivých zón. Synonymom pre tento proces je názov *ranked Boolean retrieval*.

3 Návrh systému

Táto kapitola popisuje všetky podstatné fakty a okolnosti, ktoré bolo treba brať do úvahy pri návrhu systému. V jej závere je uvedená schéma reprezentujúca výstup samotného návrhu.

3.1 Sťahovanie relevantných dát z webu

Dáta sú v prípade tejto práce jedným zo základným stavebných kameňov. Ich získavanie je teda činnosť, ktorá je veľmi dôležitá. Ako zdroje dát sú v rámci tejto práce najvhodnejšie sociálne siete a diskusie k článkom. V nich je možné nájsť veľké množstvo príspevkov na tému českých potravinových obchodných reťazcov.

Zhromažďovanie relevantných dát z webu ako také môže prebiehať niekoľkými spôsobmi. Závisí to najmä od konkrétnych zdrojov dát. Nech už je to ale akýkoľvek zdroj (sociálna sieť, diskusia k článku, fórum atď.), je tu zopár informácií, ktoré by, bez ohľadu naň, mali byť evidované.

3.1.1 Všeobecné informácie o dátach

V prípade príspevku (a tiež komentára alebo odpovede na komentár) by bolo vhodné okrem jeho textu ukladať informáciu o jeho presnej *URL* adrese (ak to je, samozrejme, možné). Pomocou nej sme totiž vo väčšine prípadov schopní vyhľadať tento príspevok znovu priamo na webe. To by sa mohlo zísť v prípadoch, ak by text samotného príspevku príliš nedával zmysel a potrebovali by sme zistiť jeho kontext. Znalosť *URL* adresy profilu autora príspevku (takáto informácia nemusí byť opäť vždy prístupná) je taktiež vítaná. Pomocou nej môžeme z tohto profilu získať informácie o konkrétnej osobe (ak to je potrebné).

Nemenej podstatnou informáciou je čas vytvorenia príspevku. Je totiž dôležité vedieť, k akému dátumu alebo obdobiu sa príspevok vzťahuje. Práve tento údaj nám môže, rovnako ako vyššie spomenutá *URL* adresa príspevku, poodhaliť kontext príspevku (slovo *prázdniny* môže byť vnímané inak v letných mesiacoch a inak zasa v mesiacoch zimných). Stiahnuté dáta je potom možné triediť na základe času ich vytvorenia (napríklad podľa dní, týždňov, mesiacov atď.). V prípade, že informácia o čase vytvorenia príspevku nie je dostupná, k dispozícii máme minimálne informáciu, kedy sme príspevok získali. Otázne ale je, či nám tento údaj v konečnom dôsledku vôbec pomôže.

Čas vytvorenia príspevku je tiež osožný pri získavaní nových dát. Či už ide o nástenku na sociálnej sieti alebo diskusiu, časový údaj vytvorenia príspevku je jednoduchý spôsob, ako zistiť, či už sme niekedy konkrétny príspevok sťahovali, prípadne ukladali, alebo nie.

3.1.2 Formát dát

Dôležitosť jednotlivých informácií týkajúcich sa príspevkov sa odlišuje s účelmi použitia dát. Ak sú takéto dáta získavané z rôznych zdrojov, v každom prípade je vhodné a efektívne zaobchádzať so všetkými dátami (bez ohľadu na ich pôvod) po ich získaní jednotne. Predpokladom tohto prístupu je existencia jednotného formátu pre dáta. Pri prevode doň ale musíme byť dôverne oboznámení s formátmi dát z pôvodných zdrojov. Hodnotou, ktorú môžu rôzne zdroje uvádzať rozdielne, je napríklad už spomínaný časový údaj vytvorenia príspevku.

3.1.3 Dáta zo sociálnych sietí

V prípade sťahovania dát zo sociálnych sietí je tu istá výhoda v podobe kontextu, v ktorom dáta sťahujeme. Zo sociálnej siete *Facebook* je napríklad možné prostredníctvom *Graph API* (viď [20]) stiahnuť príspevky nachádzajúce sa na nástenke (anglickým ekvivalentom v terminológii sociálnej siete *Facebook* je slovo *feed*) danej stránky (*page*). Znamená to veľkú pravdepodobnosť, že takéto príspevky sa budú týkať práve stránky, ku ktorej sa týmto spôsobom vzťahujú. Ďalšou podstatnou informáciou je fakt, že takéto stránky majú kvôli marketingovým účelom či interakciou so zákazníkmi i české potravinové obchodné reťazce.

Problémom by ale mohli byť také príspevky, ktoré sú vytvorené priamo stránkou, na nástenke ktorej sa nachádzajú. Takýto typ dát pre nás nie je z pohľadu analýz hodnotný, pretože pravdepodobne neobsahuje žiaden subjektívny názor ani spätnú väzbu týkajúcu sa ponuky obchodného reťazca, jeho služieb, zamestnancov atď. Reč je ale stále o nástenke konkrétnej stránky, čo znamená, že poznáme jej názov. Prostredníctvom porovnania názvu tejto stránky s menom autora príspevku vieme nechcené dáta jednoducho odfiltrovať.

Z pohľadu analýz sú pre nás síce takéto príspevky nevhodné, no v istých prípadoch môžu byť pre nás i osožné. Najmä ak ide o také príspevky, ktoré sú reakciou (komentárom, prípadne komentárom ku komentáru) na príspevok spotrebiteľa. Z takýchto dát by potom mohlo byť možné vyčítať postoj obchodných reťazcov k spätnej väzbe od zákazníkov alebo ich záujem riešiť prípadné vzniknuté problémy.

3.1.4 Dáta z diskusií k článkom

V porovnaní so zhromažďovaním relevantných dát zo sociálnych sietí je situácia so sťahovaním dát z diskusií k článkom komplikovanejšia. V prvom rade je potrebné nájsť spôsob, akým sa k samotný článkom vôbec dostať. Jedným z možných riešení je napríklad odber noviniek pomocou *RSS* kanálov rôznych spravodajských portálov. Ešte predtým sa ale treba presvedčiť, že články daného spravodajského portálu vôbec diskusiu obsahujú.

Ďalším dôležitým krokom (za predpokladu, že k dispozícii už máme nejaký zoznam článkov - pravdepodobne vo forme ich *URL* adries) je výber takých článkov, ktorých obsah je pre naše účely

relevantný (ak je relevantný ich obsah, pravdepodobne budú relevantné i diskusie k takýmto článkom). Toto je možné zabezpečiť napríklad pomocou vyhľadávania vybranej množiny kľúčových slov v titulku článku alebo v jeho samotnom obsahu.

Zoznam relevantných článkov ešte nemusí byť pre stiahnutie diskusií k nim postačujúci. Stojí za tým fakt, že v niektorých prípadoch sa diskusia k článku nenachádza na rovnakej stránke ako článok samotný. V takýchto prípadoch je nutné zistiť URL adresu diskusie k článku. Niekedy ide iba o úpravu *URL* adresy článku, ku ktorému sa diskusia vzťahuje, niekedy to zasa môže byť *URL* adresa, ktorá sa takýmto spôsobom odhadnúť nedá (môže napríklad obsahovať jednoznačný identifikátor) a treba ju vyhľadať v *HTML* kóde stránky s článkom. Všetko to, samozrejme, záleží od konkrétneho spravodajského portálu.

V tomto momente nám už nič nebráni získať príspevky z diskusie. Ani táto fáza ale nie je bezproblémová. Nejde síce o závažný problém, no otázkou je, ako určiť, ktorého obchodného reťazca sa týkajú komentáre, ktoré vo svojom tele neobsahujú názov žiadneho takéhoto reťazca. Jedným z možných riešení je prehlásiť, že takéto komentáre sa týkajú obchodného reťazca spomenutého v titulku alebo obsahu článku, ku ktorému diskusia patrí. Iným riešením je zasa (predpokladom je ale zoradenie komentárov podľa času ich vytvorenia) prípadne rekurzívne určiť, že takýto komentár sa týka toho reťazca, ktorého sa týka i predchádzajúci komentár. A to až dovtedy, pokým nie je nájdený komentár s explicitnou zmienkou názvu jedného z hľadaných obchodných reťazcov. Do úvahy prichádza i kombinácia týchto dvoch prístupov.

V popísanom procese sťahovania dát z diskusií k článkom stojí za zváženie ukladanie zoznamu *URL* adries všetkých nájdených relevantných článkov. Takéto niečo môže byť prínosné z toho dôvodu, že nové komentáre môžu pribúdať i v diskusiách starších článkov.

3.2 Analýza dát

Fázou, ktorá logicky nadväzuje na sťahovanie dát, je ich analýza. V tejto podkapitole sú popísané všetky druhy zamýšľaných analýz, ktorým sa zhromaždené dáta podrobia.

3.2.1 Predspracovanie textu

Predtým, ako budú na dátach vykonané jednotlivé analýzy, je potrebné tieto dáta predspracovať. To v praxi znamená, že by sme mali byť schopní rozdeliť text konkrétneho príspevku na samostatné vety a tie zasa na samostatné slová (tokeny). Takto získané slová by následne bolo možné transformovať do základného tvaru (na tzv. lemy), prípadne do kmeňového tvaru (v angličtine označovaného ako *stem*). K dispozícii by tiež bol zoznam často sa vyskytujúcich slov (stop slov), pomocou ktorého by sme boli schopní takéto slová z pôvodného textu odstrániť. Za súčasť predspracovania textu môžeme ďalej považovať i také transformácie dát, ktorých výsledkom je text obsahujúci iba malé písmená alebo text neobsahujúci diakritiku.

3.2.2 Analýza sentimentu

Prvou z analýz, ktoré by mali byť implementované, je analýza sentimentu. Konkrétne jej dve verzie – analýza sentimentu na úrovni dokumentu a analýza sentimentu na úrovni aspektu.

Analýzou sentimentu na úrovni dokumentu je myslená analýza príspevku ako celku. Výsledok tejto analýzy by potom reprezentoval celkové vyznenie tohto príspevku, resp. postoja jeho autora. Bola by ním jedna z hodnôt *kladný*, *neutrálny* alebo *záporný*.

Analýza sentimentu na úrovni aspektu by zasa analyzovala text príspevku detailnejšie. Snažila by sa odhaliť, o čom konkrétne sa príspevku hovorí (ktoré vlastnosti obchodného reťazca autor zmieňuje) a v akom zmysle (sú tieto vlastnosti spomenuté v kladnom, neutrálnom alebo zápornom svetle?). Výsledkom analýzy sentimentu na úrovni aspektu by teda bola množina dvojíc *(vlastnosť, polarita)*, kde hodnota *polarity* by bola, rovnako ako v prípade vyššie zmienenej analýzy sentimentu na úrovni dokumentu, jednou z hodnôt *kladný*, *neutrálny* alebo *záporný*.

3.2.3 Extrakcia lokalít

Táto analýza by zisťovala, či vôbec, prípadne aké lokality autor v texte zmieňuje. Lokalitou je v tomto prípade myslený názov mesta a/alebo konkrétna ulica (ulica, na ktorej sa nachádza jedna z pobočiek daného obchodného reťazca). Týmto spôsobom by teoreticky mohlo byť možné zistiť, k akému mestu sa vzťahuje text príspevku aj napriek tomu, že v ňom nie je explicitne spomenutý jeho názov.

Príkladom môže byť príspevok, v ktorom je uvedená ulica *Srbská* v spojitosti s obchodným reťazcom *Billa*. Z týchto informácií vieme určiť, že autor hovorí o pobočke obchodného reťazca *Billa* nachádzajúceho sa v Brne. Táto metóda ale zlyháva v prípadoch, ak sa pobočky jedného obchodného reťazca nachádzajú v rôznych mestách na uliciach s rovnakým názvom a autor príspevku neuvedie v texte presnú adresu pobočky (t.j. okrem názvu ulice i číslo ulice).

Podmienkou takejto extrakcie lokalít z textu je evidencia názvov miest v rámci Českej republiky rovnako ako i adries všetkých pobočiek jednotlivých reťazcov. Vhodné tiež môže byť evidovať spolu názvom mesta i jeho kraj. Informácia o kraji by totiž bola podstatná pre rozlíšenie lokalít v prípade, že viacero miest alebo obcí nesie rovnaký názov (obec *Bystřice* sa napríklad nachádza v Královohradeckom, v Moravskosliezskom i v Stredočeskom kraji). Okrem tohto účelu by ešte mohla informácia o kraji umožniť vizualizáciu výsledkov analýz na mape Českej republiky.

3.2.4 Extrakcia názvov obchodných reťazcov

Cieľom tohto druhu analýzy je zistiť, názvy ktorých obchodných reťazcov sú prípadne v texte spomenuté. Takáto informácia by mohla byť prínosnou v dvoch smeroch.

Tým prvým je jej využitie počas sťahovania dát z diskusií k článkom (viď podkapitolu 3.1.4). Je totiž celkom pravdepodobné, že ak je v texte spomenutý názov obchodného reťazca, komentár sa týka práve jeho.

Druhý spôsob (a v istom slova zmysle možno opačný) využitia informácie o spomenutých názvoch obchodných reťazcov v analyzovanom texte sa týka najmä dát, ktoré majú pôvod na sociálnych sieťach. Pri takýchto dátach totiž väčšinou vieme, ku ktorému obchodnému reťazcu sa vzťahujú (ako tiež približuje podkapitola 3.1.3). Ak sú teda v príspevku zmienené názvy reťazcov, z ktorých ani jeden nie je zhodný s názvom reťazca, ktorého sa pôvodne príspevok týka, tento jav je možné považovať za zmienku o konkurencii. Príkladom môže byť príspevok pochádzajúci z nástenky na oficiálnej stránke obchodného reťazca *Albert* na sociálnej sieti *Facebook*, v ktorom sa nachádza zmienka napríklad o obchodnom reťazci *Billa*.

Ak chceme byť schopní extrahovať názvy obchodných reťazcov z textu, mali by sme mať k dispozícii zoznam týchto názvov. Tvorba takéhoto zoznamu by vzhľadom na početnosť potravinových obchodných reťazcov v Českej republike nemala byť výrazným problémom.

3.2.5 Extrakcia kľúčových slov

Poslednou z vykonávaných analýz by bola extrakcia kľúčových slov z príspevku. Jej cieľom je zistiť zoznam slov, ktoré charakterizujú tento príspevok najviac. Prostredníctvom kľúčových slov by následné bolo možné zistiť, o čom užívatelia (môžeme povedať, že v tomto prípade zároveň i spotrebitelia) hovoria, resp. píšu v súvislosti s obchodnými reťazcami najviac.

3.2.6 Formát výsledkov analýz

Predpokladajme, že každú z vyššie spomenutých analýz bude v praxi reprezentovať samostatná autonómne pracujúca trieda. Potom by bolo vhodné, aby výsledky týchto analýz boli v podobnom formáte (napríklad vo formáte *XML*), a to z dôvodu toho, aby ich bolo čo najjednoduchšie zlúčiť do jedného celku. Tento celok by potom reprezentoval komplexnú analýzu daného príspevku (obsahoval by údaj o jeho celkovej polarite, údaje o polaritách nájdených aspektov, zoznam spomenutých názvov miest a/alebo ulíc či zoznam spomenutých názvov obchodných reťazcov).

Na tento výsledný formát existujú tiež určité požiadavky. Jedným z nich je jednoduchá možnosť úpravy výsledkov jednej, prípadne i viacerých analýz užívateľom. Dôležitá je teda v tomto prípade i zrozumiteľnosť tohto formátu.

Ideálne by tiež bolo, keby bolo možné výsledky jednotlivých analýz a extrakcií zvýrazniť priamo v texte príspevku. Reč je o slovách reprezentujúcich aspekty spolu s ich polaritou, o názvoch miest a ulíc a o názvoch obchodných reťazcov. O takéto zvýraznenie by sa staral vhodný formát *HTML* kódu. Na jeho získanie ale potrebujeme poznať presné pozície jednotlivých slov, ktoré chceme zvýrazniť. Práve informácia o nich by sa mala nachádzať v spomínanom výslednom formáte (túto

požiadavku je podľa potreby možné delegovať na jednotlivé analýzy, ktorých sa to týka – na analýzu sentimentu na úrovni aspektu, na extrakciu lokalít a na extrakciu názvov obchodných reťazcov).

3.3 Indexovanie dát

Indexovanie dát je ekvivalentom ukladania dát do databáze. Dôležitou požiadavkou pri práci s indexom je mať navrhnutý pre dané účely vhodný formát, v ktorom budú dáta indexované. Ten potom môže (i nemusí) uľahčiť prácu s ním.

V rámci implementácie tohto systému by práca s indexom prichádzala do úvahy v nasledujúcich prípadoch. Tým prvým by bolo indexovanie dát po ich stiahnutí a analýze. Druhým prípadom by bola interakcia indexu s grafickým užívateľským rozhraním zobrazujúcim výsledky analýz indexovaných dát. Touto interakciou sú myslené rôzne kvalitatívne i kvantitatívne dotazy na indexované dáta alebo prípadné aktualizovanie indexovaných dát.

3.4 Metadáta

Súčasťou systému by mohli byť i metadáta. Tie by združovali informácie z rôznych oblastí jeho činnosti na jednom mieste. Takéto informácie by teda zároveň boli i jednoducho zistiteľné. Reč je napríklad o zozname aspektov, ktoré by mali byť analyzované, o zozname obchodných reťazcov, o ktorých by mali byť sťahované dáta, o zozname zdrojov, z ktorých by tieto dáta mali byť sťahované alebo o údajoch týkajúcich sa indexu, ktorý systém používa.

3.5 Koordinácia a automatizácia činnosti systému

Nachádzame sa v bode, v ktorom boli popísané takmer všetky časti tvoriace vyvíjaný systém. Pre úspešný beh systému je ešte potrebné zabezpečiť koordináciu činností týchto častí (napríklad v poradí stiahnutie dát – analýza dát – indexovanie dát).

Pri tvorbe takéhoto koordinačného prvku by bolo vhodné navrhnúť ho takým spôsobom, aby bolo bez zásahu do jeho implementácie možné spracovať (t.j. stiahnuť, analyzovať a indexovať) najnovšie i ľubovoľné historické dáta. Možným riešením by mohlo byť spracovanie dát z konkrétneho dátumu v rámci jedného behu.

Ďalej je potrebné zabezpečiť, aby bolo takéto spracovanie dát vykonávané automaticky a pravidelne. Okrem automatického vykonávania spracovávania dát stojí za zváženie i automatické spúšťanie nástroja vyhľadávajúceho a ukladajúceho relevantné články, resp. ich *URL* adresy (viď podkapitolu 3.1.4).

3.6 Vizualizácia výsledkov analýz

Výsledky jednotlivých analýz iba v textovej, prípadne číselnej podobe, by nám toho nemuseli prezradiť veľa. Preto by mohlo byť prínosné vhodným spôsobom ich vizualizovať.

3.6.1 Zoznam príspevkov

Jednou z variant, ako by mohli byť analyzované dáta podané užívateľovi, je zobraziť mu ich zoznam. Konkrétna položka takéhoto zoznamu by potom napríklad obsahovala text príspevku (prípadné nájdené slová reprezentujúce aspekty či názvy miest, ulíc a reťazcov by boli od zvyšného textu vizuálne odlíšené), dátum vytvorenia príspevku či indikátor spätnej väzby komunikačného oddelenia obchodného reťazca, ktorého sa príspevok týka (v prípade, že ide o príspevok získaný z oficiálnej stránky reťazca na sociálnej sieti *Facebook*). Jednotlivé položky by tiež mohli byť podfarbené i zoskupené na základe ich celkovej polarity.

3.6.2 Vnesenie výsledkov analýz na mapu Českej republiky

Táto vizualizácia by využívala výsledky vyššie zmieneného nástroja pre extrakciu lokalít spoločne s výsledkami analýzy celkovej polarity príspevku. Ako už názov podkapitoly napovedá, základom vizualizácie by bola mapa Českej republiky. Na nej by boli zvýraznené hranice jednotlivých krajov. Pre každý kraj by potom bol v jeho oblasti znázornený percentuálny podiel kladných, neutrálnych z záporných príspevkov, v ktorých bolo niektoré z miest (prípadne ulica, na ktorej sa nachádza pobočka daného reťazca v tomto meste) tohto kraja explicitne spomenuté.

Pre doplnenie kompletnosti informácií by k dispozícii mohla byť i tabuľka zobrazujúca početnosť príspevkov vzťahujúcich sa k jednotlivým krajom. Keďže nie vždy musí byť v príspevku spomenutá lokalita, bolo by tiež vhodné zobraziť počet príspevkov neobsahujúcich explicitnú zmienku o lokalite.

3.6.3 Trend celkovej polarity príspevkov

Na rozdiel od dvoch predchádzajúcich vizualizácií, trend celkovej polarity príspevkov by bol zobrazený pomocou trendového grafu. Vodorovná os by v tomto grafe reprezentovala časovú zložku (napríklad jednotlivé dni) a zvislá os rozdiel počtu príspevkov v danom dni majúcich celkovú pozitívnu, resp. negatívnu polaritu.

3.6.4 Vzájomné porovnanie reťazcov

Obchodné reťazce by medzi sebou mohli byť priamo porovnávané prostredníctvom stĺpcového grafu. Na vodorovnej osi grafu by sa nachádzali diskrétne hodnoty, konkrétne názvy jednotlivých obchodných reťazcov. Ku každému z obchodných reťazcov by v rámci grafu patrili tri stĺpce. Tie

by znázorňovali počet príspevkov týkajúcich sa daného obchodného reťazca majúcich kladnú, neutrálnu, resp. zápornú celkovú polaritu. Na jednom grafe by sme teda videli, ako často sa o ktorom obchodnom reťazci diskutuje (početnosť príspevkov), prípadne v akom zmysle (celková polarita príspevkov).

3.6.5 Zobrazenie najfrekventovanejších slov v príspevkoch

Vďaka extrakcii kľúčových slov z jednotlivých príspevkov môžeme byť schopní zobraziť početnosť konkrétnych slov v príspevkoch v čase a odhaliť tak napríklad kauzy týkajúce sa konkrétneho obchodného reťazca. Takéto niečo je dosiahnuteľné napríklad pomocou trendového grafu. V tomto prípade by vodorovná os reprezentovala čas (napríklad dni), tá zvislá zasa počet príspevkov. Neprekážalo by, keby bolo v grafe zobrazených viacero trendových čiar súčasne (pre každé kľúčové slovo jedna). Nakoľko ale bude počet vyextrahovaných kľúčových slov pravdepodobne pomerne veľký, takýchto trendových čiar by bolo z dôvodu prehľadnosti v grafe iba niekoľko. Ideálne iba pre kľúčové slová s najvyššími počtami výskytov.

Práve kvôli tomuto obmedzeniu by bolo vhodné umiestniť ku grafu tabuľku, ktorá by uvádzala i ďalšie (menej frekventované) kľúčové slová. Tá by tiež mohla obsahovať údaj o počte príspevkov, ktoré toto kľúčové slovo obsahujú, prípadne údaj o percentuálnom zastúpení takýchto príspevkov. V oboch prípadoch pre určité časové obdobie (napríklad pre konkrétny mesiac).

3.6.6 Filtrovanie výsledkov

Súčasťou vyššie spomenutých vizualizácií analýz by mohli byť i filtre. Pomocou nich by sa užívateľ mohol zamerať na vyhľadanie určitej konkrétnejšej informácie, keďže tieto filtre by boli schopné zmenšiť množinu zobrazených výsledkov analýz. Príkladom tejto funkcionality by mohlo byť filtrovanie výsledkov analýz podľa reťazca, ku ktorému sa príspevky vzťahujú, podľa časového obdobia (napríklad mesiaca), z ktorého príspevky pochádzajú, podľa aspektu spomenutého v príspevkoch, podľa lokality spomenutej v príspevkoch alebo podľa pôvodného zdroja dát príspevkov.

3.6.7 Súhrnné štatistiky

Zobrazenie súhrnných štatistík by tiež mohlo byť, rovnako ako použitie filtrov, súčasťou vizualizácií výsledkov analýz. Informovali by napríklad, z ktorého zdroja pochádza koľko príspevkov. Taktiež by v rámci týchto štatistík mohol byť uvedený údaj o pomere celkových polarít príspevkov z jednotlivých zdrojov.

3.6.8 Detail analýzy konkrétneho príspevku

Okrem vizualizácií, ktoré zobrazujú hromadné výsledky vykonaných analýz, by tiež bolo vhodné zobraziť podrobnosti výsledku analýzy konkrétneho príspevku. Išlo by o samostatnú obrazovku, ktorá by obsahovala text príspevku (s vizuálne odlíšenými aspektmi, názvami miest, ulíc a reťazcov), informáciu o jeho zdroji s prípadnou *URL* adresou (ak by bola k dispozícii), informácie o autorovi (meno, pohlavie, vek, počet príspevkov v rámci analyzovaných dát, miesto bydliska, *URL* adresu jeho profilu na sociálnej sieti alebo na webovej diskusii), čas jeho vytvorenia, názov reťazca, ktorého sa príspevok týka, prípadný zoznam spomenutých konkurenčných reťazcov a prípadný text spätnej väzby komunikačného oddelenia reťazca, ktorého sa príspevok týka (takýto text je možné získať iba v prípade, že ide o príspevok stiahnutý z nástenky oficiálnej stránky jedného z obchodných reťazcov na sociálnej sieti *Facebook*).

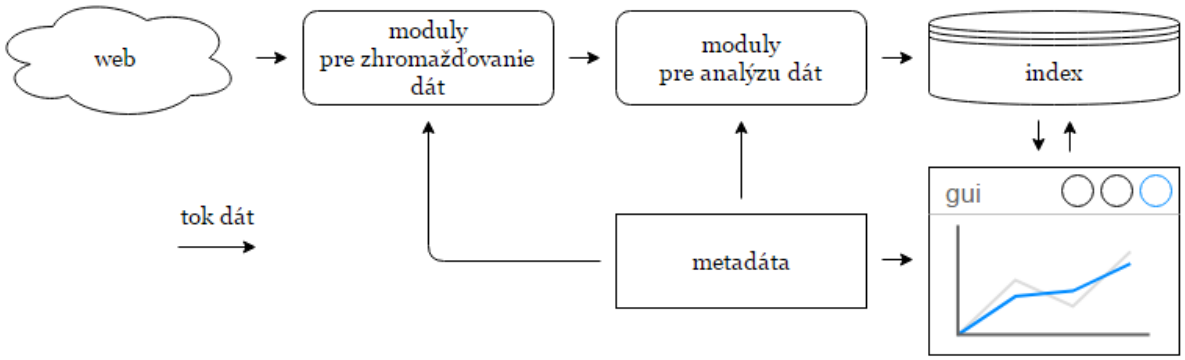
Zvyšok obrazovky by potom patril grafu, ktorý by sprostredkúval výsledok analýzy sentimentu na úrovni aspektu. Išlo by o variant stĺpcového grafu, ktorého vodorovná os by obsahovala názvy aspektov, o ktorých sa autor v príspevku zmieňuje. Táto os by bola vertikálne zarovnaná na stred grafu. Zvislá os grafu by mala tri diskrétne hodnoty – *negatívna polarita* (táto hodnota by sa nachádzala naspodku osi), *neutrálna polarita* (vertikálny stred osi) a *pozitívna polarita* (na vrchu osi).

Ak by teda analyzovaný príspevok zmieňoval konkrétny aspekt v negatívnom svetle, stĺpec by sa nachádzal pod vodorovnou osou. Ak by šlo o pozitívnu zmienku, stĺpec by bol umiestnený nad vodorovnou osou. V prípade neutrálnej zmienky by nebol zobrazený žiaden stĺpec (iba štítok s názvom aspektu pri vodorovnej osi). Za zváženie by tiež stálo zobraziť v rámci tohto grafu i údaj o celkovej polarite daného príspevku či farebne odlíšiť stĺpce reprezentujúce pozitívnu a stĺpce reprezentujúce negatívnu polaritu aspektov.

Obrazovka s detailom výsledku analýzy konkrétneho príspevku sa tiež javí ako ideálne miesto pre prípadnú úpravu tohto výsledku. Vhodným prostriedkom by mohlo byť modálne okno, ktoré by sa zobrazilo po stlačení konkrétneho tlačidla na obrazovke. Toto okno by obsahovalo textové pole s výsledkom analýzy príspevku v zrozumiteľnom formáte (viď podkapitolu 3.2.6), ktorý by bolo možné jednoducho upraviť a následne uložiť.

3.7 Návrhová schéma systému

Táto podkapitola je v istom slova zmysle akýmsi zhrnutím celej kapitoly. Práve okolnosti spomenuté naprieč kapitolou boli základom pri tvorbe schémy systému v návrhovej rovine. Túto schému ponúka Obrázok 3.1.



Obrázok 3.1: Návrhová schéma systému.

V schéme je možné vidieť, ako moduly pre zhromažďovanie dát zabezpečujú stiahnutie dát z webu (šípky medzi prvkami schémy symbolizujú tok dát). Tieto dáta následne putujú do modulov pre analýzu dát. Po analýze dát nasleduje ich indexovanie (šípka medzi prvkom reprezentujúcim moduly pre analýzu dát a prvkom reprezentujúcim index). Grafické užívateľské rozhranie (symbolizované oknom) potom pomocou dotazov (šípka smerom k prvku reprezentujúcemu index) môže získať indexované a analyzované dáta (šípka smerom od prvku reprezentujúceho index). Zo schémy môžeme taktiež vyčítať, že moduly pre sťahovanie dát, moduly pre analýzu dát a grafické užívateľské rozhranie využívajú pri svojej činnosti metadáta (viď šípky vychádzajúce z prvku reprezentujúceho metadáta).

Z dôvodu zachovania jednoduchej čitateľnosti schémy v nej nie je zahrnutý modul koordinujúci jednotlivé činnosti systému (viď podkapitolu 3.5). Ten by riadil moduly pre zhromažďovanie dát a moduly pre analýzu dát, zabezpečoval by tok dát medzi nimi a tiež by zabezpečoval indexovanie analyzovaných dát.

V schéme taktiež chýba (z rovnakých dôvodov ako v predchádzajúcom prípade) prvok automatizujúci činnosť systému. Jeho úlohou by bolo pravidelné spúšťanie koordinačného modulu.

4 Implementácia systému

Od popisu návrhu systému sa dostávame k popisu jeho implementácie. Systém je až na pár výnimiek (skripty pre bash) kompletne implementovaný v jazyku *Python* (vo verzii 2.7.6). Nasledujúce podkapitoly ponúkajú prehľad a popis balíkov a tried, z ktorých systém pozostáva.

4.1 Balík `data_collectors`

Tento balík združuje triedy zabezpečujúce najmä sťahovanie dát. Nachádzajú sa tu ale aj triedy, ktoré sú výnimkou. Konkrétne ide o triedu `FormatConverter` (slúži na konverziu formátov spracovávaných dát) a triedu `WarcAnalyser` (slúži na extrahovanie *URL* adries relevantných článkov zo súborov vo formáte *WARC*).

4.1.1 Trieda `SeleniumDataLoader`

Trieda `SeleniumDataLoader` je nadradenou triedam, ktoré pracujú s nástrojom *Selenium* (viď [21]). Ide o triedy `AccessTokenDownloader`, `FacebookUser`, `DiscussionDownloader` a `DiscussionUser`. Táto trieda uchováva cestu k súboru s prihlasovacími údajmi na sociálnu sieť *Facebook*, ktoré v rámci svojho konštruktoru načíta, a cestu k spustiteľnému súboru internetového prehliadača *Google Chrome* používaného práve nástrojom *Selenium*.

4.1.2 Skript `timeout.py`

Tento skript obsahuje definíciu dekorátoru `@timeout`. Jeho úlohou je dozerať na dobu behu funkcií, resp. metód tak, aby táto doba neprekročila stanovený limit (v sekundách). Dekorátor je využívaný triedami pracujúcimi s nástrojom *Selenium*.

4.1.3 Trieda `AccessTokenDownloader`

Ide o triedu, ktorá je priamym potomkom triedy `SeleniumDataLoader`. Metódy tejto triedy teda využívajú pri svojej činnosti nástroj *Selenium*. Poslaním tejto triedy je stiahnuť tzv. *User Access Token* (viď [20]), ktorý potom využíva trieda `FacebookDownloader` pri sťahovaní dát zo sociálnej siete *Facebook*. Dôvodom použitia nástroja *Selenium* je fakt, *User Access Token* je možné získať iba pomocou webového prehliadača, a to až po zadaní prihlasovacích údajov užívateľa sociálnej siete *Facebook*. Doba jeho platnosti je navyše obmedzená na cca dve hodiny.

4.1.4 Trieda `FacebookDownloader`

Trieda `FacebookDownloader` reprezentuje sťahovanie dát zo sociálnej siete *Facebook*, a to pomocou *Graph API* (viď[20]). Túto činnosť má na starosti metóda `getPosts`. Tá má jeden povinný parameter (názov stránky na sociálnej sieti *Facebook,* z nástenky ktorej sa budú sťahovať príspevky) a dva voliteľné parametre (prvým je dátum, z ktorého chceme získať príspevky – pri jeho vynechaní sú stiahnuté všetky príspevky z nástenky danej stránky; druhým parametrom je zasa počet dní pred zadaným dátumom, z ktorých príspevky treba kontrolovať na prítomnosť komentárov či odpovedí na komentáre vytvorených v zadanom dátume – bez zadania prvého parametra nemá tento parameter žiaden význam).

Metóda postupne zisťuje *id* stránky v rámci sociálnej siete *Facebook*, stiahne o nej dostupné informácie a následne sa venuje sťahovaniu samotných príspevkov, komentárov k príspevkom a odpovedí na komentáre k príspevkom. V rámci komentárov k príspevkom a odpovedí na komentáre k príspevkom sú tiež vyhľadávané a ukladané spätné väzby komunikačného oddelenia obchodného reťazca, z nástenky ktorého sú dáta zhromažďované.

4.1.5 Trieda `FacebookUser`

Ďalšou z tried, ktorá využíva nástroj *Selenium* (a taktiež je potomkom triedy `SeleniumDataLoader`), je trieda `FacebookUser`. Pomocou jej metódy `getInfoAboutUsers`, ktorej parametrom je zoznam číselných identifikátorov užívateľov v rámci sociálnej siete *Facebook*, je možné stiahnuť informácie o užívateľoch (ak sú k dispozícii) tejto sociálnej siete. Reč je o údaji o pohlaví, o mieste bydliska a o profiloch užívateľa na iných sociálnych sieťach, na fórach atď. Dôvod použitia nástroja *Selenium* je v tomto prípade ten, že pomocou *Graph API* nie sme schopní získať vyššie spomenuté údaje o užívateľoch.

4.1.6 Trieda `DiscussionDownloader`

Táto trieda je zodpovedná za stiahnutie diskusie k článku zo spravodajského portálu *aktualne.cz*. Pred samotným popisom implementácie tejto triedy si objasnime, prečo bol vybraný práve tento spravodajský portál.

Výstupom jedného z nástrojov vyvíjaných v rámci výskumnej skupiny *KNOT* na *Fakultě Informačních Technologií Vysokého učení technického v Brně* sú súbory vo formáte *WARC*. Tie obsahujú i *URL* adresy článkov najväčších českých spravodajských portálov. Ide o portály *aktualne.cz, idnes.cz* a *novinky.cz*. Tabuľka 4.1 uvádza početnosť relevantných článkov (teda takých, ktorých obsah súvisí s českými potravinovými obchodnými reťazcami) na jednotlivých spravodajských portáloch v období október 2016 – apríl 2017.

Rok	Mesiac	Počet relevantných článkov		
		aktualne.cz	*idnes.cz*	*novinky.cz*
2016	Október	7	0	5
	November	6	0	1
	December	7	2	4
2017	Január	9	0	4
	Február	10	0	2
	Marec	8	1	4
	Apríl	8	1	2
Spolu		55	4	22

Tabuľka 4.1: Počet *URL* adries relevantných článkov na jednotlivých spravodajských portáloch nachádzajúcich sa vo výstupoch interného projektu výskumnej skupiny *KNOT* na *Fakultě informačních technologií Vysokého učení technického v Brně*.

Z tabuľky je možné vyčítať, že počet *URL* adries relevantných článkov zo spravodajského portálu *aktualne.cz* je v spomínaných súboroch vo formáte *WARC* najvyšší. Práve to je dôvod, prečo boli diskusie k článkom tohto spravodajského portálu vybrané za jeden zo zdrojov dát.

Ako už bolo spomenuté vyššie, trieda `DiscussionDownloader` umožňuje stiahnutie diskusie ku konkrétnemu článku zo spravodajského portálu *aktualne.cz*. Toto sa deje prostredníctvom metódy `getDiscussionPosts`, ktorej parametrom je *URL* adresa článku. Z nej je získaná *URL* adresa diskusie k tomuto článku (pripojením reťazca „`v~diskuse/`“) a následne je diskusia stiahnutá. Súčasťou stiahnutých dát sú i informácie o článku – jeho titulok, jeho *URL* adresa, *URL* adresa jeho diskusie a počet príspevkov v jeho diskusii.

Pri sťahovaní diskusie je využitý nástroj *Selenium* (čo znamená, že trieda `DiscussionDownloader` je potomkom triedy `SeleniumDataLoader`). Dôvodom je dynamicky generovaný obsah stránky s diskusiou.

4.1.7 Trieda `DiscussionUser`

Trieda `DiscussionUser` reprezentuje sťahovanie informácií (ak má tieto informácie užívateľ vo svojom profile uvedené) o užívateľovi z diskusie k článku na spravodajskom portáli *aktualne.cz*. Ide o miesto bydliska a prípadný zoznam *URL* adries profilov tohto užívateľa na sociálnych sieťach, na fórach atď. Úkon samotného sťahovania informácií je možný prostredníctvom metódy `getUserInfo`, ktorej parametrom je *URL* adresa profilu užívateľa v rámci diskusie (tú je možné získať z dát stiahnutých triedou `DiscussionDownloader`, resp. jej metódou `getDiscussionPosts`).

Trieda `DiscussionUser` je potomkom triedy `SeleniumDataLoader`, čo znamená, že pri sťahovaní dát používa nástroj *Selenium*. Deje sa tak kvôli dynamicky generovanému obsahu stránky s profilom užívateľa.

4.1.8 Trieda `FormatConverter`

Keďže dáta stiahnuté zo sociálnej siete *Facebook* a dáta stiahnuté z diskusie k článku zo spravodajského portálu *aktualne.cz* majú rozdielny formát, potrebujeme prostriedok umožňujúci dostať tieto dáta do rovnakého alebo podobného formátu, a to preto, aby sme s nimi neskôr mohli zaobchádzať rovnakým či podobným spôsobom. Takýmto prostriedkom je trieda `FormatConverter`. Tá je schopná dáta z dvoch spomenutých zdrojov konvertovať do dvoch rozdielnych formátov.

Prvým cieľovým formátom je formát *JSON-LD* (viď[22]). Poslaním tohto formátu je vytvoriť koncept prepojených a strojovo čitateľných dát naprieč webom. Formát *JSON-LD* je mierne odlišný pre dáta pochádzajúce zo sociálnej siete *Facebook* a pre dáta pochádzajúce z diskusií k článkom spravodajského portálu *aktualne.cz*.

Metóda `getJsonLdFromFacebook` konvertuje do formátu *JSON-LD* dáta stiahnuté zo sociálnej siete *Facebook*. Výsledný formát má potom nasledujúci tvar:

```
{
      "@context": "http://schema.org",
      "@type": "SocialMediaPosting",
      "url": "...",
      "text": "...",
      "dateCreated": "...",
      "about": {
            "@type": "Organization",
            "name": "...",
            "url": "..."
      },
      "author": {
            "@type": "Person",
            "name": "...",
            "url": "..."
      },
      "comment": [
            ...
      ]
}
```

V tejto dátovej štruktúre sa hodnota poľa `@context` a hodnoty jednotlivých polí `@type` nemenia. Pre každý príspevok je ale špecifická informácia o jeho *URL* adrese (pole `url`), o jeho texte (pole `text`), o dátume jeho vytvorenia (pole `dateCreated`), o obchodnom reťazci, z ktorého oficiálnej stránky na sociálnej sieti *Facebook* bol príspevok stiahnutý (o jeho názve

a webovej stránke – polia `name` a `url` v rámci štruktúry, ktorá je hodnotou poľa `about`), o jeho autorovi (o jeho mene a *URL* adrese jeho profilu na sociálnej sieti *Facebook* – polia `name` a `url` v rámci štruktúry, ktorá je hodnotou poľa `author`) a tiež je evidovaný zoznam prípadných komentárov k tomuto príspevku (pole `comment`). Komentáre majú rovnaký formát ako formát príspevku uvedený vyššie až na pár výnimiek. Hodnota poľa `@type` je „`Comment`" a táto položka neobsahuje pole `about`. Komentár, rovnako ako príspevok, môže obsahovať pole `comment`. Je teda možný prípadný vznik zanorenej hierarchie komentárov.

Konverziu dát stiahnutých z diskusií k článkom spravodajského portálu *aktualne.cz* do formátu *JSON-LD* poskytuje metóda `getJsonLdFromDiscussion`. Výstup tejto konverzie je potom nasledovný:

```
{
      "@context": "http://schema.org",
      "@type": "Article",
      "url": "...",
      "discussionUrl": "...",
      "headline": "...",
      "comment": [
            {
                  "@type": "Comment",
                  "text": "...",
                  "dateCreated": "...",
                  "author": {
                        "@type": "Person",
                        "name": "...",
                        "url": "..."
                  },
                  "comment": [
                        ...
                  ]
            },
            ...
      ]
}
```

Rovnako ako v prípade príspevkov zo sociálnej siete *Facebook*, i v tomto prípade zostáva hodnota poľa `@context` a hodnoty konkrétnych polí `@type` konštantné. Okrem nich je v takomto formáte zahrnutá informácia o *URL* adrese článku (pole `url`), o *URL* adrese diskusie k tomuto článku (pole `discussionUrl`), o nadpise článku (pole `headline`) a prostredníctvom poľa `comment` i zoznam príspevkov v diskusií k danému článku. Položka tohto zoznamu obsahuje text príspevku (pole `text`), dátum jeho vytvorenia (pole `dateCreated`), informácie o jeho autorovi (o jeho mene a *URL* adrese profilu v rámci diskusie – polia `name` a `url` v rámci štruktúry, ktorá je hodnotou poľa `author`) a prípadný zoznam komentárov k príspevku (pole `comment`), ktoré majú rovnaký formát ako formát práve popisovanej položky.

Druhým z formátov, do ktorých umožňuje trieda `FormatConverter` dáta konvertovať, je formát oddeľujúci informácie o príspevkoch od informácií o ich autoroch. Dáta v pôvodnom formáte sú teda v tomto prípade rozdelené na dve časti – na zoznam informácií o jednotlivých príspevkoch a na zoznam informácií o ich autoroch.

Položka zoznamu informácií o jednotlivých príspevkoch obsahuje polia s informáciami známymi už v dobe stiahnutia príspevku a polia nachystané pre uloženie výsledkov analýz a extrakcií informácií, ktorým sa príspevok podrobí. Jej formát (bez výsledkov analýz a extrakcií informácií) je nasledovný:

```
{
      "aboutBrand": "...",
      "dateCreated": "...",
      "aspectsMentioned": [],
      "aspectsEvaluated": {},
      "branchAddress": "",
      "region": "region unknown",
      "anyReaction": False,
      "reactionText": "",
      "originalText": "...",
      "xmlFormat": "",
      "htmlFormat": "",
      "source": "...",
      "url": "...",
      "overallSentiment": 0,
      "competition": [],
      "authorId": "...",
      "authorAge": -1,
      "keywords": []
}
```

Evidovaný je názov reťazca, ktorého sa príspevok týka (pole `aboutBrand`), čas jeho vytvorenia (`dateCreated`), zoznam prípadných aspektov spomenutých v príspevku (`aspectsMentioned`), slovník obsahujúci tieto aspekty spolu s ohodnotením ich polarity (`aspectsEvaluated`), prípadne spomenutá adresa pobočky alebo mesto (`branchAddress`), kraj, v ktorom sa pobočka alebo mesto nachádza (`region`), indikátor prítomnosti spätnej väzby komunikačného oddelenia reťazca, ktorého sa príspevok týka (`anyReaction`), prípadný text tejto reakcie (`reactionText`), pôvodný text príspevku (`originalText`), výsledok analýzy príspevku vo formáte *XML* (`xmlFormat`), príspevok vo formáte *HTML* s vizuálne odlíšenými spomenutými aspektmi, názvami miest, ulíc a názvov reťazcov (`htmlFormat`), zdroj pôvodného príspevku (`source`), *URL* adresa príspevku (`url`), celková polarita príspevku (`overallSentiment`), zoznam prípadne spomenutých názvov konkurenčných reťazcov (`competition`), identifikátor autora v rámci indexu (`authorId`), vek autora (`authorAge`) a kľúčové slová (`keywords`). Pri návrhu nástroja sa počítalo s tým, že v rámci zdrojov dát bude k dispozícii i údaj o dátume

narodenia autora (na základe autorovho veku malo byť možné filtrovať vizualizácie v grafickom užívateľskom rozhraní). Keďže táto informácia nie je dostupná ani v profiloch užívateľov na sociálnej sieti *Facebook* a ani v profiloch užívateľov z diskusie k článkom spravodajského portálu *aktualne.cz*, pole `authorAge` nikdy nenadobudne hodnotu skutočného veku autora. Využiteľné ale môže byť pri prípadných budúcich rozšíreniach systému.

Položka zoznamu informácií o autoroch príspevkov obsahuje meno daného autora príspevku a polia nachystané pre uloženie prípadných dodatočne stiahnutých informácií o tomto autorovi. Formát takejto položky (bez dodatočne stiahnutých informácií) je nasledovný:

```
{
      "name": "...",
      "gender": "",
      "birthDate": "",
      "from": "",
      "postsCount": 1,
      "profiles": [],
      "reputation": {}
}
```

Ako môžeme vidieť, položka obsahuje meno autora príspevku (pole `name`), jeho pohlavie (`gender`), jeho dátum narodenia (`birthDate`), miesto jeho bydliska (`from`), počet príspevkov vytvorených autorom v rámci indexu (`postsCount`), zoznam *URL* adries profilov autora naprieč webom (`profiles`) a slovník (`reputation`) obsahujúci reputácie autora (hodnoty slovníku) na týchto weboch (kľúče slovníku). Ako už bolo spomenuté vyššie, s údajom o dátume narodenia autora príspevku sa počítalo v dobe návrhu systému, no reálne nie sme schopní tento údaj získať. V dobe implementácie sťahovania dát o užívateľovi z diskusií k článkom spravodajského portálu *aktualne.cz* tiež bola k dispozícii informácia o reputácií tohto užívateľa. Po čase tak tomu ale, žiaľ, už nie je. Pole `birthdayDate` a pole `reputation` teda zatiaľ zostáva v aktuálnom stave systému nevyužité. Tieto polia by ale mohli mať opodstatnenie pri prípadných budúcich rozšíreniach systému.

Informácie o príspevkoch a informácie o užívateľoch sú po doplnení výsledkov analýz príspevkov, resp. údajov o užívateľoch indexované práve vo vyššie popísaných formátoch. Tie boli navrhnuté tak, aby dotazy zo strany grafického užívateľského rozhrania zobrazujúceho výsledky analýz mohli byť čo najjednoduchšie. Prevod stiahnutých dát do týchto formátov vykonáva metóda `getElasticsearchFromFacebook` (v prípade, že ide o dáta stiahnuté zo sociálnej siete *Facebook*), resp. `getElasticsearchFromDiscussion` (v prípade, že ide o dáta získané z diskusie k článku na spravodajskom portáli *aktualne.cz*).

4.1.9 Trieda `WarcAnalyser`

Trieda `WarcAnalyser` reprezentuje nástroj pre evidenciu histórie relevantných článkov (týkajúcich sa českých potravinových obchodných reťazcov) z vybraných spravodajských portálov (*aktualne.cz*, *idnes.cz* a *novinky.cz*). Prostredníctvom metódy `doUpdateArticlesHistory` je spracovaný konkrétny súbor vo formáte *WARC* (predpokladom je, že obsah tohto súboru sa vzťahuje ku konkrétnemu dátumu, ktorý je súčasťou jeho názvu). Z tohto súboru sú extrahované relevantné *URL* adresy článkov. Ich relevantnosť je podmienená prítomnosťou aspoň jedného z názvov obchodných reťazcov v titulku článku, na ktorý odkazujú. Bez ohľadu na to, či boli v danom *WARC* súbore nájdené *URL* adresy relevantných článkov alebo nie, pre dátum, z ktorého pochádza obsah tohto súboru, je vytvorený záznam v akomsi archíve relevantných článkov. Takýmto archívom je *JSON* súbor s nasledujúcim všeobecným formátom:

```
{
      "YYYY-MM-DD": [
            ...
      ],
      ...
}
```

Kľúč `YYYY-MM-DD` je v praxi nahradený konkrétnym dátumom a hodnotou tohto kľúča je zoznam takých *URL* adries relevantných článkov, ktoré sa v dobe vytvorenia kľúča v tomto archíve ešte nevyskytujú. Môže sa teda stať, že hodnotou kľúča reprezentujúceho konkrétny dátum bude prázdny zoznam. To znamená, že vo *WARC* súbore vzťahujúcemu sa k tomuto dátumu neboli nájdené žiadne *URL* adresy relevantných článkov alebo nájdené boli také *URL* adresy relevantných článkov, ktoré sú v rámci archívu už evidované. Takýto archív teda ponúka množinu *URL* adries relevantných článkov spolu s informáciou o dátume prvého výskytu každého článku v analyzovaných *WARC* súboroch.

4.1.10 Ostatné skripty

Súčasťou balíka `data_collectors` sú i dva ďalšie skripty. Prvý z nich, skript `articles_history_update.py`, riadi aktualizáciu archívu *URL* adries relevantných článkov. Jeho jediným argumentom je názov *WARC* súboru, z ktorého sú prostredníctvom volania metódy `doUpdateArticlesHistory` triedy `WarcAnalyser` prípadne extrahované a uložené *URL* adresy relevantných článkov (takých, ktorých obsah sa týka českých potravinových obchodných reťazcov).

Skript `run_articles_history_update.sh` zisťuje dátum predchádzajúceho dňa (vzhľadom ku dňu, v ktorom je spustený), na základe dátumu v názve získava *WARC* súbor s archívom z toho dátumu (zo zložky jedného z interných projektov výskumnej skupiny *KNOT*

na *Fakultě informačních technologií Vysokého učení technického v Brně*) a následne spúšťa aktualizáciu archívu *URL* adries relevantných článkov o prípadné extrahované *URL* adresy relevantných článkov z tohto *WARC* súboru (spúšťa teda skript `articles_history_update.py` s názvom tohto *WARC* súboru ako argumentom skriptu). Pravidelné spúšťanie skriptu `run_articles_history_update.sh` môže zabezpečiť `cron` (nástroj pre správu úloh).

4.2 Balík `analysers`

V balíku `analysers` sa nachádzajú triedy schopné vykonávať analýzu dát či extrahovať z nich informácie. Okrem takýchto tried sa tiež v balíku nachádza trieda reprezentujúca klasifikátor, ktorý je základným stavebným kameňom analýz (trieda `Classifier`), trieda vyhodnocujúca výsledky klasifikácie (trieda `ClassificationEvaluator`), trieda poskytujúca operácie pre predspracovanie textu (trieda `TextPreprocessor`) a trieda poskytujúca operácie pre prácu s výsledkami jednotlivých analýz a extrakcií v *XML* formáte (trieda `XmlProcessor`).

4.2.1 Trieda `TextPreprocessor`

Trieda `TextPreprocessor` reprezentuje predspracovanie textu. Práve to využívajú všetky triedy vykonávajúce analýzu dát. Jej metódy sú schopné rozdeliť text na jednotlivé vety, jednotlivé vety zasa na slová, tieto slová transformovať do základného a koreňového tvaru, zo zoznamu získaných slov odstrániť stop slová, odstrániť zo slov diakritiku či konvertovať tieto slová tak, aby boli tvorené iba malými písmenami. Metóda `getProcessedText` združuje vyššie spomenuté operácie do jedného celku. Jej vstupom je text, ktorý chceme predspracovať, spolu s voľbou jednotlivých operácií, ktoré sa majú pri jeho predspracovaní uplatniť. Výstupom tejto metódy je predspracovaný text.

Pri delení textu na vety a slová používa táto trieda balík `polyglot` (viď [23]). Prevod slov do základného a koreňového tvaru a odstránenie stop slov majú na starosti balíky popísané nižšie. Odstránenie diakritiky zabezpečuje balík `unidecode` (viď [24]) a transformáciu slov tak, aby boli tvorené malými písmenami, zasa štandardná funkcia jazyka *Python* (funkcia `lower()`).

Pomocou balíku `lemmatizer` (súčasť balíku `analysers`) sme schopní získať slovo v jeho základnom tvare. Túto operáciu poskytuje trieda `Lemmatizer`, konkrétne jej metóda `getLemma`. Trieda `Lemmatizer` využíva pri prevode slova na jeho základný tvar nástroj *MorphoDiTa* (viď [25]).

V balíku `stemmer` (súčasť balíku `analysers`) sa nachádza jediná trieda – trieda s názvom `Stemmer`. Jej metóda `getStem` vracia kmeňový tvar pôvodného slova, ktoré je parametrom tejto metódy, a to pomocou nástroja, ktorý je súčasťou balíku `sumy` (viď [26]).

Trieda `Stopwords` z balíku `stopwords` (súčasť balíku `analysers`) združuje zoznamy stop slov nachádzajúce sa v textových súboroch v rámci tohto balíku (zdroje jednotlivých stop slov sú uvedené v súbore `README.txt`, ktorý je súčasťou balíku). Metóda `getStopwords` triedy `Stopwords` vracia množinu týchto stop slov.

4.2.2 Trieda `Classifier`

Klasifikátor, ktorý je základom analýzy sentimentu na úrovni dokumentu i na úrovni aspektu, reprezentuje trieda `Classifier`. Tá vo svojom konštruktore vytvára model z poskytnutej tréningovej sady. Povinným parametrom konštruktora je práve tréningová sada a voliteľnými parametrami sú okrem zvolenej metódy klasifikácie i voľby predspracovania tréningovej sady. V rámci tejto triedy je možné použiť tri klasifikačné metódy – naivnú Bayesovskú klasifikáciu (použitá je jej implementácia z balíku `textblob`, viď [27]), klasifikačnú metódu s názvom *Maximum Entropy* (použitá je jej implementácia z balíku `nltk`, viď [28]) a klasifikačnú metódu *SVM* (použitá je jej implementácia z balíku `sklearn`, viď [29]). Príznakom klasifikácie je indikátor prítomnosti konkrétneho slova v texte. Výsledok klasifikácie textu môžeme získať prostredníctvom metódy `getClassificationResult`, ktorej parametrom je klasifikovaný text. K dispozícii je tiež metóda poskytujúca pravdepodobnosť príslušnosti klasifikovaného textu k jednotlivým triedam (metóda `getClassificationProbabilities`) a metódy poskytujúce detaily vytvoreného modelu (metóda `doShowInformativeFeatures`) alebo klasifikácie konkrétneho textu (metóda `doExplainClassification`).

4.2.3 Trieda `ClassificationEvaluator`

Prostriedkom analýzy výsledkov klasifikácií je trieda `ClassificationEvaluator`. Jej metóda `getConfusionMatrix` vracia maticu zmätenia pre príslušné referenčné a klasifikované dáta.

4.2.4 Trieda `PolarityAnalyser`

Keďže neexistuje voľne dostupný nástroj na analýzu sentimentu na úrovni dokumentu v českom jazyku a keďže potrebujeme takýto nástroj, ktorý je doménovo špecifický, bolo treba tento nástroj implementovať. Reprezentuje ho trieda `PolarityAnalyser`.

Analýza sentimentu na úrovni dokumentu (dokument v tomto kontexte chápeme ako konkrétny príspevok so sociálnej siete alebo diskusie k článku) je v prípade tejto triedy založená na klasifikácii (využitá je teda vyššie popísaná trieda `Classifier`). Vytvorenie modelu klasifikátoru, jeho uloženie do súboru a jeho načítanie zo súboru implementujú metódy `doCreateClassifierModel`, `doSaveClassifierModel`, resp.

`doLoadClassifierModel`. Odhadovanú polaritu konkrétneho príspevku vracia metóda `getPostPolarity`, ktorej parametrom je text tohto príspevku.

Tréningovou sadou klasifikátoru, ktorý používa trieda `PolarityAnalyser`, je spolu 1200 príspevkov zo sociálnej siete *Facebook*. Konkrétne ide o 150 príspevkov z násteniek oficiálnych stránok ôsmych českých potravinových obchodných reťazcov (*Albert*, *Billa*, *Globus*, *Kaufland*, *Lidl*, *Makro*, *Penny Market* a *Tesco*). Tie boli pri anotovaní radené do troch tried, a to na základe toho, či je ich celková polarita kladná, neutrálna alebo negatívna. Tréningová sada je uložená v súbore vo formáte *JSON*.

4.2.5 Trieda `AspectBasedSentimentAnalyser`

Dôvody pre implementáciu nástroja vykonávajúceho analýzu sentimentu na úrovni aspektu (ktorý reprezentuje táto trieda) sú rovnaké ako v prípade nástroja poskytujúceho analýzu sentimentu na úrovni dokumentu (trieda `PolarityAnalyser`). V rámci analýzy sentimentu na úrovni aspektu používa trieda `AspectBasedSentimentAnalyser` predspracovanie textu (prostredníctvom triedy `TextPreprocessor`) a klasifikáciu (prostredníctvom triedy `Classifier`). Táto analýza prebieha nasledujúcim spôsobom.

Analyzovaný príspevok je najprv rozdelený na vety. Každá z týchto viet je potom podrobená samostatnej analýze (to vlastne znamená, že výsledkom analýzy sentimentu príspevku na úrovni aspektu je množina výsledkov analýz jednotlivých viet tohto príspevku). Pri analýze konkrétnej vety je táto veta rozdelená na slová a slová sú zasa prevedené do ich základného tvaru. V rámci týchto slov sú vyhľadávané vopred dané kľúčové slová príznačné pre jednotlivé aspekty (tzv. *aspect terms*), napríklad *kvalita*, *cena*, *fronta* atď. Veta sa následne podrobí klasifikácii. Pre každý aspekt je vopred vytvorený model klasifikátoru, ktorý ohodnocuje, či sa veta tohto aspektu týka alebo nie. Veta sa totiž môže týkať aspektu i keď ho explicitne nespomína (napríklad veta *„Čakanie na platbu bolo nekonečné“* sa v kontexte obchodných reťazcov s najväčšou pravdepodobnosťou týka *fronty*). Ak bolo zistené, že veta sa týka aspoň jedného aspektu (či už pomocou prítomnosti slova reprezentujúceho aspekt alebo pomocou klasifikácie), veta prejde ďalším procesom klasifikácie. Jeho úlohou je určiť, akú polaritu má veta z pohľadu aspektov, ktorých sa týka. Toto majú na starosti taktiež vopred vytvorené modely klasifikátorov. Takýto model klasifikátoru je vytvorený pre každý aspekt.

Po vykonaní vyššie popísaného procesu máme teda k dispozícii informáciu o prípadných aspektoch, ktorých sa veta, resp. celý príspevok týka i o ich polaritách. Takýto proces implementuje metóda `getAnalysisResult`. Tá vracia výsledok analýzy i vo formáte *XML*, ktorý popisuje tento *DTD* (*Document Type Definition*):

```
<!DOCTYPE post [

<!ELEMENT post (sentence+)>
<!ELEMENT sentence (text,aspectTerms?,aspectCategories?)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT aspectTerms (aspectTerm+)>
<!ELEMENT aspectTerm EMPTY>
<!ELEMENT aspectCategories (aspectCategory+)>
<!ELEMENT aspectCategory EMPTY>

<!ATTLIST aspectTerm term CDATA #REQUIRED>
<!ATTLIST aspectTerm polarity (positive|neutral|negative) "neutral">
<!ATTLIST aspectTerm from CDATA #REQUIRED>
<!ATTLIST aspectTerm to CDATA #REQUIRED>
<!ATTLIST aspectCategory category CDATA #REQUIRED>
<!ATTLIST aspectCategory polarity (positive|neutral|negative)
"neutral">

]>
```

Inšpiráciou pre použitie tohto formátu je jeho výskyt v súťaži *SemEval* (viď [30]). Príspevok (element `post`) je pomocou neho rozdelený na vety (element `sentence`). Okrem vlastného textu (element `text`) môže veta obsahovať konkrétne slová reprezentujúce aspekty (element `aspectTerm`, potomok elementu `aspectTerms`), pri ktorých je spolu s tvarom, v ktorom sa tieto slová vo vete nachádzajú (atribút `term` elementu `aspectTerm`), uvedená i polarita týchto slov (atribút `polarity` elementu `aspectTerm`) a ich počiatočný a koncový index v rámci tejto vety (atribúty `from`, resp. `to` elementu `aspectTerm`). Taktiež môžu byť uvedené aspekty, ktorých sa veta týka (atribút `category` elementu `aspectCategory`, ktorý je potomkom elementu `aspectCategories`), a ich polarita (atribút `polarity` elementu `aspectCategory`).

Za zmienku ešte stoja metódy `doCreateClassifierModels`, `doSaveClassifierModels` a `doLoadClassifierModels`. Tie zabezpečujú prácu (vytvorenie, uloženie do súboru, resp. načítanie zo súboru) s modelmi vyššie spomenutých dvoch druhov klasifikátorov, ktoré sú pri analýze použité.

Aspekty, ktoré sú v rámci analýzy sentimentu na úrovni aspektu hľadané, boli zvolené na základe ručnej analýzy 400 príspevkov zo sociálnej siete *Facebook* (50 príspevkov z nástenky oficiálnej stránky každej z ôsmych českých potravinových obchodných reťazcov, ktorými sú *Albert*, *Billa*, *Globus*, *Kaufland*, *Lidl*, *Makro*, *Penny Market* a *Tesco*). Výsledky tejto analýzy uvádza Tabuľka 4.2.

Aspekt	Percentuálne zastúpenie v príspevkoch
Kvalita tovaru	21.5 %
Množstvo tovaru v predajni	14.0 %
Personál	13.3 %
Cena tovaru	7.5 %
Rad pri pokladni	5.5 %
Reklama	5.3 %
Vernostné programy a dlhodobé služby	3.8 %
Komunikácia prostredníctvom oficiálnych kanálov	1.8 %

Tabuľka 4.2: Percentuálne zastúpenie aspektov v ručne analyzovanej vzorke príspevkov.

Pri analýze sentimentu na úrovni aspektu sú teda hľadané a ohodnocované práve aspekty uvedené v tejto tabuľke. Pri anotovaní dát pre klasifikátory popísanom nižšie bol z dôvodu jeho častého výskytu (jeho percentuálne zastúpenie nie je k dispozícii, keďže anotovanie prebiehalo spôsobom, z ktorého nie je možné tento údaj spätne zistiť) doplnený ešte jeden aspekt. Tento aspekt možno nazvať *ponuka a krátkodobé akcie*.

Poďme sa pozrieť na tréningové sady klasifikátorov, ktoré sú používa analýza sentimentu na úrovni aspektu. Presnejšie povedané na proces ich vzniku. V rámci neho bolo na vety rozdelených 1200 príspevkov zo sociálnej siete *Facebook* (ich detailnejší pôvod je popísaný v podkapitole 4.2.4). Vzniklo tak 4094 viet. Tieto vety boli následne ručne rozdelené do tried. Triedy reprezentovali jednotlivé aspekty, takže veta bola zaradená do konkrétnej triedy na základe toho, či sa jej obsah týkal aspektu, ktorý táto trieda reprezentuje. Konkrétna veta mohla patriť do viacerých tried súčasne (mohla napríklad zmieňovať kvalitu tovaru i jeho cenu). Pre vety, ktorých obsah sa nevzťahoval k žiadnemu z aspektov, bola vytvorená špeciálna trieda, do ktorej boli zaradené. O vetách z tejto triedy budeme ďalej hovoriť ako o irelevantných. Výsledný počet viet v jednotlivých triedach uvádza Tabuľka 4.3.

Trieda	Počet viet
Irelevantné vety	2471
Ponuka a krátkodobé akcie	430
Kvalita tovaru	418
Personál	384
Množstvo tovaru v predajni	221
Cena tovaru	152
Reklama	135
Vernostné programy a dlhodobé služby	130
Rad pri pokladni	82
Komunikácia prostredníctvom oficiálnych kanálov	26

Tabuľka 4.3: Počet viet týkajúcich sa jednotlivých aspektov reprezentovaných triedami.

Takto roztriedené vety boli základom pre vznik tréningových sád (tie sú uložené vo formáte *JSON*). Tréningová sada pre klasifikátory určujúce, či sa veta týka daného aspektu, potom pozostávala z viet triedy reprezentujúcej tento aspekt, ktoré boli automaticky anotované ako *týka sa aspektu*. Tie boli doplnené o rovnaký počet náhodne vybraných viet z ostatných tried, ktoré boli automaticky anotované ako *netýka sa aspektu*. Keďže jedna veta sa mohla nachádzať vo viacerých triedach, bolo treba zaistiť, aby sa takáto veta v tréningovej sade nenachádzala dvakrát – raz anotovaná ako *týka sa aspektu*, raz anotovaná ako *netýka sa aspektu*. Tento typ tréningových sád bol generovaný automaticky, a tréningová sada bola vytvorená i pre triedu s irelevantnými vetami.

Tréningové sady pre klasifikátory určujúce polaritu vety týkajúcej sa konkrétneho aspektu boli anotované ručne. Tréningovú sadu pre klasifikátor daného aspektu potom tvoria vety z triedy reprezentujúcej tento aspekt. Každá z viet bola anotovaná jednou z hodnôt *pozitívna*, *neutrálna* alebo *negatívna*.

Na záver si popíšme proces vzniku zoznamov slov príznačných pre jednotlivé aspekty. Ide o nadväznosť na vyššie popísané delenie viet do tried reprezentujúcich jednotlivé aspekty. V každej z týchto tried boli vety rozdelené na slová a tie boli prevedené do základného tvaru. Z takto získaného zoznamu slov boli manuálne vybrané slová príznačné pre daný aspekt. Uložené sú vo formáte *JSON*.

4.2.6 Trieda `BrandNamesExtractor`

Trieda `BrandNamesExtractor` reprezentuje nástroj pre extrakciu názvov obchodných reťazcov z textu. Referenčný zoznam názvov reťazcov je získaný od triedy `Metadata` z balíku `metadata`. Výsledok extrakcie vracia metóda `getAnalysisResult`. V rámci nej je analyzovaný príspevok rozdelený na vety a prípadné názvy obchodných reťazcov sú extrahované z každej vety zvlášť. Toto

sa deje pomocou rozdelenia vety na slová a následného porovnávania kmeňového tvaru jednotlivých slov s kmeňovými tvarmi názvov obchodných reťazcov z referenčného zoznamu. Porovnávanie pomocou kmeňového tvaru slova dostalo v tomto prípade prednosť pred porovnávaním pomocou základného tvaru slova z dôvodu nedostatkov nástroja *MorphoDiTa* (použitého pre prevod slov do ich základného tvaru) pri určovaní základného tvaru názvov niektorých obchodných reťazcov (nedokáže určiť základný tvar slova *Billa* z jeho rôznych pádov, slovu *Globus* v rôznych pádoch chybne priraďuje základný tvar *glóbus*).

Spomínaná metóda `getAnalysisResult` vracia výsledok extrakcie i vo formáte *XML*. *DTD* tohto formátu je nasledovný:

```
<!DOCTYPE post [

<!ELEMENT post (sentence+)>
<!ELEMENT sentence (text,brands?)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT brands (brand+)>
<!ELEMENT brand EMPTY>

<!ATTLIST brand name CDATA #REQUIRED>
<!ATTLIST brand term CDATA #REQUIRED>
<!ATTLIST brand from CDATA #REQUIRED>
<!ATTLIST brand to CDATA #REQUIRED>

]>
```

V tomto formáte je príspevok (element `post`) rozdelený na vety (element `sentence`). Element reprezentujúci vetu obsahuje jej text (element `text`) a prípadný zoznam extrahovaných názvov obchodných reťazcov (element `brands` obsahujúci aspoň jeden element `brand`). Položka tohto zoznamu potom obsahuje názov reťazca v základnom tvare (atribút `name` elementu `brand`), jeho tvar nachádzajúci sa vo vete (atribút `term`) a počiatočný a koncový index tohto tvaru v texte (atribút `from` a atribút `to`).

4.2.7 Trieda `KeywordsExtractor`

Nástroj pre extrahovanie kľúčových slov z príspevkov reprezentuje trieda `KeywordsExtractor`. Zoznam kľúčových slov konkrétneho príspevku vracia metóda `getPostKeywords`, ktorej parametrom je práve tento príspevok. Príspevok je v rámci metódy rozdelený na samostatné slová, tie sú prevedené do základného tvaru a následne sú transformované tak, aby ich tvorili iba malé písmená. Spomedzi týchto slov sú taktiež odstránené stop slová spolu s ďalšími vybranými slovami. Je to opodstatnené tým, že na základe najfrekventovanejších kľúčových slov v príspevkoch je vytvorený jeden z grafov v rámci vizualizácie výsledkov analýz. Ten by mal odhaľovať, o čom sa najčastejšie diskutuje. Ak v ňom chceme mať zobrazené čo najrelevantnejšie výsledky, musíme

okrem stop slov odstrániť i niekoľko ďalších slov (napríklad slovo *človek*, ktoré má v doméne českých potravinových obchodných reťazcov v porovnaní so slovom *cena* menšiu výpovednú hodnotu).

Základom pre vytvorenie zoznamu týchto slov bolo 1200 príspevkov zo sociálnej siete *Facebook* (pre bližšie informácie viď podkapitolu 4.2.4). Tieto príspevky boli rozdelené na slová, slová boli prevedené do základného tvaru a konvertované tak, aby ich tvorili iba malé písmená. Takýto zoznam slov bol usporiadaný podľa početnosti jednotlivých slov v pôvodných dátach (od slov s najvyššou početnosťou po tie s najnižšou), a to z dôvodu uľahčenia výberu vhodných slov. Z tohto zoznamu boli potom adekvátne slová vybrané manuálne. Na základe výskytu vo vyššie spomenutom grafe k nim bolo neskôr doplnených i zopár ďalších slov, ktoré boli pre účely grafu irelevantné. Zoznam takýchto slov je uložený vo formáte *JSON*.

4.2.8 Trieda `LocationsExtractor`

Extrahovanie názvov miest a ulíc (adries pobočiek) z príspevkov má na starosti trieda `LocationsExtractor`, konkrétne jej metóda `getAnalysisResult`. Tá najprv rozdelí príspevok na prípadné vety a extrakcii je podrobená každá z viet samostatne. V rámci tejto extrakcie je veta rozdelená na jednotlivé slová a tie sú prevedené do ich základného tvaru. V tomto zozname slov sa následne hľadajú zhody so sekvenciami reprezentujúcimi jednotlivé mestá, resp. ulice.

Poďme sa bližšie pozrieť na takéto sekvencie. Názvy miest a ulíc môžu byt viacslovnými pomenovaniami. Práve preto sú pred porovnávaním transformované na spomínané sekvencie. Takáto sekvencia je tvorená jednotlivými slovami tvoriacimi názov mesta alebo ulice, a to v základom tvare (ak ide o jednoslovný názov, výsledkom je sekvencia dĺžky jedna). V prípadných nájdených sekvenciách musí tiež dôjsť k zhode malých, resp. veľkých písmen na začiatkoch jednotlivých položiek sekvencie a k nim príslušných slov vo vete. Ak pôvodný názov ulice obsahuje i číslo (čo väčšinou obsahuje, keďže ide o adresu pobočky) toto číslo je zo sekvencie odstránené a porovnávať sa nebude.

Databáza názvov miest a adries pobočiek jednotlivých obchodných reťazcov bola vytvorená nasledovne. Zoznam miest, resp. obcí a im prislúchajúcich krajov bol získaný z portálu [31]. Zoznam adries (obsahujúcich i názov mesta) pobočiek jednotlivých reťazcov zasa z oficiálnych stránok týchto reťazcov. Ak sa zoznam pobočiek na oficiálnych stránkach reťazca nenachádzal, ako zdroj tohto zoznamu poslúžil portál [32]. Tieto dva druhy zoznamov boli pomocou skriptu zlúčené dokopy. Výsledok tohto zlúčenia je uložený v *JSON* súbore vo formáte:

```
{
      "názov_kraja": {
            "názov_mesta": {
                  "názov_reťazca": [
                        ...
                  ],
                  ...
            },
            ...
      },
      ...
}
```

V rámci tohto formátu teda vzniká hierarchia reflektujúca skutočnosť. Jednotlivé kraje obsahujú mestá, mestá zasa obsahujú prípadné obchodné reťazce. Hodnotami kľúčov reprezentujúcich názvy reťazcov sú zoznamy adries pobočiek daného reťazca v danom meste. V prípade, že obchodný reťazec nemá v niektorom z miest žiadnu pobočku, slovník, ktorý je hodnotou kľúču nesúceho názov mesta, neobsahuje názov tohto reťazca ako kľúč. V zoznamoch adries pobočiek sa vyskytli i adresy v takých mestách, resp. obciach, ktorých názvy sa v zozname získanom z portálu [31] nenachádzali. Takéto záznamy boli do *JSON* súboru s automaticky zlúčenými zoznamami doplnené manuálne.

Uveďme si tiež *DTD* výsledku extrakcie vo formáte *XML*, ktorý vracia metóda `getAnalysisResult`. Vyzerá nasledovne:

```
<!DOCTYPE post [

<!ELEMENT post (sentence+)>
<!ELEMENT sentence (text,cities?,branches?)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT cities (city+)>
<!ELEMENT city EMPTY>
<!ELEMENT branches (branch+)>
<!ELEMENT branch EMPTY>

<!ATTLIST city name CDATA #REQUIRED>
<!ATTLIST city region CDATA #REQUIRED>
<!ATTLIST city term CDATA #REQUIRED>
<!ATTLIST city from CDATA #REQUIRED>
<!ATTLIST city to CDATA #REQUIRED>
<!ATTLIST branch brand CDATA #REQUIRED>
<!ATTLIST branch street CDATA #REQUIRED>
<!ATTLIST branch city CDATA #REQUIRED>
<!ATTLIST branch region CDATA #REQUIRED>
<!ATTLIST branch term CDATA #REQUIRED>
<!ATTLIST branch from CDATA #REQUIRED>
<!ATTLIST branch to CDATA #REQUIRED>

]>
```

Príspevok (element `post`) v tomto *XML* formáte je teda tvorený zoznamom viet (element `sentence`). Každá veta môže okrem jej textu (element `text`) obsahovať i prípadný zoznam zmienených názvov miest (element `cities`) a/alebo adries pobočiek (element `branches`). O zmienenom meste (element `city`) je evidovaný jeho názov v základnom tvare (atribút `name` elementu `city`), kraj, v ktorom sa toto mesto nachádza (atribút `region`), tvar, v ktorom je názov tohto mesta uvedený vo vete (atribút `term`) a počiatočná a koncová pozícia tohto tvaru v texte vety (atribút `from`, resp. `to`). Pri zmienke adresy pobočky (element `branch`) je uvedená informácia o obchodnom reťazci, ktorému táto pobočka patrí (atribút `brand` elementu `branch`), presná adresa tejto pobočky (atribút `street`), mesto, v ktorom sa pobočka nachádza (atribút `city`), kraj, v ktorom sa toto mesto nachádza (atribút `region`), tvar, v ktorom je adresa pobočky uvedená vo vete (atribút `term`) a počiatočná a koncová pozícia tohto tvaru vo vete (atribút `from`, resp. `to`).

Na záver podkapitoly ešte podotknime, že na ulici s rovnakým názvom (v rovnakom i odlišnom meste) sa môžu nachádzať pobočky viacerých reťazcov. Vo výsledku získanom metódou `getAnalysisResult` sa v takomto prípade nachádzajú úplne všetky pobočky nájdené podľa názvu ulice. Výber tej, ktorej sa zmienka v texte s najväčšou pravdepodobnosťou týka, už má na starosti trieda `XmlProcessor` (viď nasledujúcu podkapitolu).

4.2.9 Trieda `XmlProcessor`

Poslaním triedy `XmlProcessor` je práca s výsledkami analýz a extrakcií vo formátoch *XML*. Metóda `getMergedXmlFormats` zlučuje tieto výsledky do jedného celku. Jej vstupom je výsledok analýzy sentimentu na úrovni aspektu (jej XML formát je popísaný v podkapitole 4.2.5), výsledok extrakcie názvov reťazcov (jej XML formát je popísaný v podkapitole 4.2.6), výsledok extrakcie názvov miest a adries pobočiek (jej XML formát je popísaný v podkapitole 4.2.8), odhadovaná celková polarita príspevku a názov reťazca, ktorého sa príspevok týka.

Práve názov reťazca, ktorého sa príspevok týka, pomáha riešiť situáciu spomenutú v závere predchádzajúcej podkapitoly (názov ulice v adrese pobočky jedného i viacerých obchodných reťazcov môže byť zhodný, a to v rámci jedného i viacerých miest). Predpokladajme teda, že nastal takýto prípad. Na výber pobočky, ktorá je v konkrétnej vete zmienená s najväčšou pravdepodobnosťou, slúži okrem názvu reťazca, ktorého sa príspevok týka, i prípadný zoznam názvov reťazcov explicitne spomenutých v tejto vete (v rámci tejto metódy dostupný z výsledku extrakcie názvov obchodných reťazcov v príspevku) a prípadný zoznam názvov miest, ktoré daná veta spomína. Tieto informácie sú potom použité podľa potreby tak, aby bola vybraná práve jedna zo všetkých nájdených pobočiek, ktorých názov ulice v adrese je zhodný (element reprezentujúci pobočku totiž nesie údaj i o obchodnom reťazci, ktorému táto pobočka patrí rovnako ako o meste, v ktorom sa táto pobočka nachádza).

Výstupom metódy `getMergedXmlFormats` sú dáta vo formáte *XML*. Ich *DTD* je nasledovný:

```
<!DOCTYPE post [

<!ELEMENT post (sentence+)>
<!ELEMENT sentence (text,aspectTerms?,aspectCategories?,brands?,
     cities?,branches?)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT aspectTerms (aspectTerm+)>
<!ELEMENT aspectTerm EMPTY>
<!ELEMENT aspectCategories (aspectCategory+)>
<!ELEMENT aspectCategory EMPTY>
<!ELEMENT brands (brand+)>
<!ELEMENT brand EMPTY>
<!ELEMENT cities (city+)>
<!ELEMENT city EMPTY>
<!ELEMENT branches (branch+)>
<!ELEMENT branch EMPTY>

<!ATTLIST post polarity (positive|neutral|negative) "neutral">
...

]>
```

Z vyššie uvedeného *DTD* možno vyčítať, že výsledný *XML* formát je podobný *XML* formátom výsledkov jednotlivých analýz či extrakcií. Príspevok (element `post`) obsahuje informáciu o svojej polarite (atribút `polarity` elementu `post`) a zoznam viet, ktoré ho tvoria (element `sentence`). Okrem textu vety (element `text`) môže byť evidovaná informácia o prípadných slovách reprezentujúcich aspekty (element `aspectTerm`, potomok elementu `aspectTerms`), o aspektoch, ktorých sa veta týka (element `aspectCategory`, potomok elementu `aspectCategories`), o názvoch obchodných reťazcov, ktoré sú vo vete zmienené (element `brand`, potomok elementu `brands`), o názvoch miest, ktoré veta zmieňuje (element `city`, potomok elementu `cities`) a o adresách pobočiek spomenutých vo vete (element `branch`, potomok elementu `branches`). Z rozsahových dôvodov nie sú v *DTD* uvedené atribúty elementov `aspectTerm`, `aspectCategory`, `brand`, `city` a `branch` (táto skutočnosť je reprezentovaná tromi bodkami na konci *DTD*). Tie sa ale zhodujú s atribútmi príslušných elementov v pôvodných analýzach (ktoré sú vstupom metódy `getMergedXmlFormats`), takže ich popis je možné nájsť v rámci *DTD* uvedeného v podkapitole 4.2.5, v podkapitole 4.2.6, resp. v podkapitole 4.2.8.

Za zmienku tiež stojí metóda `getInformationFromXmlFormat`. Tá extrahuje z dát vo vyššie popísanom formáte *XML* (ktoré sú jej vstupom) informáciu o celkovej polarite príspevku, o aspektoch, ktorých sa príspevok týka, spolu s ich polaritou, o názvoch spomenutých miest, ulíc (adries pobočiek) a obchodných reťazcov. Okrem týchto údajov je k dispozícii i text príspevku

v *HTML* formáte. V rámci neho sú vizuálne zvýraznené prípadne spomenuté slová reprezentujúce aspekty, názvy miest, ulíc (adresy pobočiek) a obchodných reťazcov.

4.3 Balík `metadata`

Balík `metadata` zhromažďuje všetky metadáta potrebné pre beh systému na jednom mieste. Nachádza sa v ňom jediná trieda – trieda `Metadata`. Práve pomocou metód tejto triedy sú metadáta dostupné. Samotné metadáta sú uložené v rámci tohto balíku v súboroch vo formáte *JSON* tak, aby bola manipulácia s nimi (ich pridanie, úprava alebo odstránenie) čo možno najjednoduchšia.

Evidované sú informácie o aspektoch, ktoré sú analyzované (ich zoznam a tiež interný názov každého aspektu v rámci systému spolu s jeho ekvivalentom v rámci vizualizácie výsledkov analýz), o obchodných reťazcoch, ktoré sú systémom monitorované (ich zoznam, pričom o každom reťazci sú okrem jeho názvu uvedené údaje týkajúce sa jeho oficiálnej stránky na sociálnej sieti *Facebook*), o zdrojoch dát (zoznam ich názvov, ktoré budú použité v rámci vizualizácie výsledkov analýz) a o používanom indexe (jeho *URL* adresa a konvencie použité v názvoch indexov a typov).

Pozostatkom z doby návrhu systému je zoznam vekových skupín. Tie mali byť pôvodne použité v grafickom užívateľskom rozhraní na filtrovanie výsledkov analýz podľa veku autorov príspevkov. Informácia o veku autora ale nie je dostupná v žiadnom z použitých zdrojov dát, takže využitie tohto zoznamu vekových skupín momentálne prichádza do úvahy iba v rámci prípadných budúcich rozšírení systému.

4.4 Trieda `IdGenerator`

Trieda `IdGenerator` reprezentuje nástroj pre generovanie unikátnych identifikátorov. Tie sú potrebné pri indexovaní výsledkov analýz príspevkov. Identifikátory týchto príspevkov musia mať vopred stanovený formát (z dôvodu kompatibility s rozhraním pre vizualizáciu výsledkov analýz).

Získanie takéhoto identifikátoru umožňuje metóda `getId`. Tá zo súboru načíta číselnú hodnotu, túto hodnotu upraví do požadovaného formátu, pôvodnú číselnú hodnotu inkrementuje a uloží naspäť do súboru a hodnotu upravenú do požadovaného formátu napokon vráti.

4.5 Trieda `ElasticsearchConnector`

Prostriedok pre prácu s indexom (v rámci tejto práce je použitý fulltextový vyhľadávač *Elasticsearch*, viď [33]) reprezentuje trieda `ElasticsearchConnector`. Jej metóda `getData` slúži na získanie indexovaných dát, metóda `doSaveDataToIndex` slúži na indexovanie dát a metóda `doUpdateData` slúži na aktualizovanie dát v indexe.

Pri práci s fulltextovým vyhľadávačom *Elasticsearch* platia v rámci tohto systému nasledujúce konvencie. Pri indexovaní výsledkov analýz príspevkov je súčasťou názvu indexu (v terminológii fulltextového vyhľadávača *Elasticsearch*) dátum, v ktorom bol príspevok vytvorený. Názov typu dokumentu zasa obsahuje názov obchodného reťazca, ktorého sa príspevok týka. Identifikátor dokumentu (výsledku analýzy konkrétneho príspevku) je generovaný pomocou triedy `IdGenerator`. Vyššie popísané pravidlá sa týkajú iba výsledkov analýz príspevkov, a to z toho dôvodu, aby dotazy na fulltextový vyhľadávač zo strany rozhrania pre vizualizáciu výsledkov analýz mohli byť čo najjednoduchšie.

Názvy indexov a typov, v rámci ktorých sú uložené informácie o užívateľoch, resp. príspevkoch vo formáte *JSON-LD* sú už konštantné hodnoty. Ich identifikátory sú v oboch prípadoch prevzaté zo zdroja, z ktorého boli tieto informácie získané (t.j. zo sociálnej siete *Facebook* alebo z diskusií k článkom na spravodajskom portáli *aktualne.cz*). Konkrétne názvy indexov a typov (v niektorých prípadoch ich prefixy), o ktorých bola v rámci tejto podkapitoly zmienka, poskytuje balík `metadata`.

4.6 Trieda `AnalysisManager`

Prvok riadiaci činnosť systému (sťahovanie dát, ich analýzu a následné indexovanie) reprezentuje trieda `AnalysisManager`. O všetko vyššie spomenuté sa stará jej metóda `doProcessPostsFromDate`, ktorej parametrom je dátum, z ktorého majú byť príspevky stiahnuté, analyzované a indexované. Stiahnuté sú teda postupne príspevky z vybraného dátumu (z násteniek oficiálnych stránok vybraných obchodných reťazcov na sociálnej sieti *Facebook*, zoznam ktorých poskytuje balík `metadata`, a z diskusií všetkých doposiaľ nájdených relevantných článkov zo spravodajského portálu *aktualne.cz*) a prípadné informácie o ich autoroch (ak ešte nie sú tieto údaje v rámci indexu evidované). Príspevky sú následne analyzované a uložené sú nielen výsledky týchto analýz, ale i príspevky vo formáte *JSON-LD*. Informácie o autoroch príspevkov sú buď uložené alebo iba aktualizované (o počet nových príspevkov od tohto autora). Závisí to od toho, či ide o užívateľa, o ktorom sa už v indexe nachádza záznam, alebo nie.

4.7 Zabezpečenie automatickej činnosti systému

Automatickú činnosť systému zabezpečujú skripty popísané v tejto podkapitole. Prvým z nich je skript `posts_from_date_processing.py`. Povinným parametrom pri jeho spustení je konkrétny dátum. Skript vytvára inštanciu triedy `AnalysisManager` a následne volá jej metódu `doProcessPostsFromDate` so spomenutým dátumom ako parametrom. Tento skript teda zabezpečuje spracovanie príspevkov zo zadaného dátumu.

Spracovanie príspevkov z predchádzajúceho dňa má na starosti skript `run_posts_from_date_processing.sh`. Ten zisťuje dátum predchádzajúceho dňa (v porovnaní s dňom, v ktorom je spustený) a následne s týmto dátumom ako parametrom spúšťa skript `posts_from_date_processing.py`. V kombinácii s tohto skriptu s nástrojom na správu úloh (`cron`) môžeme zabezpečiť automatické a pravidelné spracovanie dát.

Skript `run_posts_from_history_processing.sh` slúži na spracovanie príspevkov z dátumov v určitom historickom intervale. Počiatočný a koncový dátum tohto intervalu je potrebné nastaviť priamo v skripte. Skript potom v rámci svojho behu spúšťa skript `posts_from_date_processing.py`, pričom parametrami spúšťaného skriptu sú postupne všetky dátumy z tohto intervalu.

4.8 Vizualizácia výsledkov analýz

Vizualizácia výsledkov analýz je možná pomocou webového rozhrania. Na jeho implementáciu bol použitý framework *Django* (viď [34]).

Z dôvodu prehľadnosti kódu bol pre pohľady vytvorený samostatný balík (balík `views`). Každý pohľad je v tomto balíku reprezentovaný samostatnou triedou, ku ktorej je pridelená konkrétna šablóna. V balíku sa tiež nachádzajú triedy, ktoré nemajú pridelenú šablónu. Takéto triedy implementujú funkcionalitu, ktorú používa viacero pohľadov, pričom triedy reprezentujúce jednotlivé pohľady sú potomkami týchto tried (podľa toho, či danú funkcionalitu využívajú alebo nie). Ide o triedu `ViewUsingElasticsearch` (zabezpečuje interakciu s fulltextovým vyhľadávačom *Elasticsearch*), o triedu `ViewWithFilters` (napĺňa hodnotami filtre, ktoré používa väčšina pohľadov), o triedu `ViewWithSentimentStats` (poskytuje štatistiky o percentuálnom pomere polarity príspevkov) a o triedu `ViewWithSourceStats` (poskytuje štatistiky o početnosti príspevkov z jednotlivých zdrojov dát).

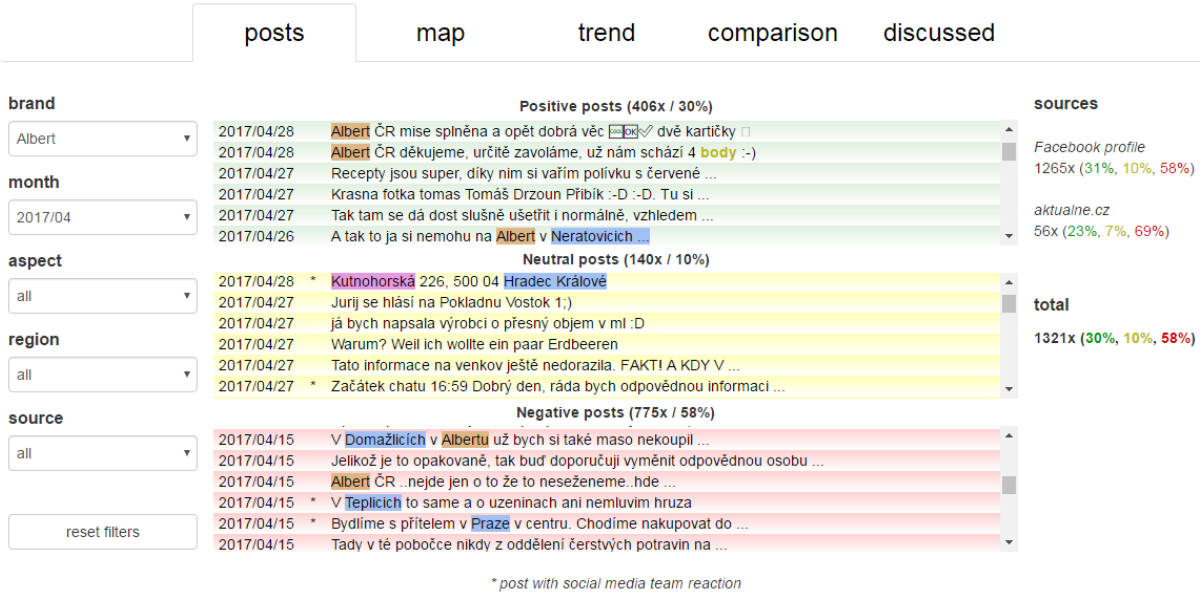
Šablóny reprezentujúce jednotlivé pohľady sú okrem svojho vlastného kódu tvorené prvkami, ktoré sa v rámci stránok môžu opakovať (napríklad filtre alebo štatistiky). Zdrojové kódy takýchto prvkov boli presunuté do samostatných súborov. Tvorbu šablón uľahčilo použitie frameworku *Bootstrap* (viď [35]) a na vykresľovanie interaktívnych grafov v rámci šablón bol použitý framework *Highcharts* (viď [36]).

4.8.1 Zoznam príspevkov

Poďme sa pozrieť na obsah jednotlivých obrazoviek webového rozhrania. Podobu obrazovky, na ktorej sa nachádza zoznam príspevkov ponúka Obrázok 4.1. Tieto príspevky sú rozdelené podľa ich celkovej polarity, pričom ich podfarbenie znázorňuje práve túto polaritu. Pri každej takejto skupine príspevkov je uvedený údaj o ich počte spolu s ich percentuálnym podielom. V rámci textov

jednotlivých príspevkov sú slová reprezentujúce aspekty zvýraznené tučným písmom, pričom ich farba symbolizuje ich polaritu v danej vete (zelená symbolizuje kladnú polaritu, žltá neutrálnu a červená zápornú). Od zvyšku textu sa zasa farbou svojho pozadia odlišujú názvy obchodných reťazcov (farba ich pozadia je hnedá), názvy miest (farba ich pozadia je modrá) a adresy pobočiek (farba ich pozadia je fialová). Okrem textov jednotlivých príspevkov sú uvedené i dátumy ich vytvorenia (na základe ktorých sú usporiadané) a prípadný indikátor značiaci, že na príspevok odpovedalo komunikačné oddelenie obchodného reťazca, ktorého sa príspevok týka (týmto indikátorom je znak „*“ nachádzajúci sa medzi dátumom vytvorenia príspevku a jeho textom). Po nabehnutí kurzora na konkrétny príspevok je možné zobraziť jeho celý text.

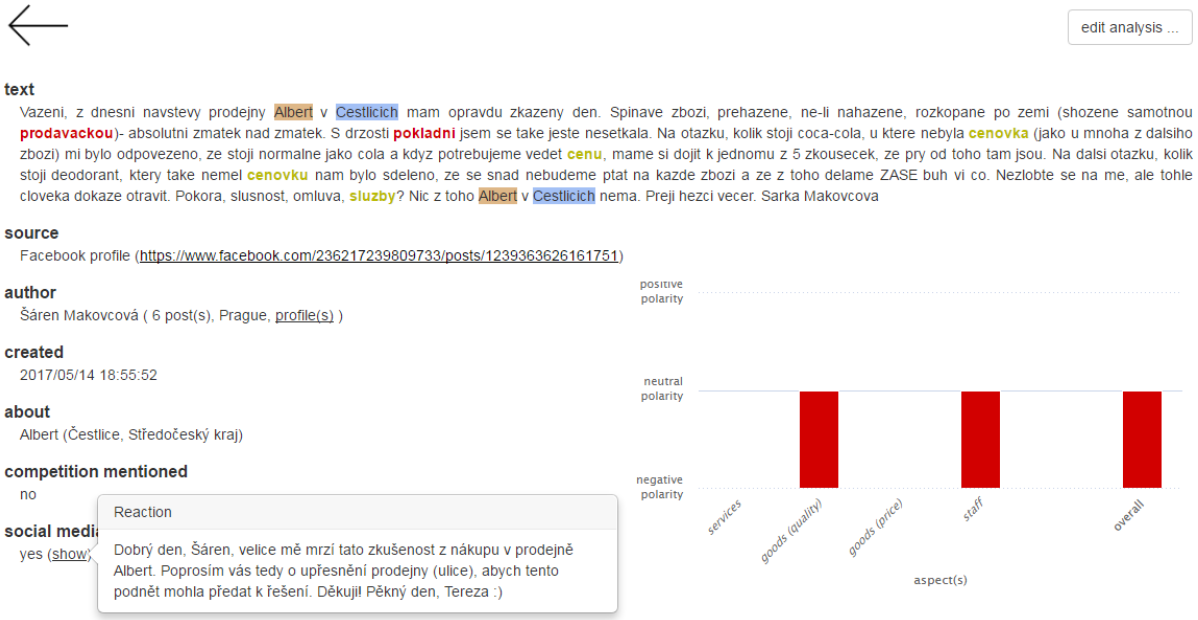
Na ľavej strane obrazovky vidíme filtre. Pomocou nich je možné zobraziť príspevky podľa reťazca, ktorého sa týkajú, podľa mesiaca, z ktorého pochádzajú, podľa aspektu, ktorý spomínajú, podľa kraja, z ktorého mesto alebo adresa pobočky, ktorá sa v tomto kraji nachádza, sú v príspevku zmienené a taktiež podľa zdroja, z ktorého príspevky pochádzajú. Na pravej strane obrazovky sa zasa nachádzajú štatistiky. V tých je uvedené, koľko zobrazených príspevkov pochádza z akého zdroja spolu s percentuálnym podielom celkovej polarity týchto príspevkov pre každý zdroj. Nižšie je potom uvedený celkový počet zobrazených príspevkov spolu s percentuálnym podielom ich celkových polarít. Vo vrchnej časti obrazovky je navigácia, ktorá umožňuje prechod na iné obrazovky.



Obrázok 4.1: Obrazovka obsahujúca zoznam príspevkov.

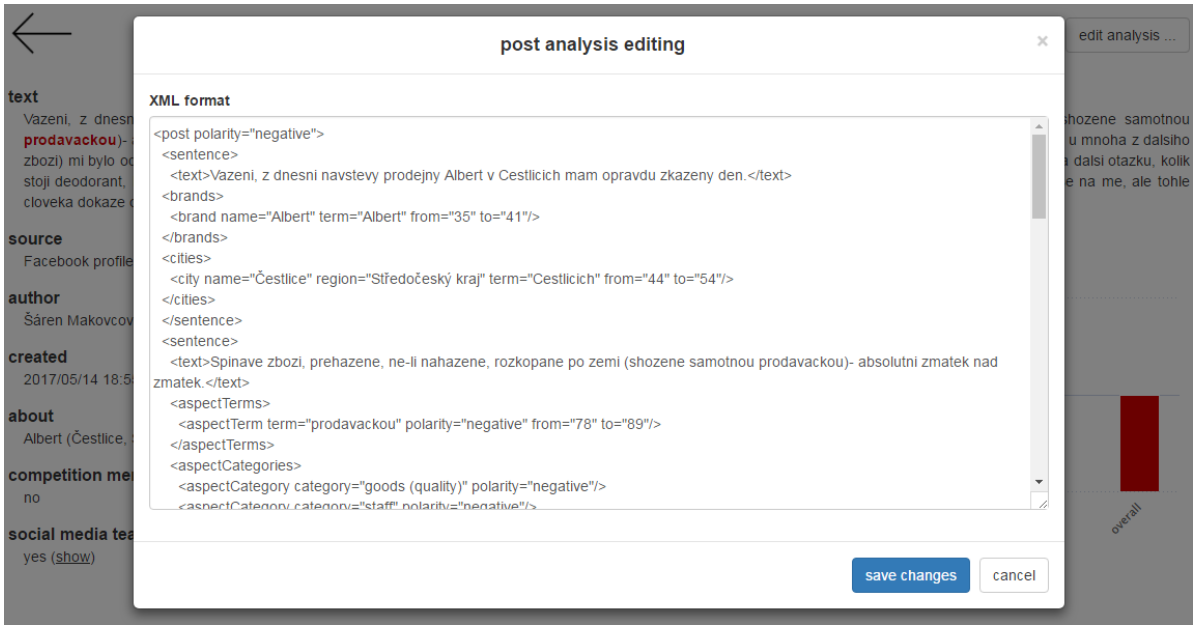
4.8.2 Detail príspevku

Po kliknutí na konkrétny príspevok v zozname príspevkov popísanom v predchádzajúcej podkapitole sa dostaneme do detailu príspevku. Obrázok 4.2 zobrazuje obrazovku detailu jedného z príspevkov. V detaile sa nachádza text príspevku, v ktorom sú znovu farebne odlíšené (rovnakým spôsobom ako v zozname príspevkov) nájdené slová reprezentujúce aspekty, názvy obchodných reťazcov, názvy miest a adresy pobočiek. Okrem textu príspevku sú k dispozícii informácie o jeho zdroji (jeho názov spolu s *URL* adresou príspevku), o jeho autorovi (meno, počet príspevkov v rámci tohto systému, zoznam jeho profilov naprieč webom, prípadné miesto bydliska a prípadný údaj o pohlaví), o čase jeho vytvorenia, o reťazci, ktorého sa týka (s prípadným údajom o zmienenej pobočke a/alebo zmienenom meste spolu s krajom, v ktorom sa táto pobočka, resp. mesto nachádza), o prípadne spomenutých názvoch konkurenčných reťazcov či o prípadnom obsahu spätnej väzby na tento príspevok od komunikačného oddelenia reťazca, ktorého sa príspevok týka. Súčasťou detailu príspevku je tiež modifikovaná verzia stĺpcového grafu vizuálne znázorňujúca polaritu jednotlivých aspektov spomenutých v príspevku. V grafe je tiež uvedená celková polarita príspevku.



Obrázok 4.2: Obrazovka obsahujúca detail jedného z príspevkov.

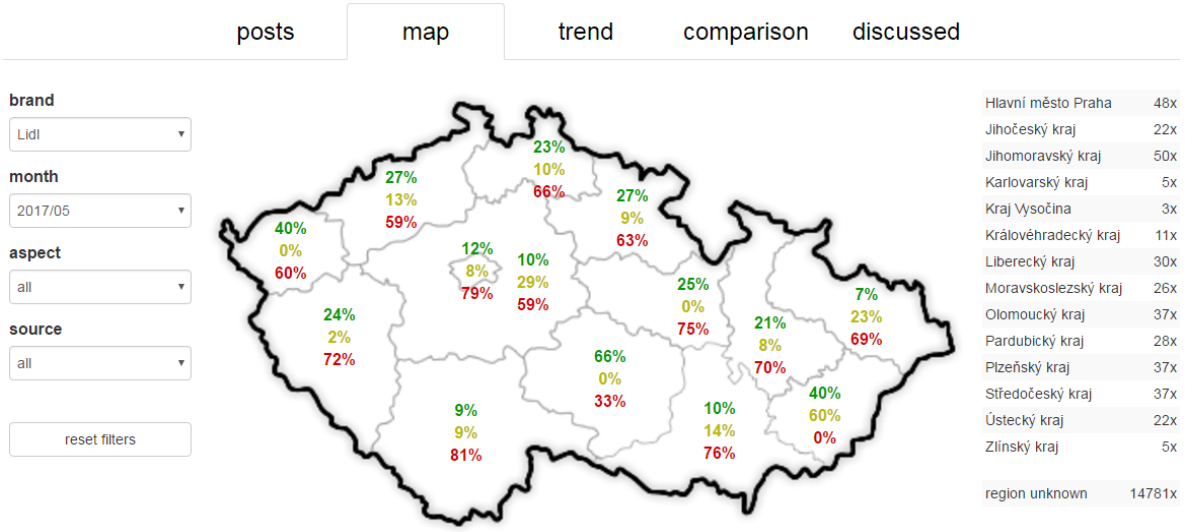
Vo vrchnej časti obrazovky sa v podobe šípky nachádza tlačidlo pre návrat do okna so zoznamom príspevkov. Na pravej strane vrchnej časti obrazovky sa zasa nachádza tlačidlo pre manuálnu editáciu výsledku analýzy príspevku. Po jeho stlačení sa zobrazí modálne okno, ktorého podobu demonštruje Obrázok 4.3. V tomto modálnom okne sa nachádza textové pole obsahujúce zlúčený výsledok jednotlivých analýz v *XML* formáte popísanom v podkapitole 4.2.9. Po jeho prípadnej úprave a potvrdení tejto úpravy sa v nadväznosti na vykonané zmeny aktualizujú informácie v detaile príspevku.



Obrázok 4.3: Modálne okno umožňujúce manuálnu úpravu výsledku analýzy príspevku.

4.8.3 Štatistiky jednotlivých krajov Českej republiky

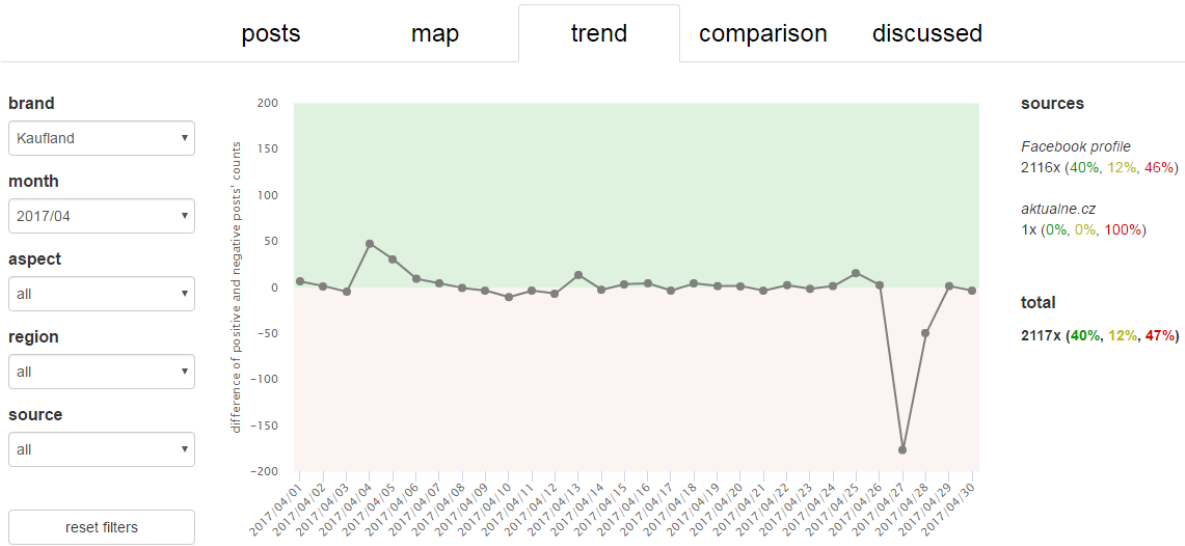
Jednou z obrazoviek implementovaného webového rozhrania je obrazovka obsahujúca regionálne rozdelenie štatistík. Podobu tejto obrazovky zobrazuje Obrázok 4.4. Ako môžeme vidieť, na obrazovke sa nachádza mapa Českej republiky s vyznačenými hranicami jednotlivých krajov. V oblasti každého kraja je uvedené percentuálne rozloženie príspevkov, v ktorých sa spomína mesto alebo pobočka z tohto kraja, a to podľa ich celkovej polarity (zelené číslo značí percentuálny podiel kladných príspevkov, žlté číslo symbolizuje percentuálny podiel neutrálnych príspevkov a červené číslo zasa percentuálny podiel tých negatívnych). Napravo od mapy sa nachádza tabuľka uvádzajúca počet príspevkov zmieňujúcich mestá alebo pobočky v jednotlivých krajoch. Pre porovnanie je v nej taktiež uvedený údaj o počte príspevkov, v ktorých nebolo spomenuté žiadne mesto alebo adresa pobočky. Na obrazovke sa ešte v jej ľavej časti nachádzajú filtre (v tomto prípade je možné filtrovať štatistiky podľa reťazcov, mesiacov, spomenutých aspektov a zdrojov dát) a v jej hornej časti sa zasa nachádza navigácia.



Obrázok 4.4: Obrazovka obsahujúca štatistiky jednotlivých krajov Českej republiky.

4.8.4 Trend celkovej polarity príspevkov

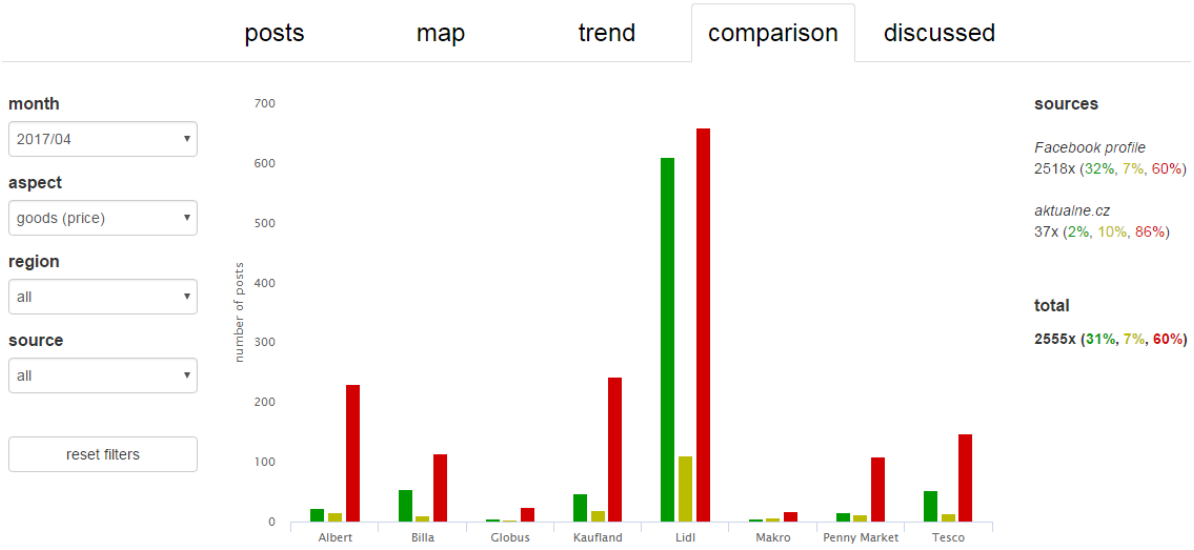
Prvou z obrazoviek, ktoré ponúkajú výsledky analýz v podobe grafu, je obrazovka obsahujúca graf trendu. Obrázok 4.5 ponúka ukážku takejto obrazovky. V spomínanom trendovom grafe je pre každý deň zvoleného mesiaca vypočítaný rozdiel medzi počtom príspevkov s kladnou a zápornou celkovou polaritou (tento rozdiel je pre vybraný deň možné zistiť nabehnutím kurzora na príslušný bod v grafe). Trendová čiara potom spája takéto hodnoty. Okrem grafu sa na obrazovke nachádzajú filtre (na ľavej strane obrazovky, pričom filtrovať výsledky je možné podľa reťazcov, mesiacov, spomenutých aspektov, krajov, z ktorých mestá alebo pobočky boli v príspevkoch spomenuté alebo podľa zdrojov príspevkov), štatistiky o zdrojoch príspevkov a polarite príspevkov (na pravej strane) a navigácia (vo vrchnej časti).



Obrázok 4.5: Obrazovka obsahujúca trend celkovej polarity príspevkov.

4.8.5 Porovnanie reťazcov

Obrázok 4.6 obsahuje ukážku obrazovky umožňujúcej porovnanie reťazcov. K samotnému porovnaniu je použitý stĺpcový graf. Hodnotami jeho vodorovnej osi sú názvy obchodných reťazcov. Pre každý z reťazcov sú v grafe zobrazené tri stĺpce. Tie udávajú počet príspevkov týkajúcich sa reťazca, ktoré majú kladnú (zelený stĺpec), neutrálnu (žltý stĺpec), resp. zápornú (červený stĺpec) celkovú polaritu. Nabehnutím kurzora na vybranú trojicu stĺpcov je možné zistiť presné hodnoty počtov, ktoré jednotlivé stĺpce reprezentujú. V ľavej časti obrazovky sa opäť nachádzajú filtre, pomocou ktorých je možné filtrovať zobrazované výsledky (v tomto prípade nie je možné filtrovať podľa reťazcov, keďže tie sa nachádzajú priamo v grafe). Pravá časť obrazovky obsahuje štatistiky o zdrojoch a polarite príspevkov a v hornej časti obrazovky sa nachádza navigácia.

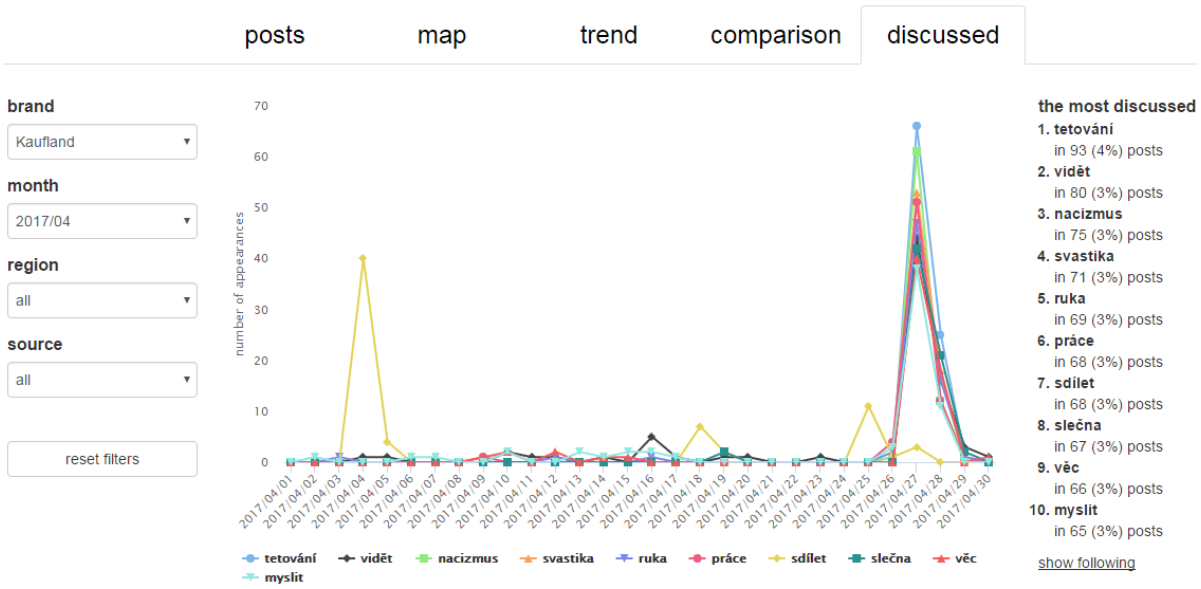


Obrázok 4.6: Obrazovka umožňujúca porovnanie reťazcov.

4.8.6 Najfrekventovanejšie kľúčové slová

Posledná z implementovaných obrazoviek ponúka časové rozloženie výskytu kľúčových slov, ktoré sa v analyzovaných príspevkoch objavujú najčastejšie. Vzhľad tejto obrazovky ponúka Obrázok 4.7. Na zobrazenie spomenutého časového rozloženia výskytov kľúčových slov príspevkov je použitý graf trendu. V ňom sa nachádza trendová čiara pre každé z prvých desiatich najfrekventovanejších kľúčových slov z analyzovaných príspevkov. Trendová čiara konkrétneho slova spája počty výskytov tohto slova v jednotlivých dňoch. Je tak možné zistiť, kedy sa o čom diskutovalo najviac. Presný počet výskytov kľúčového slova v konkrétnom dni je možné zistiť nabehnutím kurzora na príslušný bod v grafe. Kliknutím na kľúčové slovo v legende grafu je zasa možné jeho trendovú čiaru z grafu odstrániť či prípadne znovu zobraziť. Napravo od trendového grafu sa tiež nachádza tabuľka s kľúčovými slovami zoradenými podľa počtu výskytov (od najfrekventovanejšieho) v príspevkoch z konkrétneho mesiaca. Pri konkrétnom slove je uvedený i samotný údaj o jeho počte výskytov

a percentuálny podiel príspevkov, v ktorých sa toto slovo nachádza. Zobraziť je tiež možné niekoľko kľúčových slov, ktoré sa do tabuľky, resp. grafu nevošli (tie, ktoré sa v rebríčku nachádzajú na jedenástej a vyššej pozícii). Toto je možné po kliknutí na odkaz pod tabuľkou. Výsledky zobrazené v tabuľke i grafe je možné filtrovať podľa reťazcov, mesiacov, krajov a zdrojov dát (pomocou filtrov v ľavej časti obrazovky). V hornej časti obrazovky je k dispozícii navigácia.



Obrázok 4.7: Obrazovka obsahujúca rebríček najfrekventovanejších kľúčových slov spolu so zobrazením ich výskytu v čase.

5 Vyhodnotenie systému

Po návrhu systému a jeho následnej implementácii boli vyhodnotené jeho vybrané časti. Práve výsledky týchto vyhodnotení sú obsahom tejto kapitoly. V prípade triedy implementujúcej klasifikáciu (ktorá je použitá v rámci analýzy sentimentu na úrovni dokumentu i na úrovni aspektu) a triedy implementujúcej extrakciu lokalít ide o ich experimentálne vyhodnotenie. V prípade implementovaného webového rozhrania zasa ide o demonštráciu odhalenia konkrétnej kauzy, a to pomocou grafov vykresľovaných v tomto rozhraní.

5.1 Porovnanie modelov klasifikátorov

V tejto podkapitole sa zameriame na vyhodnotenie triedy `Classifier` (popísanej v podkapitole 4.2.2), ktorá reprezentuje klasifikátor. V troch testovacích scenárov si vyhodnotíme správnosť (*accuracy*), s ktorou je model klasifikátoru, vytvorený touto triedou, schopný v rámci daného scenáru klasifikovať dáta.

Pri samotnom testovaní použijeme metódu nazývanú *repeated holdout*. Vo všetkých prípadoch bude priebeh testovania podobný. Modely sa budú vytvárať a testovať postupne pre všetky kombinácie vybraných možností predspracovania tréningovej sady a klasifikačných metód. Vo všetkých prípadoch sa v rámci predspracovania tréningovej sady rozdelí text na slová, zo slov sa odstráni diakritika, odstránia sa stop slová a slová, ktoré zostali, budú transformované tak, aby boli tvorené iba malými písmenami. To, či budú slová transformované do svojho základného alebo kmeňového tvaru, a aká klasifikačná metóda sa pri testovaní použije (trieda `Classifier` implementuje tri – naivnú Bayesovskú metódu, metódu *Maximum Entropy* a metódu *SVM*), už bude záležať od parametrov konkrétnej iterácie. Parametrami je myslená vyššie spomenutá kombinácia vybraných možností predspracovania tréningovej sady a klasifikačných metód. Pre konkrétnu kombináciu bude desaťkrát vytvorený a testovaný model. Tréningovú sadu bude tvoriť 80 % príspevkov dodanej sady anotovaných dát, tú testovaciu zasa zvyšných 20 % príspevkov tejto sady. Pred každým vytvorením modelu bude anotovaná sada dát náhodne zamiešaná tak, aby bol pomer tried jednotlivých položiek v neskôr oddelenej časti určenej na testovanie klasifikátoru rovný pomeru tried jednotlivých položiek v rámci celej tejto anotovanej sady. Celkový počet správne klasifikovaných položiek spolu s celkovým počtom klasifikovaných položiek v rámci danej kombinácie predspracovania dát a použitej klasifikačnej metódy potom poslúži na určenie celkovej správnosti klasifikácie za použitia tejto kombinácie.

5.1.1 Určenie celkovej polarity príspevkov

Prvým testovacím scenárom je testovanie klasifikácie celkovej polarity príspevkov. Takýto typ klasifikácie sa používa v rámci analýzy sentimentu na úrovni dokumentu (dokument je v našom prípade príspevok), čo znamená, že implementovaná je v triede `PolarityAnalyser`. Použitou sadou dát, ktorá bude v rámci testovania rozdelená na tréningovú sadu a testovaciu sadu, je v tomto prípade 1200 anotovaných príspevkov z násteniek oficiálnych stránok českých potravinových obchodných reťazcov na sociálnej sieti *Facebook*. Trieda `PolarityAnalyser` spolu s touto sadou anotovaných dát je bližšie popísaná v podkapitole 4.2.4. Výsledky tohto testovania ponúka Tabuľka 5.1.

Predspracovanie textu		**Klasifikačná metóda**			**Správnosť**
Základný tvar slov	Kmeňový tvar slov	Naivná Bayesovská	Maximum Entropy	SVM	
✓		✓			*0.64*
✓			✓		*0.72*
✓				✓	*0.71*
	✓	✓			*0.65*
	✓		✓		*0.74*
	✓			✓	*0.73*

Tabuľka 5.1: Kombinácie možností predspracovania textu s použitými metódami klasifikácie a nimi dosiahnuté správnosti pri klasifikácii celkovej polarity príspevkov.

Z tejto tabuľky vidíme, že najvyššiu hodnotu správnosti (0.74) dosahuje klasifikačná metóda *Maximum Entropy*, a to pri transformovaní slov na ich kmeňový tvar v rámci predspracovania tréningovej sady. Taktiež môžeme vidieť, že pri použití naivnej Bayesovskej metódy pri klasifikácii (bez ohľadu na voľby predspracovania tréningovej sady) je v tomto konkrétnom prípade dosiahnutá správnosť najhoršia.

5.1.2 Detekcia viet týkajúcich sa jednotlivých aspektov

Súčasťou analýzy sentimentu na úrovni aspektu je i detekcia viet týkajúcich sa konkrétnych aspektov (ak sa v týchto vetách explicitne nenachádza slovo alebo viacero slov, ktoré daný aspekt reprezentujú). Takáto detekcia je teda implementovaná v triede `AspectBasedSentimentAnalyser` a v tejto podkapitole bude i otestovaná. Zdrojom

tréningových a testovacích dát budú v tomto prípade vety rozdelené do tried podľa toho, či sa daného aspektu týkajú alebo nie. Táto anotovaná sada je spolu s triedou `AspectBasedSentimentAnalyser` popísaná v podkapitole 4.2.5. V rámci každej kombinácie predspracovania tréningovej sady s použitou metódou klasifikácie je desať iterácií spomínaných v popise priebehu testovania vykonaných pre každý aspekt (a tiež triedu irelevantných viet) zvlášť. Je to spôsobené tým, že pre každý aspekt (a triedu irelevantných viet) je vytvorená samostatná tréningová sada, resp. súbor túto sadu obsahujúci. Tabuľka 5.2 obsahuje výsledky testovania.

Predspracovanie textu		**Klasifikačná metóda**			**Správnosť**
Základný tvar slov	Kmeňový tvar slov	Naivná Bayesovská	Maximum Entropy	SVM	
✓		✓			*0.78*
✓			✓		*0.73*
✓				✓	*0.78*
	✓	✓			*0.77*
	✓		✓		*0.73*
	✓			✓	*0.78*

Tabuľka 5.2: Kombinácie možností predspracovania textu s použitými metódami klasifikácie a nimi dosiahnuté správnosti pri klasifikácii príslušností viet ku konkrétnym aspektom.

Vidíme, že v tomto prípade dosahuje správnosť klasifikácie vyššie hodnoty. Najvyššia hodnota správnosti (0.78) je pritom dosiahnutá v troch prípadoch. Deje sa tak pri použití naivného Bayesovského klasifikátoru s prevodom slov na ich základný tvar pri predspracovaní tréningovej sady a pri použití klasifikačnej metódy *SVM* bez ohľadu na voľbu predspracovania tréningovej sady. Najhoršie výsledky dosahuje pri tomto type klasifikácie metóda *Maximum Entropy*.

5.1.3 Určenie polarity viet týkajúcich sa jednotlivých aspektov

V rámci analýzy sentimentu na úrovni aspektu je taktiež využitá klasifikácia polarity viet týkajúcich sa konkrétneho aspektu. Testovanie tohto typu klasifikácie spolu s jeho výsledkami popisuje táto podkapitola. V tomto prípade sú zdrojom tréningových a testovacích dát vety týkajúce sa konkrétnych aspektov, ktoré sú rozdelené do tried podľa ich polarity. Tie sú popísané opäť v podkapitole 4.2.5, pričom v samotnom priebehu testovania sú iterácie vykonávané pre každý aspekt zvlášť, rovnako, ako je tomu i pri testovaní popísanom v predchádzajúcej podkapitole. Výsledky testov uvádza Tabuľka 5.3.

Predspracovanie textu		**Klasifikačná metóda**			**Správnosť**
Základný tvar slov	Kmeňový tvar slov	Naivná Bayesovská	Maximum Entropy	SVM	
✓		✓			*0.63*
✓			✓		*0.65*
✓				✓	*0.64*
	✓	✓			*0.63*
	✓		✓		*0.65*
	✓			✓	*0.64*

Tabuľka 5.3: Kombinácie možností predspracovania textu s použitými metódami klasifikácie a nimi dosiahnuté správnosti pri klasifikácii polarít viet týkajúcich sa jednotlivých aspektov.

V porovnaní s dvomi predchádzajúcimi testovaniami sú hodnoty dosiahnutých správností v tomto type klasifikácie najhoršie. Hodnoty dosiahnutých správností klasifikácií sú pre konkrétnu klasifikačnú metódu rovnaké bez ohľadu na použitú voľbu predspracovania tréningových dát. Najvyššiu hodnotu správnosti klasifikácie (0.65) dosahuje v tomto prípade klasifikačná metóda *Maximum Entropy*.

5.2 Správnosť nástroja pre extrakciu lokalít

Experimentálnemu vyhodnoteniu sa podrobil i nástroj extrahujúci názvy miest a adresy pobočiek českých potravinových obchodných reťazcov z textu. Ide o triedu `LocationsExtractor` popísanú v podkapitole 4.2.8. Referenčnými dátami bolo v tomto prípade 152 zo 400 ručne analyzovaných príspevkov z nástenieck oficiálnych stránok českých potravinových obchodných reťazcov na sociálnej sieti *Facebook*. Práve v týchto 152 príspevkoch bola explicitne zmienená lokalita.

V rámci testu boli z každého príspevku pomocou implementovaného nástroja extrahované lokality. Tie boli potom porovnané s tými referenčnými a za úspech sa považovala ich zhoda. Zo 152 testovaných príspevkov bola správne extrahovaná lokalita v 103 príspevkoch. Hodnota správnosti je v tomto prípade 0.68.

Treba tiež spomenúť faktory zlyhania extrakcie lokalít v tomto teste. Boli nimi chybné adresy pobočiek uvedené v príspevkoch, názvy miest a ulíc začínajúce malými písmenami, chýbajúca diakritika v názvoch miest a ulíc a zmienky o mestách pomocou prídavných mien (napríklad

olomoucký), ktoré nástroj extrahovať nedokáže. Tieto faktory sa vyskytli v 39 príspevkoch. Pri zostávajúcich 10 príspevkoch išlo o zlyhania neovplyvnené vonkajšími faktormi. To znamená, že všetky predpoklady pre správnu extrakciu lokalít z príspevku boli splnené (napríklad názov mesta s veľkým písmenom i s prípadnou diakritikou), no nástroj tieto lokality nevyextrahoval.

5.3 Príklad detekcie aktuálnych káuz

Jednou z možností použitia implementovaného systému je detekcia káuz týkajúcich sa českých potravinových obchodných reťazcov. Tá je možná pomocou webového rozhrania, resp. ním zobrazených grafov. Poďme sa pozrieť na príklad odhalenia konkrétnej kauzy.

Už pri prvom pohľade na trend celkovej polarity príspevkov z apríla 2017 týkajúcich sa obchodného reťazca *Kaufland* (viď Obrázok 4.5) vidíme, že spomedzi všetkých príspevkov vytvorených v dňoch 27.4.2017 a 28.4.2017 majú výraznú prevahu tie so zápornou polaritou. Po zobrazení časového rozloženia výskytov najviac spomínaných kľúčových slov z apríla 2017, ktoré sa nachádzajú v príspevkoch týkajúcich sa obchodného reťazca *Kaufland* (viď Obrázok 4.7) zasa vidíme, že v rovnakých dňoch boli s vysokou frekvenciou (v porovnaní s ostatnými dňami) spomínané isté slová. Ide o slová ako *tetování, nacizmus, svastika, ruka* alebo *slečna*. Tieto dva grafy nám teda poodhaľujú kauzu zamestnankyne obchodného reťazca *Kaufland*, ktorá má na ruke tetovanie obsahujúce hákový kríž. O tejto kauze sa zmieňuje napríklad článok [37] (vydaný dňa 26.4.2017), článok [38] (vydaný dňa 27.4.2017) alebo článok [39] (vydaný dňa 27.4.2017).

6 Záver

Cieľom tejto práce bolo navrhnúť a implementovať systém, ktorý dokáže pravidelne sťahovať dáta z webu, analyzovať stiahnuté dáta a indexovať ich. Jeho súčasťou tiež mal byť prostriedok pre vizualizáciu výsledkov analýz. Stanovený cieľ sa podarilo naplniť.

V dobe odovzdania tejto diplomovej práce je implementovaný systém nasadený na dvoch serveroch. Na serveri `athena2.fit.vutbr.cz` zabezpečuje nástroj `cron` spúšťanie skriptu `run_articles_history_update.sh` (popísaného v podkapitole 4.1.10), a to každý deň v čase 1:00. O 1:10 je potom na serveri `athena1.fit.vutbr.cz`, taktiež každý deň a taktiež pomocou nástroja `cron`, spúšťaný skript `run_posts_from_date_processing.sh` (popísaný v podkapitole 4.7). Systém používa fulltextový vyhľadávač *Elasticsearch* dostupný na *URL* adrese `http://athena1.fit.vutbr.cz:9200`. Webové rozhranie systému beží na *URL* adrese `http://athena1.fit.vutbr.cz:2424`.

Obsahom tejto technickej správy je okrem teoretického rozboru problematiky so systémom úzko súvisiacej i popis jeho návrhu a implementácie. Nasleduje vyhodnotenie vybraných implementovaných častí systému. V rámci tohto vyhodnotenia je na konkrétnom príklade demonštrované možné použitie systému na odhaľovanie káuz českých potravinových obchodných reťazcov.

Námetov na prípadný ďalší vývoj systému existuje hneď niekoľko. Pokryté by mohli byť ďalšie zdroje relevantných dát. Tiež by mohli byť zlepšené výsledky analýzy sentimentu na úrovni dokumentu i aspektu, a to prostredníctvom rozšírenia tréningových sád, pridaním nových príznakov klasifikácie alebo prostredníctvom pretrénovania používaných modelov klasifikátorov na základe manuálnej úpravy výsledku analýzy vo webovom rozhraní. Nástroj pre extrakciu lokalít z textu by mohol byť vylepšený tak, aby bol schopný extrahovať i názvy mestských častí (napríklad *Bohunice* alebo *Královo Pole*) alebo prídavné mená týkajúce sa miest (napríklad *olomoucký*). Užívatelia evidovaní systémom by mohli byť analyzovaní na základe obsahu nimi vytvorených príspevkov. Z dôvodu občasnej dlhšej odozvy webového rozhrania by do úvahy prichádzala i optimalizácia formátu indexovaných dát spolu s dotazmi na index zo strany webového rozhrania. Samotný systém by tiež mohol byť použitý pre iný jazyk (napríklad pre slovenčinu) a/alebo inú doménu (napríklad pre doménu mobilných operátorov).

Literatúra

[1] DE S SIRISURIYA, S.C.M. A Comparative Study on Web Scraping. *Proceedings of 8th International Research Conference* [online]. 2015 [cit. 2017-05-13]. Dostupné z: http://www.kdu.ac.lk/proceedings/irc2015/2015/com-020.pdf

[2] *Welcome to the microformats wiki!* [online]. 2017 [cit. 2017-05-13]. Dostupné z: http://microformats.org/

[3] LIU, Bing. *Sentiment Analysis and Opinion Mining* [online]. Morgan & Claypool Publishers, 2012 [cit. 2016-12-25]. ISBN 978-1608458844. Dostupné z: https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

[4] LIU, Bing a Lei ZHANG. A Survey of Opinion Mining and Sentiment Analysis. *Mining Text Data* [online]. Boston, MA: Springer US, 2012, 415 [cit. 2016-12-25]. DOI: 10.1007/978-1-4614-3223-4_13. ISBN 978-1-4614-3222-7. Dostupné z: http://link.springer.com/10.1007/978-1-4614-3223-4_13

[5] LIU, Bing. Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing* [online]. 2010 [cit. 2016-12-26]. Dostupné z: http://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment%20Analysis%20and%20Subjectivity-NLPHandbook-2010.pdf

[6] PAVLOPOULOS, Ioannis. *Aspect Based Sentiment Analysis* [online]. 2014 [cit. 2016-12-26]. Dostupné z: http://nlp.cs.aueb.gr/theses/ipavlopoulos-thesis.pdf. Ph.D.Thesis. Department of Informatics, Athens University of Economics and Business.

[7] WANG, Bo a Min LIU. *Deep Learning for Aspect-Based Sentiment Analysis* [online]. [cit. 2016-12-27]. Dostupné z: https://cs224d.stanford.edu/reports/WangBo.pdf

[8] SemEval. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-12-27]. Dostupné z: https://en.wikipedia.org/wiki/SemEval

[9] PATERIA, Shubham a Prafulla Kumar CHOUBEY. AKTSKI at SemEval-2016 Task 5: Aspect Based Sentiment Analysis for Consumer Reviews. *Proceedings of SemEval-2016* [online]. 2016 [cit. 2016-12-28]. Dostupné z: https://www.aclweb.org/anthology/S/S16/S16-1051.pdf

[10] ALGHUNAIM, Abdulaziz, Mitra MOHTARAMI, Scott CYPHERS a James GLASS. A Vector Space Approach for Aspect Based Sentiment Analysis. *Proceedings of NAACL-HLT 2015* [online]. 2015 [cit. 2016-12-28]. Dostupné z: http://www.aclweb.org/anthology/W15-1516

[11] BENYU, Zhang. Text Indexing and Retrieval. *Encyclopedia of Database Systems* [online]. 2009, 3055-3058 [cit. 2016-12-20]. ISBN 978-0-387-39940-9. Dostupné z: http://link.springer.com/book/10.1007/978-0-387-39940-9

[12] DE MOURA, Edleno Silva. Text Indexing Techniques. *Encyclopedia of Database Systems* [online]. 2009, 3058-3061 [cit. 2016-12-20]. ISBN 978-0-387-39940-9. Dostupné z: http://link.springer.com/book/10.1007/978-0-387-39940-9

[13] BEITZEL, Steven, Eric C. JENSEN a Ophir FRIEDER. Index Creation and File Structures. *Encyclopedia of Database Systems* [online]. 2009, 1425-1427 [cit. 2016-12-21]. ISBN 978-0-387-39940-9. Dostupné z: http://link.springer.com/book/10.1007/978-0-387-39940-9

[14] MIRNA, Adriani a W. Bruce CROFT. *Retrieval Effectiveness Of Various Indexing Techniques On Indonesian News Articles* [online]. 1997 [cit. 2016-12-21]. Dostupné z: http://ciir-publications.cs.umass.edu/getpdf.php?id=170

[15] SHAH, Najmus Saher. Review of Indexing Techniques Applied in Information Retrieval. *Pak. j. eng. technol. sci.* [online]. 2015, , 27-47 [cit. 2016-12-22]. ISSN 2224-2333. Dostupné z: http://journals.iobmresearch.com/index.php/PJETS/article/view/324/82

[16] RAMAKRISHNA, Kolikipogu a Dr.B.Padmaja RANI. Study of Indexing Techniques to Improve the Performance of Information Retrieval in Telugu Language. *International Journal of Emerging Technology and Advanced Engineering* [online]. 2013 [cit. 2016-12-22]. ISSN 2250-2459. Dostupné z: http://www.ijetae.com/files/Volume3Issue1/IJETAE_0113_76.pdf

[17] MANNING, Christopher D., Prabhakar RAGHAVAN a Hinrich SCHÜTZE. *Introduction to information retrieval* [online]. Cambridge: Cambridge University Press, 2008 [cit. 2016-12-23]. ISBN 978-0-521-86571-5. Dostupné z: http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

[18] SAGAYAM, R., S. SRINIVASAN a S. ROSHNI. A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques. *International Journal Of Computational Engineering Research* [online]. 2012 [cit. 2016-12-23]. ISSN 2250-3005. Dostupné z: https://pdfs.semanticscholar.org/e6b6/fa3ad3215ad7b9a61c0e6e1f706b96f2f5df.pdf

[19] RMIT, Justin Zobel, Alistair MOFFAT a Kotagiri RAMAMOHANARAO. Inverted Files Versus Signature Files for Text Indexing. *ACM Transactions on Database Systems* [online]. 1998 [cit. 2016-12-24]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.576&rep=rep1&type=pdf

[20] *Graph API - Documentation - Facebook for Developers* [online]. [cit. 2017-05-15]. Dostupné z: https://developers.facebook.com/docs/graph-api

[21] *Selenium Documentation* [online]. 2011 [cit. 2017-05-15]. Dostupné z: https://seleniumhq.github.io/selenium/docs/api/py/api.html

[22] *JSON-LD - JSON for Linking Data* [online]. 2014 [cit. 2017-05-16]. Dostupné z: http://json-ld.org/

[23] *Welcome to polyglot's documentation!* [online]. c2014-2015 [cit. 2017-05-17]. Dostupné z: http://polyglot.readthedocs.io/en/latest/

[24] *Unidecode 0.04.20* [online]. c1990-2017 [cit. 2017-05-17]. Dostupné z: https://pypi.python.org/pypi/Unidecode

[25] *MorphoDiTa* [online]. c2017 [cit. 2017-05-17]. Dostupné z: https://ufal.mff.cuni.cz/morphodita

[26] *Sumy 0.6.0* [online]. c1990-2017 [cit. 2017-05-17]. Dostupné z: https://pypi.python.org/pypi/sumy

[27] *TextBlob: Simplified Text Processing* [online]. c2017 [cit. 2017-05-17]. Dostupné z: https://textblob.readthedocs.io/en/dev/

[28] *Natural Language Toolkit - NLTK 3.2.4 documentation* [online]. c2017 [cit. 2017-05-17]. Dostupné z: http://www.nltk.org/

[29] *Scikit-learn: machine learning in Python - scikit-learn 0.18.1 documentation* [online]. [cit. 2017-05-17]. Dostupné z: http://scikit-learn.org/stable/

[30] *Task Description: Aspect Based Sentiment Analysis (ABSA) < SemEval-2014 Task 4* [online]. c2017 [cit. 2017-05-18]. Dostupné z: http://alt.qcri.org/semeval2014/task4/

[31] *Města a obce online - portál územní samosprávy* [online]. c1996-2017 [cit. 2017-05-19]. Dostupné z: http://mesta.obce.cz/

[32] *Akční ceny, aktuální letáky a zajímavé slevy | AkcniCeny.cz* [online]. c2000-2017 [cit. 2017-05-19]. Dostupné z: https://www.akcniceny.cz/

[33] *Elasticsearch: RESTful, Distributed Search & Analytics | Elastic* [online]. c2017 [cit. 2017-05-20]. Dostupné z: https://www.elastic.co/products/elasticsearch

[34] *The Web framework for perfectionists with deadlines | Django* [online]. c2005-2017 [cit. 2017-05-20]. Dostupné z: https://www.djangoproject.com/

[35] *Bootstrap · The world's most popular mobile-first and responsive front-end framework.* [online]. [cit. 2017-05-20]. Dostupné z: http://getbootstrap.com/

[36] *Interactive JavaScript charts for your webpage | Highcharts* [online]. c2017 [cit. 2017-05-20]. Dostupné z: https://www.highcharts.com/

[37] Kaufland má problém. Pokladní v Praze prodává s vytetovaným hákovým křížem na ruce | info.cz. *Info.cz - Česko, svět, politika, zpravodajství, analýzy, události, byznys* [online]. c2001-2017 [cit. 2017-05-21]. Dostupné z: http://www.info.cz/magazin/kaufland-ma-problem-pokladni-v-praze-prodava-s-vytetovanym-hakovym-krizem-na-ruce-8550.html

[38] Pokladní měla na ruce vytetovaný hákový kříž. Kaufland ji vyhodil - iDNES.cz. *IDNES.cz – s námi víte víc* [online]. c1999-2017 [cit. 2017-05-21]. Dostupné z: http://ekonomika.idnes.cz/pokladni-s-hakovym-krizem-na-ruce-kaufland-fj1-/ekonomika.aspx?c=A170427_092939_ekonomika_rny

[39] Neonacistka za pultem? V Kauflandu měli pokladní s hákovým křížem | Blesk.cz. *Blesk.cz - zprávy, celebrity, sport, zábava* [online]. c2001-2017 [cit. 2017-05-21]. Dostupné z: http://www.blesk.cz/clanek/zpravy-udalosti/465383/neonacistka-za-pultem-v-kauflandu-meli-pokladni-s-hakovym-krizem.html

Príloha A

Obsah CD

Priložený CD nosič obsahuje:

- Zložku `data`, v ktorej sa nachádzajú dáta, ktoré boli v priebehu tvorby práce zhromaždené alebo vytvorené
- Zložku `poster`, v ktorej sa nachádza plagát prezentujúci prácu
- Zložku `report`, v ktorej sa nachádza elektronická verzia tejto práce spolu s jej zdrojovým súborom
- Zložku `source_codes`, v ktorej sa nachádzajú zdrojové kódy implementovaného systému (zložka `system`), zdrojové kódy webového rozhrania (zložka `gui`) a ostatné skripty vytvorené a použité v priebehu tvorby práce (zložka `other_scripts`)
- Textový súbor `readme.txt` popisujúci obsah CD nosiča a inštaláciu webového rozhrania