EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/I/2017/1220388174

VYTVORENIE JEDNODUCHÉHO MODELU NA ZBER, TRIEDENIE A VYHODNOTENIE DÁT S RELAČNÝM VÝBEROM

Diplomová práca

2017 **Bc. Peter Černek**

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

VYTVORENIE JEDNODUCHÉHO MODELU NA ZBER, TRIEDENIE A VYHODNOTENIE DÁT S RELAČNÝM VÝBEROM

Diplomová práca

Študijný program: Informačný manažment (Jednoodborové štúdium, inžiniersky II. st., denná forma)

Študijný odbor: Kvantitatívne metódy v ekonómii

Školiace pracovisko: Katedra aplikovanej informatiky

Vedúci záverečnej práce: RNDr. Ľubomír Turňa, CSc.

Bratislava 2017 **Bc. Peter Černek**

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum: .....

Poďakovanie

Chcel by som sa poďakovať pánovi RNDr. Ľubomírovi Turňovi, CSc. za odbornú pomoc a usmernenie pri písaní diplomovej práce, za cenné rady a informácie a v neposlednom rade za ochotu.

Abstrakt

Černek, Peter: Vytvorenie jednoduchého modelu na zber, triedenie a vyhodnotenie dát s relačným výberom - Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky - Vedúci záverečnej práce: RNDr. Ľubomír Turňa, CSc.- Bratislava: FHI EU, 2017, počet strán 80

Cieľmi predkladanej diplomovej práce je návrh a implementácia modelu, prostredníctvom ktorého dokážeme extrahovať dáta z Internetu, transformovať neštruktúrované dát do podoby vhodnej pre získanie znalostí a uloženie týchto dát do formy, aby boli v budúcnosti k dispozícii pre štatistické analýzy. Diplomová práca je rozdelená do 5 kapitol. V prvej kapitole charakterizujeme pojmy v oblastiach extrakcie dát z Internetu, transformácie neštruktúrovaných dát a možnostiach uloženia dát. Druhá kapitola sa zaoberá definovaním cieľov a metodík použitých v predkladanej diplomovej práce. Tretia kapitola je venovaná návrhu všeobecného modelu zameriavajúceho sa na tri oblasti definované v prvej kapitole. Štvrtá kapitola sa venuje implementácii navrhovaného modelu z tretej kapitoly na riešenie konkrétnej úlohy. Úloha sa týka automatizovaného sťahovania recenzií z vybratých webových zdrojov, získanie informácii z týchto neštruktúrovaných dáta a tieto informácie uložiť, aby boli k dispozícii pre štatistické metódy na získanie znalostí. Implementácia modelu je realizovaná v dvoch vývojových prostrediach: RStudio a MS SQL Server. V záverečnej kapitole prezentujeme výsledky zo štvrtej a kapitoly a zamýšľame sa nad možnosťami, ako zlepšiť jednotlivé fázy modelu pri riešení našej úlohy.

Kľúčové slova: extrakcia dát, Text Mining, databázy, neštruktúrované dáta, webové stránky

Abstract

Černek, Peter: Creating a simple model of collecting, sorting and evaluation of data with relational selection – Economic university in Bratislava, Faculty of Economic Informatics, Department of Applied Informatics, Supervisor: RNDr. Ľubomír Turňa, CSc. – Bratislava: FHI EU, 2017, 80 pages

The aims of the master thesis are to design and implement a model, through which we will be able to extract data from the Internet, to transform unstructured data into the form suitable for obtaining of information and to store data into a data storage, in which it will be available for statistical analysis in the future. The document is divided into fifth main chapters. In the first chapter, we characterize terms used in the areas: extracting data from websites, Text Mining and data storages. Second chapter is focusing on defining goals and methods, which are used in the thesis. In the third part, we design the model focusing on the areas mentioned in the previously chapters. Fourth chapter introduces the practical implementation of our model for solving of task. The task consists of automatic extracting reviews of users from websites, obtaining information from unstructured this data and storing information into a selected data storage. For the application of the model, we use two open-source software: RStudio a MS SQL Server. In the last chapter, we present the result of fourth chapter a we think about improvements of model for our task.

Key words: extraction of data, Text Mining, databases, unstructured data, websites

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam fragmentov

Úvod

V priebehu posledných rokov je využitie informačných technológií čoraz viac orientované na oblasť spracovania dát. Je to spôsobné dopytom spoločností, ktoré čelia veľkému množstvu dát v štruktúrovanej alebo neštruktúrovanej podobe z rôznych zdrojov. Schopnosť získavať dáta z rôznych zdrojov a dolovať z nich informácie o zákazníkoch, konkurencií alebo trhu, umožňujú spoločnosti rýchlo reagovať na zmeny, robiť správne rozhodnutia a získavať nových zákazníkov. Táto schopnosť zahrňuje efektívne zhromažďovanie veľkého množstva dát, ich spracovanie do vhodnej podoby, uchovávanie a analyzovanie. Tieto činnosti sú súčasťou procesného modelu spoločností na zber, uchovávanie a analyzovanie dát. Ich rozsah a náročnosť závisia od množstva dát, s ktorými daná organizácia pracuje, a od znalostí zamestnancov spoločnosti.

Cieľom predkladanej diplomovej práce je navrhnúť model na zber, triedenia a analyzovanie dát. Model sa skladá z viacerých procesov, ktoré sú nevyhnutné na získanie dát z externých zdrojov, ich transformáciu do vhodnej podoby, získať znalosti z týchto dát a ukladať ich, aby boli dostupné kedykoľvek a vo vhodnej forme. Súčasťou diplomovej práce je aplikácia navrhnutého modelu vo vývojovom prostredí R a databázovom systéme MS SQL Server.

V diplomovej práci sa zameriavame na extrahovanie dát z webových stránok. Webové stránky obsahujú veľké množstvo štruktúrovaných, aj neštruktúrovaných dát, v ktorých môžu byť ukryté cenné informácie pre spoločnosť. Každá webová stránka má inú štruktúru a dáta sú uložené v rôznych formátoch. Preto časť diplomovej práci zahŕňa procesy, ktoré detailne popisujú analyzovanie webových stránok, extrakciu dát a transformáciu dát do vhodnej podoby. Ďalšia časť práce je zameraná na analýzu dát a získania informácii z týchto dát. Posledná časť sa týka procesov ukladania dát a spôsobov, ako ukladať dáta.

1 Súčasný stav v oblasti extrakcie, transformácie a analýzy dát

V prvej kapitole vymedzíme pojmy v troch oblastiach: Internet, transformácie neštruktúrovaných dát a relačné databázy. Každá z týchto oblastí je dôležitá pre náš model, navrhnutý v tretej kapitole. Na začiatku definujeme oblasť Internetu z pohľadu zdroja dát. Následne vymedzíme techniky a algoritmy, ktoré dokážu pracovať so všetkými typmi dát. V práci sa zameriavame najmä na neštruktúrované dáta. V poslednej časti kapitoly sa venujeme relačným modelom a databázam, do ktorých je možné ukladať dáta, aby boli dostupné v budúcnosti vo vhodnej forme pre ďalšie analýzy.

1.1 Zdroje dát

Spoločnosti získavajú dáta z rôznych zdrojov. Tieto zdroje môžeme rozdeliť do dvoch skupín: interné a externé. Do skupiny interných dát patria systémy a aplikácie ako CRM alebo ERP systémy, v ktorých sú dáta tvorené zamestnancami samotných spoločností. Dáta z externých zdrojov pochádzajú z webových stránok, sociálnych sietí alebo súborov, ktoré sú dostupné na Internete. Tieto dáta sú uložené v rôznych formátoch, pričom pred analýzou a ukladaní dát je nutné dáta transformovať do vhodnej formy. Dáta môžeme rozdeliť na [1]:

- Štruktúrované – dáta sú uložené v presne definovaných schémach, ktoré sú často popísané. Dáta sú uložené v tabuľkách alebo v dokumentoch, ktoré majú presne definovanú štruktúru (XML, JSON, atď.).
- Neštruktúrované – dáta nie sú v presne definovaných štruktúrach. Objavujú sa v zvukových záznamoch, textových dokumentoch, webových stránkach alebo statusoch na sociálnych sieťach.
- Semištruktúrované – časť dát je v štruktúrovanom formáte a časť v neštruktúrovanom. Napr. dáta v emaily môžeme označiť ako semištrukturované. Dáta v prílohách a v texte správy sú neštruktúrované a údaje o emaily (odosielateľ, prijímateľ, dátum poslania, atď.) sú štruktúrované dáta.

Spoločnosti ukladajú štruktúrované dáta do relačných databáz a z týchto dát je možné pomocou štatistických metód získavať informácie, užitočné pre zamestnancov v procese rozhodovania. Neštruktúrované dáta nie je možné uložiť do databáz a v neštruktúrovanej podobe, majú malú alebo žiadnu informačnú hodnotu pre spoločnosť. Podľa analýzy [2] až

80% podnikových dát nie je v kvantitatívnej alebo štruktúrovanej podobe, ale je možné ich transformovať do podoby vhodnej pre relačné databázy.

1.1.1 Webové sídla a webové služby

Internet sa stal primárnym zdrojom informácii pre ľudí aj firmy na svete. Ľudia využívajú informácie z internetu pre svoje osobné účely alebo v ich pracovnom živote. Na začiatku je dôležité zadefinovať základné pojmy z internetového prostredia ako webová stránka, webové sídlo alebo informačná architektúra, s ktorými budeme pracovať v ďalších častiach.

Webová stránka je časť webového sídla, ktorá sa zobrazuje užívateľovi v prehliadači a má svoju unikátnu adresu. Pozostáva z klientskej a serverovej časti. Klientska časť je tvorená programovacími jazykmi JavaScript, Flash alebo Silverslight a značkovacími jazykmi HTML a XHTML. Príkazy týchto programovacích jazykov sú priamo spúšťané vo webovom prehliadači pri interakcii s užívateľmi. Serverová časť zahŕňa príkazy z programovacích jazykov ako PHP, ASP.NET alebo JAVA. Dáta v serverovej časti nie sú zobrazené v zdrojovom kóde webovej stránky a príkazy týchto programovacích jazykov vykonávajú definované činnosti [3]. **Webové sídlo** sa skladá z viacerých webových stránok, ktoré sú navzájom prepojené. Poskytuje ucelené informácie a služby, za účelmi ktorých bolo webové sídlo navrhované a vytvorené. Pri webovom sídle je dôležitý jeho návrh, navigácia, obsah a klasifikácia, ktoré určujú či bude webové sídlo užitočné pre užívateľov a majiteľa/majiteľov sídla. **Informačná architektúra** je oblasť, ktorá sa zaoberá návrhmi a obsahmi webových sídiel. Cieľom informačnej architektúry je priehľadnosť štruktúry webových stránok pre užívateľov a majiteľov webových stránok, aby bolo možné obsluhovať webovú stránku a dáta na webových stránkach boli ľahko dostupné. **Webové štandardy** zahŕňajú nástroje, pomocou ktorých sa vytvárajú webové stránky. Zabezpečujú dizajn webových stránok a ich funkcionalitu. Cieľom webových štandardov je zobrazenie a správna funkcionalita webovej stránky v moderných prehliadačoch, ale aj v ich starších verziách. Boli vytvorené organizáciou W3C, aby webové stránky boli kompatibilné so všetkými webovými prehliadačmi. Definujú štandardy a smernice webových stránok, čím uľahčujú prácu webovým programátorom a dizajnérom, ktorí nemusia vyvíjať stránku niekoľkokrát, pre každé zariadenie a prehliadač zvlášť. Tým sa znižujú finančné a časové náklady pre spoločnosti pri vytváraní a aktualizácii webových sídiel. Zabezpečujú dostupnosť stránok na starších alebo nových webových prehliadačoch a pri aktualizácii

prehliadačov webové stránky nemusia byť aktualizované. Stránka je webovými prehliadačmi rozdelená na tri základné časti: štruktúra, prezentácia a správanie[4].

Webové štandardy – štruktúra

Pri tvorbe štruktúry webovej stránky sa využívajú programovacie jazyky HTML a XHTML, ktoré sú označované ako značkovacie jazyky. Organizácia W3Techs sleduje vývoj technológii využívaných na webových stránkach. Organizácia sleduje vyše 10 miliónov webových sídiel. Na základe reportu zobrazeného na ich webovej stránke, jazyk HTML je využívaný v 74,2% webových stránok a XHTML v 26,3% zo sledovaných webových stránok [3]. Existujú webové stránky, na ktorých sa využívajú oba spomínané jazyky. Webový dokument je tvorený pomocou HTML tágov a dát, ktoré sú zobrazované na stránkach. Prostredníctvom tágov definujeme ako sú texty a grafické prvky zobrazené na stránke. HTML dokument začína tágom, v ktorom je zverejnené, ktorá verzia HTML kódu je použitá v dokumente, aký jazyk je použitý na webovej stránke a url adresa webového sídla. Táto časť nie je povinná, ale slúži ako informácia pre osobu, ktorá by chcela pracovať so zdrojovým kódom stránky. Ďalší tág <html> definuje začiatok html dokumentu. Tento tág je párový, na konci dokumentu je jeho druhá časť </html>, ktorá definuje koniec HTML dokumentu. V rámci tohto tágu sú definované dve hlavné časti. V prvej časti v tágu <head> je názov webovej stránky a informácie o štýle stránky. V druhom tágu <body> sú definované všetky prvky, ktoré tvoria obsah stránky. Kombináciou jazyka XML a HTML vznikol jazyk XHTML. Výhodou jazyka XHTML je , že dokumenty v tomto jazyku neobsahujú chybne definované tágy, na rozdiel od dokumentov v HTML jazyku. Chybne definovaný tág môže spôsobovať problém webovému prehliadaču s načítaním obsahu webovej stránky alebo programom a automatom, ktoré získavajú dáta z týchto stránok. Úlohou jazyka XML je prenos dát z jednej aplikácie do druhej a popísanie údajov. HTML slúži na zobrazenie dát na stránke.

Webové štandardy – prezentácia

Ich súčasťou sú jazyky ako CSS alebo Compression, ktoré definujú formát stránok, typ a farbu písma a podobne. Použitie týchto jazykov na stránke zabezpečuje oddelenie obsahu od štruktúry stránky a preto pri zmene písma nezasahujeme do štruktúry stránky [4].

Webové štandardy – správanie

Jazyky v rámci tejto skupiny umožňujú definovať správanie stránky a definovať

objekty na stránke. Z celej škály programovacích jazykov, či už skriptovacích alebo objektovo orientovaných, vývojári najčastejšie pri webových stránkach využívajú PHP a JavaScript. Podľa W3Techs, JavaScript je využívaný na 94,4% stránok a PHP na 82,5% [3]. PHP jazyk je označovaný ako skriptovací jazyk, v ktorom sa programujú skripty na stranu servera. Server je miesto, odkiaľ sa načítava stránka, ktorá sa má zobraziť užívateľovi v jeho prehliadači. Týmto jazykom zabezpečujeme ukladanie vstupov od užívateľa do databáz alebo zobrazenie obsahu na stránku na základe návštevníka webovej stránky. PHP skript je zvyčajne súčasťou HTML dokumentu, pričom je možné kombinovať PHP a HTML príkazy. Na odlíšenie PHP príkazov od HTML sa používajú tágy <?php (niekedy len <?) na začiatku a ?> na konci. V rámci týchto tágov môžeme využiť príkazy PHP jazyka [5].

JavaScript je skriptovací jazyk na strane klienta, čo znamená, že príkazy zbehnú priamo v prehliadači. Prostredníctvom JavaScriptu môžeme definovať správanie webovej stránky po prihlásení užívateľa alebo na základe reakcie na jeho vstupy. Príkazy jazyka JavaScript na webovú stránku je možné pridať pomocou párového tágu <script>. Prostredníctvom oboch spomínaných jazykov môžeme definovať premenné, funkcie, cykly a podobne, čo sa značkovacími jazykmi nedá.

S rozvojom webu ako informačného média sa webové stránky stali zdrojom informácií užitočné pre každého užívateľa. S týmto rozvojom vznikol dopyt po extrakcii a analyzovaní dát z týchto stránok. Úlohou Web Miningu je získať rôzne typy dát generované webovými stránkami. Web Mining je definovaný ako „Data Mining-ová technika na extrakciu znalostí zo štruktúry webu, z obsahu webovej stránky a z logovacích súborov servera“ [6]. Na základe typu sťahovaných dát delíme Web Mining do troch typov: Web Content Mining, Web Structure Mining a Web Usage Mining.

1.2 Web Content Mining

Je proces extrahovania rôznych typov dát ako texty, obrázky, videá alebo dáta v tabuľkách z webových dokumentov. Web Content Mining je možné použiť v úlohách, v ktorých chceme získať dáta z jedného alebo viacerých webových dokumentov. Aplikovať extrakciu dát je možné v dokumentoch ako XML alebo HTML, ktoré obsahujú štruktúrované aj neštruktúrované dáta. Štruktúra webového dokumentu je tvorená príkazmi programovacích jazykov a z tágov, v ktorých je uložené množstvo dát. Z tágov je možné zistiť, aký typ dát je uložený v dokumente. Ďalšou možnosťou je odstrániť z dokumentu

tágy, čím je dokument čitateľný pre program, ktorý analyzuje text. Techniky na extrakciu dát z webových dokumentov môžeme rozdeliť do viacerých prístupov, ktoré sa založené na postupoch a algoritmoch a ich výsledkom je systém nazývaný web wrapper.

1.2.1 Web wrappery

Web wrapper je procedúra, ktorá umožňuje použiť jeden alebo viacero algoritmov, ktoré sú schopné nájsť dáta na základe kritérií [8]. Pomocou wraperov dokážeme extrahovať dáta z neštruktúrovaných a semištruktúrovaných zdrojov do formy štruktúrovaných dát, získať z nich informácie pre ďalšie procesy, čím je možné celý proces extrakcie zautomatizovať. Existuje viacero prístupov pri tvorbe web wraperov, ale ich cieľom by mala byť:

- funkčnosť,
- pravidelná extrakcia informácií,
- schopnosť vedieť reagovať na zmeny, spôsobené zmenou štruktúry dokumentu.

Na začiatku tvorby wrapera je nutné definovať pravidlá, na základe ktorých sa budú dáta automatizovane extrahovať. Štruktúra webového dokumentu je tvorená tágami a dátami uložených v nich. Pomocou tágov vieme identifikovať typ dát, ale koncovému používateľovi neposkytujú informačnú hodnotu. Dát je v dokumente veľké množstvo a nie všetky sú potrebné a užitočné pre používateľa. Preto úlohou pravidiel je extrahovať len tie dáta, ktoré prinesú užívateľovi zisk alebo pomôžu pri jeho rozhodovaní. Pretože každý dokument má špecifickú štruktúru, je potrebné definovať extrakčné pravidlá pre každý dokument. Existujú 4 typy wraperov podľa spôsobu extrahovania dát a prístupu k dátam [8]: Regular-expression-based, Logic-based, Tree-based a Machine Learning.

Regular-expression-based

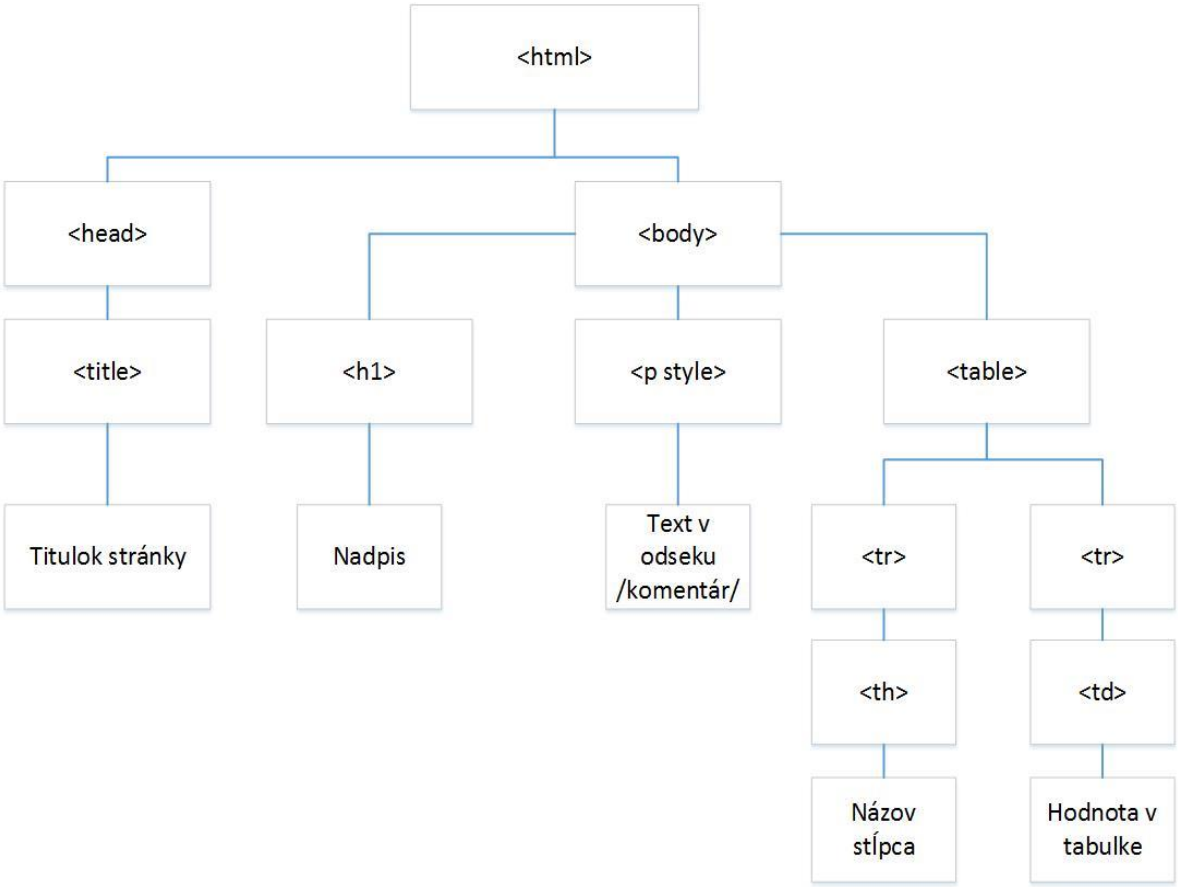
Regular-expression-based prístup pri tvorbe výrazov, ktoré generujú extrakčné pravidlá. Čím lepšie sú definované, tým kvalitnejšie budú pravidlá a následne aj dáta a šablóny nájdené pravidlami. Výhodou tohto prístupu je nezávislosť wrapera od HTML kódu. Používateľ tiež môže hocikedy zadať výrazy a wraper dokáže definovať ďalšie pravidlá a extrahovať nové dáta [8]. Nevýhodou prístupu je, že používateľ nemusí rozumieť vygenerovaným pravidlám a wraper môže mať problém s identifikáciou vzťahu dvoch typovo odlišných hodnôt (napr. číslo a text).

Logic-based prístup

Logic-based od predchádzajúceho prístupu považuje webový dokument za semištruktúrovaný dokument a pri extrahovaní dát z dokumentov využíva tágy. Wraper neextrahuje dáta priamo z dokumentu, ale z dokumentového objektového modelu (DOM), v ktorom sú dáta zoskupené do stromovej štruktúry. DOM je tvorený na základe tágov v dokumente a pomocou uzlov, ktoré sú definované postupnosťou tágov. Uzly vyjadrujú „cestu“ akou je možné dostať sa k dátam. DOM pozostáva z týchto objektov [9]:

- Elementy – tágy z webového dokumentu
- Atribúty – prvky definujúce štýl dokumentu a textu
- Komentár – text popisujúci časť v dokumente
- Text – texty zobrazené na webovej stránke

Na Obr. 1 je príklad dokumentového objektového modelu, ktorý sa skladá z viacerých úrovní.



Obr. 1 Štruktúra dokumentového objektového modelu

Na prvej úrovni je graf definovaný tágom <html>, ktorý sa nachádza v každom webovom dokumente. Druhá úroveň grafu sa vetví na dve časti a je tiež tvorená elementami. V prvej časti sú prvky nachádzajúce sa v rámci tágu <head> v dokumente a v druhej časti v tágu <body>. Ďalšiu úroveň tvoria elementy a atribúty z jazyka CSS, ktoré definujú typ dát a (spôsob,) ako sa zobrazujú na webovej stránke. V ďalších úrovniach grafu sú použité všetky štyri základné objekty DOM a každý uzol v grafe končí textom, ktorý je zobrazený na webovej stránke a môže byť pre užívateľa užitočný. Na tvorbu stromu DOM a extrahovanie informácii z uzlov je možné použiť dopytovací jazyk Xpath.

Xpath poskytuje možnosti ako extrahovať uzle a dáta z dokumentu. Bol vyvíjaný organizáciou W3C, ktorá sa zaoberá aj návrhom webových dokumentom, čím je zabezpečené, že novinky spôsobené v oblasti webových dokumentov spôsobia aktualizáciu Xpath jazyka a pridanie nových funkcionalít. Pomocou tohto jazyka môžeme extrahovať len jeden alebo viacero objektov z DOM. Výhodou Xpath jazyka je ľahká syntax na pochopenie a vytvorenie kódu, na extrahovanie dát a možnosť nájdenia vzťahov medzi dátami odlišného typu. Pri použití jazyka Xpath je nevyhnutné počítať s dvomi komplikáciami:

- Menej než 20% webových dokumentov úplne spĺňa štandardy W3C a pri použití na webové stránky, ktoré nespĺňajú štandardy môžeme extrahovať neúplné dáta [10].
- Aktualizácia alebo pridanie nových prvkov a funkcionalít do webového dokumentu spôsobuje nevyhnutnú aktualizáciu wrapera v Xpath jazyku.

Xpath jazyk poskytuje užitočnú a ľahko použiteľnú možnosť na extrakciu dát z webových dokumentov, avšak je dôležité počítať s pravidelnou údržbou navrhnutého wrapera v tomto jazyku [11].

Tree-based

Tree-based prístup je zameraný na techniku extrahovania dát z dokumentov pozostávajúci z dvoch krokov: segmentácia a extrakcia dát zo segmentov. V segmentácii dochádza k rozdelenie súvisiacich častí webových dokumentov do segmentov. V tejto časti nedochádza k extrahovaniu informácii, ale vytvára sa graf DOM a aplikuje sa priestorová technika (Spatial reasoning). Priestorová technika je užitočná pri hľadaní medzier/ chýb v dátových záznamoch. Dáta získava priamo z webovej stránky, ktorá je dostupná užívateľovi aj vo webovom prehliadači. Pomocou súradníc X a Y je možné nadefinovať presnú súradnicu dát vo webovom prehliadači. Táto metóda je užitočná najmä v prípadoch,

ak webový dokument nie je navrhnutý podľa štandardov. Nevýhodou je náročnosť aplikovania tejto techniky a možnosť, že dáta nadobúdajú rôzne súradnice závislé od použitia iných prvkov na stránke, ako napríklad - od animácií. V druhom kroku dochádza k extrahovaniu textov z jednotlivých stromových grafov do hlavného grafu DOM, pretože niektoré dáta sa môžu nachádzať vo viacerých grafoch, ktoré vznikli v prvom kroku. Počas tohto kroku prebieha odstraňovanie, prepisovanie uzlov a transformácia uzlov do iných. Cieľom je vytvoriť prehľadný DOM graf a zabezpečiť, aby sa v ňom neobjavovali duplicitné uzly a dáta [8].

Machine Learning

Machine Learning prístup sa zameriava na extrakciu dát z webových stránok v rámci vybranej problematiky. Systém na začiatku potrebuje získať znalosti z vybranej problematiky. Doménový expert v skúmanej oblasti vloží manuálne do systému znalosti z tejto problematiky z vybraných webových stránok. Tieto webové stránky by mali mať rozdielnu štruktúru, aby bol systém schopný naučiť sa extrahovať dáta z rôznych typov dokumentov. V prípade, že sú v systéme dáta z týchto stránok, systém sa pomocou štatistických metód a metód umelej inteligencie naučí dáta extrahovať.

1.3 Web Structure Mining

Web Structure Mining je založený na tvorbe webového grafu, ktorý pozostáva z uzlov zastupujúce webové stránky a hrán tvorené z hyperlinkov webových stránok, ktoré spájajú súvisiace webové stránky. Zameriava sa na extrakciu dát, pomocou ktorých môžeme vyriešiť tri typy úloh [7]:

- hodnotenie kvality webových stránok,
- získanie špecifických typov dát ako citácie, grafy, atď.,
- triedenie webových dokumentov do skupín na základe rôznych kritérií.

Tieto typy úloh môžeme vyriešiť pomocou hyperlinkovej analýzy a použitím rôznych štatistických metód.

1.3.1 Hyperlinková analýza

Hyperlink je referencia alebo odkaz použitý v dokumente, ktorý sa odkazuje na iný dokument alebo špecifické časti dokumentu. Existujú dva typy hyperlinkov: intra-document

a inter-document hyperlink. Intra-document hyperlink spája odlišné časti/stránky webového sídla a inter-document hyperlink spája odlišné webové sídla [6]. Cieľom hyperlinkovej analýzy je ohodnotiť webové stránky a vytvoriť poradie od najlepšie ohodnotenej stránky po najhoršiu. Existuje viacero algoritmov, pomocou ktorých je možné vypočítať hodnoty stránok, ktoré budú určovať kvalitu stránky. Sú to napríklad PageRank, OPIC alebo HITS. Algoritmy sa líšia v postupnosti krokov, avšak všetky predpokladajú na začiatku, že budú hodnotiť stránky alebo dokumenty, ktoré súvisia.

HITS algoritmus je založený na delení stránok do dvoch skupín: hubs a authorities. Stránky z týchto skupín spolu súvisia, nakoľko stránka z jednej skupiny odkazuje na stránky z druhej skupiny. Algoritmus prideľuje každej stránke dve hodnoty, ktoré určujú jej poradie vo výslednom zozname. Ich veľkosť závisí od počtu prepojení medzi stránkou a ostatnými stránkami v skupinách. Prvé číslo vyjadruje prepojenosť stránky s ostatnými stránkami v skupine hubs a druhé v authorities. Na konci procesu získame hodnotenie jednotlivých webových sídiel [12].

1.3.2 Triedenie dokumentov

Triedenie webových dokumentov môže byť súčasťou procesu, ktorého cieľom je extrakcia informácií z webových dokumentov alebo tvoriť celý proces, ktorého cieľom je zatriediť dokumenty do skupín. Triede webových skupín je možné pomocou týchto techník [13]:

- Klasifikácia - metóda triedi dokumenty do predom definovaných skupín. Tieto skupiny vznikli z testovacích dát. Skupiny majú špecifické charakteristiky a je dôležité, aby testovacie dokumenty boli podobné s tými, ktoré chceme klasifikovať. Podobnosťou sa myslí, aby sa dokumenty týkali podobných problematík a obsahovali podobné dáta.
- Zhlukovanie – na rozdiel od klasifikácie sa pri zhlukovaní nevyužívajú testovacie dáta. Taktiež sa dokumenty triedia do skupín nazývané klastry, avšak tieto skupiny pred začiatkom použitia metódy nie sú známe. Klastre vznikajú počas procesu zhlukovania a každý z nich by mal byť definovaný, aké typy dokumentov obsahuje.
- Sumarizácia - poskytuje základné informácie o dokumente a užívateľovi možnosť získať základné informácie o dokumente, bez potreby naštudovania celého dokumentu. Sumarizácia dokumentu je vypracovaná extrahovaním informácií z textov v dokumente pri použití štatistických a heuristických metód.
- Selekcia - hľadanie dokumentov, ktoré spĺňajú zadané kritéria vo forme kľúčových slov.

- Vizualizácia - technika vhodná ako doplnok pre prvé dve techniky. Pomocou nej môžeme graficky zobraziť vzťahy v rámci skupín resp. klastrov, medzi dokumentami a medzi skupinami resp. klastrami navzájom.

Tieto techniky je možné navzájom kombinovať. Záleží to najmä od typu úlohy a zobrazenia výsledku.

1.4 Web Usage Mining

Web Usage Mining sa zaoberá zberom troch typov dát o užívateľoch webových stránok:

a) dáta generované z logov,
b) osobné dáta o užívateľoch ako meno alebo adresa, ktoré vznikajú cez formuláre priamo na webových stránkach,
c) dáta zachytávajúce správanie užívateľov na webových stránkach.

Dáta pozostávajúce z logov sú generované serverom a môžeme z nich získať informácie o IP adresách, času stráveného na stránke alebo o tom, aký webový prehliadač užívateľ využíva. Výhodou takýchto typov dát je, že zachytávajú informácie o registrovaných ale aj neregistrovaných používateľoch. Zo servera môže byť generovaný ECLF log file format, vo formáte, Tab. 1.

Tab. 1 ECLF Log File Format

IP Address	rfc931	authuser	Date and time of request	request	status	bytes	referer	user agent
128.101.35.92	-	-	[09/Mar/2002:00:03:18-0600]	"GET/~harum/HTTP/1.0"	200	3014	http://www.cs.umn.edu/	Mozilla/4.7 [en] (X11; I; SunOS 5.8 sun4u)

Súbor pozostáva 9 stĺpcov, v ktorých sú zaznamenané informácie o užívateľovi, ktorý sa prihlásil na webovú stránku a o udalosti, ktorá nastala na webovej stránke. Udalosť môže nastať prihlásením užívateľa na jednu z webových stránok alebo vykonaním akcie na stránke. Akcia na stránke môže byť zaznamenaná po kliknutí na jeden z grafických nástrojov na stránke ako sú napríklad tlačidlá, alebo po kliknutí na hyperlink alebo video.

Súbor poskytuje nasledovné informácie [7]:

- IP address - identifikačné číslo počítača v sieti,
- Rfc931 - meno počítača v sieti,

- Authuser – poskytuje informáciu či je používateľ prihlásený na webovej stránke a pod akým menom,
- Date and time of request - dátum a čas uskutočnenia akcie,
- Request – informuje, z ktorej časti webovej stránky bola akcia vykonaná,
- Status – číslo, ktoré identifikuje, akú odpoveď dostane po jeho akcii,
- Bytes – počet bytov vznikajúcich pri spojení,
- Referer - URL adresa, z ktorej sa užívateľ prihlásil na stránku,
- User agent - názov webového prehliadača, ktorý používa užívateľ na prezeranie webovej stránky.

Z týchto dát je možné napríklad zistiť, z ktorých webových stránok prichádzajú ľudia na webovú stránku, alebo v prípade problémov na stránke, aké prehliadače využívajú.

Osobné dáta o užívateľoch je možné získať prostredníctvom webových formulárov. Následne sú tieto údaje ukladané do databáz a je možné ich analyzovať. Ich kvalita závisí najmä od užívateľov, či sú ochotní poskytnúť pravdivé osobné informácie a od kvality formulára. Kvalitne navrhnutý formulár môže obmedziť riziko uchovávania nekvalitných dát v databáze. Napríklad úspešné odoslanie vyplneného formuláru len v prípade zadania emailu so znakom @ a bez interpunkcií, prípadne, aby všetky polia a otázky boli zrozumiteľné pre užívateľa. Pomocou tohto typu zdroja dát môžeme zistiť percentuálny podiel zákazníkov podľa veku či pohlavia, alebo odkiaľ pochádzajú užívatelia stránky.

Analýza dát generovaných z aktivity užívateľa na webovej stránke sa nazýva Clickstream Analysis. Počas tejto analýzy prebieha zber, analyzovanie a reportovanie dát, ktoré sú generované po každom kliknutí na hocijaký prvok na stránke. Pomocou analýzy je možné zachytiť dáta o potencionálnych, aj súčasných klientoch. Clickstream analýza je vhodná najmä pre spoločnosti, ktoré cez ich webové sídlo ponúkajú produkty alebo služby. Spoločnosť môže zachytiť veľké množstvo dát, ako napríklad [14]:

- či si užívatelia aj niečo kúpili, alebo len pozerali na produkty alebo služby,
- koľko času strávili pozeraním na jednotlivé produkty,
- či sa na stránku vrátili,
- ktoré produkty alebo služby sú najviac sledované a ktoré najmenej.

Existuje oveľa viac otázok, na ktoré sme schopní použitím Clickstream Analysis odpovedať a tým identifikovať správanie klienta a ponúknuť im produkt alebo službu, o

ktorú budú mať záujem.

1.5 Text Mining

Existuje viacero prístupov, ktoré definujú Text Mining iným spôsobom. Každý z prístupov definuje Text Mining z iného pohľadu [15], podľa:

- techniky extrakcie faktov z textu,
- procesu, ktorý zahŕňa sériu krokov ako extrakcia textu z dokumentov, analyzovanie textu a vizualizáciu výsledkov,
- oblasti, podobnej Data Miningu, ktorá zahŕňa metódy a techniku, pomocou ktorých dokážeme extrahovať informácie z neštruktúrovaného textu.

V tejto práci sa prikláňame poslednému prístupu, v rámci ktorého techniky a štatistické metódy Text Mining-u sú použité v oblasti dolovania dát, kedy je úlohou získanie informácií zo všetkých typov dokumentov alebo textov.

1.5.1 Techniky Text Miningu

Text Mining sa zameriava na prácu s veľkým počtom textových dokumentov. Na začiatku je nutné spracovať texty z týchto dokumentov a texty uložiť do štruktúr vhodných pre analýzu. Väčšina Text Miningových prístupov je založená na myšlienke, že „textový dokument môže byť reprezentovaný skupinou slov, tzv.. textový dokument je popísaný na základe skupiny slov v bag-of-words reprezentácii“ [15]. Jednotlivé techniky je možné kombinovať v závislosti od cieľov. Techniky môžeme rozdeliť na pre-procesingové a techniky uloženia dát. *Tokenizáciou* sa odstraňujú z dokumentu znaky, ktoré nemajú informačnú hodnotu, akými sú interpunkčné znamienka alebo biele miesta v rámci textov. V rámci tokenizácie sa definujú nasledovné premenné [15]:

- množina, zahrnujúca všetky skúmané dokumenty,
- množina slov, objavujúcich sa v dokumentoch,
- frekvencia určujúca počet výskytov slov v dokumentoch.

Po extrahovaní slov z textov je nutné znížiť množinu slov filtráciou, lemitizáciou a indexami. *Filtráciou* je možné odstrániť slová na základe podmienok alebo slová, ktoré neposkytujú informačnú hodnotu, napríklad predložky alebo spojky. Nemusia sa odstraňovať slová len na základe slovného druhu, ale aj slová vyskytujúce sa vo veľkých

počtoch alebo len zriedka v dokumentoch. *Lemitizáciou* sa snažíme upraviť slová do základných tvarov. Pri tejto technike je dôležité, aby slová v dokumentoch boli gramaticky správne a boli spisovné. Tento proces je užitočný najmä v jazykoch ako slovenčina, keď slová môžu nadobúdať viacero tvarov. Pri *indexovaní* dochádza k ohodnoteniu jednotlivých slov v dokumentoch. K slovám sú priradené číselné hodnoty (entropie), ktoré určujú schopnosť slov oddeliť dokumenty na základe vyhľadávania kľúčových slov. Čím sa slovo častejšie vyskytuje v dokumentoch, tým má nižšiu hodnotu entropie a neodporúča sa ho použiť na zníženie počtu slov v dokumentoch.

Ďalšie procesy, ktorými je možné zredukovať počet slov, sú procesy zaoberajúce sa jazykovou stránkou slov a slovných spojení. Do tejto skupiny patri tieto 4 metódy [15]:

- určovanie slovného druhu slov,
- Text chunking - zoskupovanie slov vo vete,
- obmedziť slová majúce viacero významov,
- Parsovanie – nájdenie vzťahov medzi slovami vo vete a určovanie funkcie slov vo vete prostredníctvom vetných členov.

Po znížení počtu slov v dokumentoch, Vector Space Model ponúka možnosť uchovať slová a slovné spojenia z dokumentov. Pozostáva z multidimenzionálnych vektorov, ktorých počet závisí od skúmaných dokumentov. Do každého vektora sú vkladané slová alebo slovné spojenia z jednotlivých dokumentov, ktoré prešli technikami na zníženie počtu slov. Každý riadok vo vektore môžeme nazývať element reprezentujúci unikátne slovo, slovné spojenie, vetu alebo odstavec z dokumentu. Vo Vector Space Modeli sú definované váhy pridelené k jednotlivým elementom vektora, ktoré hodnotia relevanciu elementov v modeli [16]. Príklad takéhoto modelu je zobrazený v Tab. 2.

Tab. 2 Príklad Vector Space model

	D1	D2	D3
cena		1	4
kvalita	2		3
rýchlosť	1	3	1
spoľahlivosť	2		
vek		1	

Stĺpce D1, D2 a D3 reprezentujú jednotlivé dokumenty. V riadkoch sú elementy (slová) extrahované z dokumentov. Čísla v tabuľke vyjadrujú váhu elementov v kontexte dokumentov. Váhy jednotlivých elementov je možné vyjadriť vlastnou stupnicou alebo rôznymi matematickými metódami ako entropia alebo euklidovská veta.

1.5.2 Metódy Text Miningu

Sú štatistické metódy, pomocou ktorých môžeme charakterizovať dokumenty na základe kľúčových slov, zlučovať dokumenty na základe spoločných znakov, dolovať informácie a šablóny z dokumentov a vizualizovať výsledky extrahovania informácií z dokumentov. Podľa typu úloh delíme Text Mining-ové metódy do 4 skupín:

- klasifikačné,
- zhlukovacie,
- extrakcia informácií,
- vizualizačné.

Do každej skupiny patrí viacero metód. V rámci tejto kapitoly vysvetlíme niekoľko vybraných metód z každej kategórie.

Klasifikácia

Metóda Naive Bayes klasifikátor je založená na princípe výpočtu pravdepodobnostní, pomocou ktorých sa klasifikujú webové dokumenty a elementy (slová, slovné spojenia, atď.) charakterizujúce skupiny. Predpokladá sa nezávislosť jednotlivých elementov medzi skupinami. „Vychádza z predpokladu, že efekt, ktorý má hodnota (každého) atribútu na danú triedu, nie je ovplyvnený hodnotami ostatných atribútov“ [17]. V rámci metódy dochádza k výpočtu posteriórnej pravdepodobnosti, ktorá charakterizuje pravdepodobnosť, kľúčové slová extrahované dokumentu majú spojitosť práve s jednou triedou. Posteriórnu pravdepodobnosť môžeme vypočítať podľa vzorca [16]:

$$p(L_c|t_1, \dots, t_n) = \frac{\mathrm{p}(t_{1,\dots,}t_n|L_c)*p(L_c)}{\sum_{L\in L} p(t_{1,\dots,}t_n|L)*p(L)}\,, \tag{1}$$

kde:

$\mathrm{p}(t_{1,} \dots, t_n|L_c)$ - pravdepodobnosť, že vybrané slovné spojenie bude zaradené do triedy L_c,

p(L) – pravdepodobnosť, že dokument patrí do triedy L_c bez ohľadu na obsah dokumentu.

Po vypočítaní posteriórnych pravdepodobností sme schopní na základe výsledkov klasifikovať jednotlivé dokumenty. Dokument bude zaradený do triedy, pre ktorú nadobúda najvyššiu postreriórnu pravdepodobnosť.

Rozhodovacie stromy generujú množinu pravidiel, podľa ktorých sú dokumenty klasifikované do skupín. Výsledkom metódy je strom, pozostávajúci z uzlov reprezentujúci skupinu a z hrán, charakterizujúcich slovné spojenia, ktoré definujú skupinu. Z testovacej množiny dokumentov sú vytvorené skupiny a graf. Následné dokumenty prechádzajú grafom cez jednotlivé uzly a klasifikujú sa do skupín. Existuje viacero algoritmov Rozhodovacích stromov. Jedným z najznámejších je C4.5, pri ktorom sa počíta Shannonova entropia [17]:

$$H = -\sum_{j=1}^{n} p(x_j) * log_2(p(x_j)) \ . \quad (2)$$

Pred identifikáciou kľúčových slovných spojení je entropia dokumentu vysoká. Je nutné vybrať tie slovné spojenia, ktoré znížia entropiu na minimum.

Zhlukovanie

Pri zhlukovaní dochádza k triedeniu dokumentov do zhlukov. Výsledkom metódy je množina zhlukov. V rámci každého zhluku by mali byť obsahovo podobné dokumenty na základe kľúčových slovných spojení, ale zhluky by mali byť rozdielne. Pri zhlukovaní môžeme využiť rôzne štatistické parametre a zhlukovacie algoritmy pre čo najpresnejší výsledok. Pomocou štatistických parametrov nie je možné zaradiť dokumenty do klastrov, ale ohodnotiť výsledky algoritmov. Môžu byť použité na ohodnotenie výsledkov generovaných jednotlivými algoritmami alebo na získanie informácií o klastroch.

Môžeme použiť tieto štatistické merania [16]:

- Metóda najmenších štvorcov – hodnotí kompaktnosť procesu zhlukovania. Nevýhodou je závislosť hodnoty od počtu klastrov.
- Silhouette koeficient – hodnotí kvalitu klastrov, či každý z klastrov je dostatočne špecifický a charakterizovaný. Koeficient nadobúda hodnoty z intervalu (0,1), pričom hodnoty od 0,7 vyššie vyjadrujú, že štruktúru klastrov je ľahké identifikovať a sú navzájom odlišné. Hodnoty 0,25 nižšie vyjadrujú slabú štruktúru klastrov a problém s pochopením, aké dokumenty sú zhromaždené v jednotlivých klastroch.

- Porovnávací koeficient – hodnotí homogénnosť/odlišnosť klastrov. „S každým výsledným klastrom P z množiny obsahujúcej všetky dokumenty sa pracuje ako s výsledkom dotazu.“ [16] Rovnako sa pracuje aj s množinami dokumentov v klastroch označovanými L. Najskôr je nutné vypočítať presnosť Pr klastra P a následne čistotu C toho istého klastra ako vážený priemer maximálnych hodnôt presností klastra [16]:

$$Pr\ (P, L) = \frac{|P \cap \mathrm{L}|}{|\mathrm{P}|}\ , \tag{3}$$

$$C\ (P, L) = \sum_{\mathrm{P} \in \mathrm{P}\ |} \frac{|\mathrm{P}|}{|\mathrm{D}|}\ max_{L \in L}\ \mathrm{Pr}(\mathrm{P}, \mathrm{L}). \tag{4}$$

Čistota nadobúda hodnoty od 0 po 1. Číslo 1 vyjadruje dobre definované klastre, do ktorých boli rozdelené dokumenty.

Algoritmus k priemerov (k means) patrí medzi najznámejšie algoritmy zhlukovania. Prostredníctvom tohto algoritmu zhlukujeme dokumenty do k zhlukov, pričom vzdialenosť dokumentov a centrom zhluku je minimalizovaná. Algoritmus je využiteľný aj pri veľkých počtoch dokumentov obsahujúcich množstvo dát a prebieha v týchto štyroch krokoch:

1. Určíme počet zhlukov a náhodne vyberieme dokumenty, ktoré budú reprezentovať centrá zhlukov.
2. Vypočítajú sa vzdialenosti (podobnosti) medzi dokumentami a centrami zhlukov. Podľa najkratšej vzdialenosti zaradíme dokumenty do zhlukov.
3. Definujú sa nové centrá a opäť sa vypočítajú vzdialenosti medzi dokumentami a novými centrami.
4. Algoritmus končíme, keď získame optimálne centrá zhlukov a nebude existovať lepšia možnosť.

Na výpočet vzdialenosti medzi dokumentami a centrami sa využívajú napríklad Euklidovská, Hammingova alebo Minkowskeho vzdialenosť. Algoritmus k priemerov patrí do nehierarchického zhlukovania, keď dokumenty nevytvárajú stromovú štruktúru, ale sú postupne rozkladané do zhlukov. Naopak hierarchické zhlukové algoritmy ako „Algoritmus najbližšieho suseda“ alebo „Pravidlo priemernej vzdialenosti“, vytvárajú stromovú štruktúru a hľadajú podobnosti medzi zhlukmi. Rozlišujeme dva typy princípov: aglomeratívny a divízny. Pri aglomeratívnom každý dokument tvorí jeden zhluk. Postupne dva najbližšie

vzdialené (najviac podobné) zhluky spojíme do jedného. Algoritmus končí, keď sa všetky zhluky spoja do jedného. Pri divíznom princípe sú na začiatku všetky dokumenty v jednom zhluku a postupne dochádza k tvorbe nových zhlukoch. Najskôr oddeľujeme dokumenty, ktoré sú najviac vzdialené (najodlišnejšie). Hierarchické algoritmy sú často používané na začiatku procesu zhlukovania a na získanie koncového riešenia sa použije nehierarchický algoritmus.

Extrakcia informácií

Dokumenty často obsahujú veľa textov, ktoré nie je vhodné ihneď analyzovať. Extrakciou informácií je možné z týchto textov extrahovať časti, v podobe slov, slovných spojení alebo fráz. Extrakcia prebieha na základe sémantiky jazyka, kedy sme schopní definovať presné tvary extrahovaných informácií. Tento typ extrakcie si môžeme ukázať na konkrétnom príklade. Uvažujme o extrakcii informácií z tejto vety: Andrej Kiska ako občiansky kandidát bol zvolený za prezidenta 29. marca 2014. Z tejto vety môžeme extrahovať tieto informácie: meno a priezvisko (Andrej Kiska), pozícia vo voľbách (občiansky kandidát), nová pozícia (prezident), dátum zvolenia (29. marca 2014). Na rozpoznávanie slov v textoch je možné použiť skryté markove modely, ktoré patria v tejto oblasti medzi najúspešnejšie. Sú schopné zachytiť rôznorodosť slov vo vetách. Model pozostáva z troch komponentov: množina stavov, pravdepodobnosťami medzi nimi a pravdepodobnosťami generovania výstupných symbolov. Na začiatku je definovaný počiatočný stav (začiatočné slovo vety). Pri každom prechode z jedného slova na druhé je definovaný znak. Tento znak je náhodný a určený svojou funkciou hustoty pravdepodobnosti. Pravdepodobnosti medzi jednotlivými slovami určujú, ako sa model dostane z jedného slova na druhé [18].

Vizualizácia

Grafická prezentácia umožňuje lepšie pochopiť výsledky Text Mining-ových techník a metód. Výsledky týchto techník v tabuľkách alebo v dokumentoch sú častokrát nezrozumiteľné, najmä pre ľudí bez štatistických vedomostí. Výsledné vzťahy a výsledky (najmä štatistické) je možné zobraziť napríklad v trojdimenzionálnej reprezentácii kategórií, ktoré sú prehľadné a je v nich ľahké vyhľadať dáta podľa definovaných kritérií. Na vizualizáciu dokumentov alebo skupín dokumentov sa využíva dvojrozmerný priestor, v ktorom sú dokumenty alebo skupiny zobrazené farbami na odlíšenie a zvýraznenie rôznych typov. Na identifikáciu dokumentov alebo skupín sa používajú rôzne grafické prvky

ako textové vlajky a bubliny, v ktorých sú zobrazené kľúčové slová, charakterizujúce dokument alebo skupinu. Self-Organizing mapy ponúkajú možnosť triediť dokumenty do skupín a zobraziť podobnosti a odlišnosti medzi jednotlivými skupinami. Na mape sú jednotlivé skupiny zobrazené v podobe buniek, ktoré majú definovanú farbu na odlíšenie od iných buniek. Self-Organizing mapy rovnako ako zhlukovanie zhlukujú dokumenty do skupín, avšak zhlukovanie nezachytáva vzťahy medzi jednotlivými zhlukmi [16].

1.6 Dátové modely a databázy

Dáta, ktoré chceme uchovávať a neskôr dolovať, je nutné ukladať v štruktúrovanej a organizovanej podobe. Dátové modely definujú vzťahy medzi dátovými prvkami v modeli a definujú tiež formát a štruktúru dát. Na základe spôsobu uchovávania dát a vzťahov medzi dátami rozlišujeme tri typy dátových modelov: sieťový, hierarchický a entitno-relačný. V *hierarchickom modeli* sú dáta ukladané do stromovej štruktúry. Vzťahy medzi dátami v tomto modeli sú vyjadrené väzbou rodič-potomok. Hlavná dátová štruktúra (tabuľka) vystupuje v modeli ako koreň, na ktorý sa napájajú ostatné dátové štruktúry označované ako uzly. Uzly sú umiestňované na rôznych úrovniach a sú pospájané prostredníctvom vetiev. V *sieťovom modeli* sú dáta usporiadané do entít, medzi ktorými sú vzťahy 1:1,1:N alebo N:M. Súčasťou modelu je aj graf, v ktorom entity sú zobrazované ako uzly a hrany medzi uzlami predstavujú vzťahy medzi reláciami. *Entitno-relačný model* je z týchto modelov najmladší, avšak v oblasti podnikových dát najpoužívanejší. V modeli sa pracuje s dvomi základnými komponentami: tabuľka a relácia. Po dokončení definovania tabuliek a vzťahov medzi nimi je model pripravený na nasadenie do systému. Model, ktorý je nasadený v systéme, sú v ňom uložené dáta a vykonávajú sa na ňom databázové operácie (napríklad procedúry, funkcie, atď.), sa nazýva databáza. Existuje viacero názorov čo je databáza a ktoré objekty sú súčasťou databáz, a ktoré do databáz nepatria. My definujeme databázu ako množinu obsahujúcu štruktúrované dáta, ktoré reprezentujú fakty reálneho sveta. Databázy uchovávajú dáta a je možné ich kedykoľvek získať prostredníctvom dopytovacieho jazyka (najčastejšie SQL). Dáta sú ukladané do tabuliek, ktoré sú dvojrozmerného priestoru. Skladajú sa zo stĺpcov a riadkov, pričom v tabuľke neexistujú dva riadky, ktoré by mali rovnaké hodnoty vo všetkých stĺpcoch.

V databáze existujú tri typy vzťahov medzi tabuľkami [19]:

- 1:1 – vyjadrenie vzťahu medzi dvomi tabuľkami, kedy jeden záznam má vzťah práve s jediným jedinečným záznamom z druhej tabuľky.

- 1:N - väzba v rámci ktorej jedinečný záznam z prvej tabuľky má väzbu s jedným alebo viacerými záznamami z druhej tabuľky.
- N:M –väzba pri ktorej, každému záznamu z prvej tabuľky môžeme priradiť viacero záznamov z druhej tabuľky a tieto záznamy z druhej tabuľky majú ľubovoľný počet väzieb so záznamami z prvej tabuľky.

Vzťah 1:N je medzi tabuľkami najpoužívanejší. Návrhári databáz sa snažia vyhnúť vzťahu N:M medzi tabuľkami. Preferuje sa rozbitie tohto vzťahu spôsobom, že medzi tabuľky sa použije tretia tabuľka, ktorá bude mať s týmito dvomi tabuľkami vzťah 1:N. Pomocou tejto tabuľky je možné prepojiť dáta z týchto dvoch tabuliek.

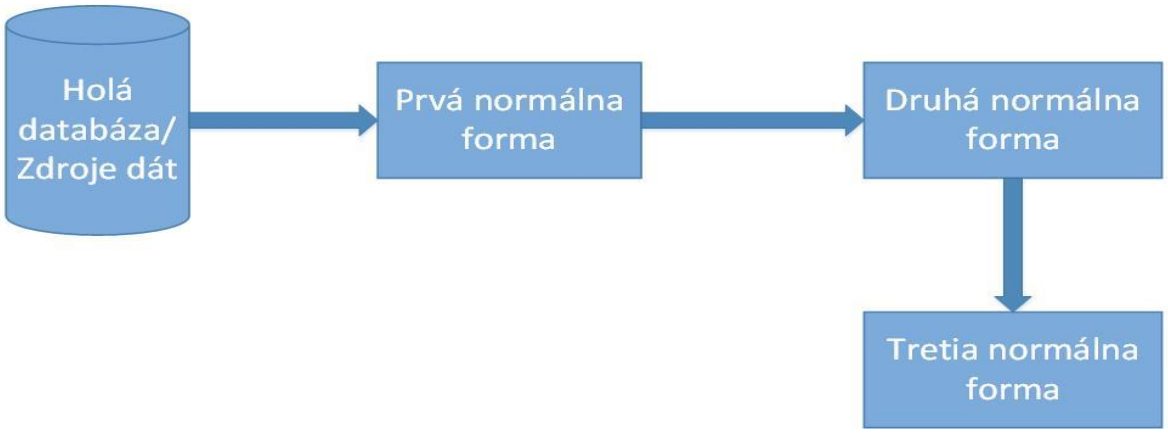
Návrh modelu a štruktúra dát v ňom môže byť definovaná tromi základnými schémami: 3.normálna forma, hviezdicová schéma a snehová vločka.

1.6.1 Normalizácia

Jedným z najdôležitejších predpokladov správneho fungovania databáz je neopakovateľnosť dát v tabuľkách a odstránenie redundancie dát v tabuľkách v rámci jedného modelu alebo databázy. Redundancia je označenie pre stav databázy, v ktorej sa nachádza veľa zbytočných údajov. Za zbytočné môžeme považovať dáta, ktoré sú duplicitné alebo nevyužívané. Nadbytočné údaje zaberajú len veľa miesta, ale aj spôsobujú chaos v rámci databáz. Úlohou procesu normalizácie je znížiť tieto riziká a navrhnúť tabuľky v databáze tak, aby boli ľahko použiteľné a poskytovali správe dáta pre rôzne typy užívateľov ako administrátori, programátori alebo analytici.

Normalizáciu môžeme označiť za proces, skladajúci sa z troch hlavných činností. Tieto činnosti nie je možné vykonávať naraz, nakoľko bez výsledkov jednej činnosti nemôžeme začať nasledujúcu. Po každej vykonanej aktivite sú tabuľky v určitom tvare. Tieto činnosti sa nazývajú prvá normálna forma, druhá normálna forma a tretia normálna forma. Avšak ešte pred začatím prvej normálnej formy je dôležité si zozbierať požiadavky od budúcich užívateľov alebo zadávateľov požiadavky. Po dokončení analyzovania požiadaviek, sa zvyčajne dostávame k holej databáze alebo k zdrojom dát. Holá databáza sa skladá z tabuliek obsahujúcich veľké množstvo stĺpcov a opakujúcich sa dát vo viacerých tabuľkách. Pri holých databázach je často požiadavka na migráciu dát z jednej implementácie do druhej [20]. Zdrojmi dát myslíme dáta nachádzajúce sa na webových stránkach alebo tabuľkových editorov (Excel, Access, atď.). Pri týchto zdrojoch je nutné, na

začiatku definovať dáta, aké chceme uchovávať. Definujeme tabuľky, ktoré v tejto fáze môžu obsahovať veľa stĺpcov. Proces normalizácie je zobrazený na Obr. 2.



Obr. 2 Proces normalizácia

V rámci prvej normalizačnej formy dochádza k prerozdeľovaniu dát do tabuliek. Po vytvorení tabuliek je v každej definovaný primárny kľúč. Primárny kľúč definuje stĺpec alebo kombináciu stĺpcov, ktorých hodnoty sú v tabuľke jedinečné. V tejto fáze je nutné, aby každý stĺpec bol priradený k definovanému primárnemu kľúču. Dochádza k zníženiu počtu stĺpcov v tabuľkách, ktoré sú rozdelené do viacerých menších tabuliek. V druhej normalizačnej forme dochádza opäť k zvýšeniu počtu tabuliek, nakoľko stĺpce, závislé len na časti primárneho kľúča, sú rozdelené do nových tabuliek. V poslednej normalizačnej forme odstránime z tabuliek stĺpce, ktoré nie sú závislé na primárnom kľúči a miesto týchto stĺpcov budú použité cudzie kľúče v týchto tabuľkách. V novovzniknutých tabuľkách budú tieto stĺpce primárnymi kľúčmi [20].

1.6.2 Výhody a nevýhody normalizácie

Jednou z najväčších nevýhod relačných databáz po normalizácii je nižší výkon databázy [20]. Táto nevýhoda komplikuje situáciu pri selektovaní dát z databáz. Normalizované databázy spotrebujú viac pamäti a procesorového času počas databázových dotazov a transakcií. Je to spôsobené najmä tým, že pri normalizácii dochádza k rozbitiu veľkých tabuliek na malé. Tento prístup poskytuje veľa výhod uvedených v tejto kapitole, avšak pri práci s dátami musí databázový systém prehľadávať viac tabuliek a následne ich spojiť, čo má za následok dlhší transakčný čas.

Normalizácia poskytuje tieto najdôležitejšie výhody [20]:

- lepšie usporiadanie databáz – šetrí čas prístupu k tabuľkám a pomáha lepšie pochopiť vzťahy medzi dátami všetkým užívateľom databázy na rôznych úrovniach pozícii.
- menší počet nadbytočných údajov (nižšia úroveň redundancie) – šetrí pamäť na diskoch a zjednodušuje dátové štruktúry.
- konzistencia dát v tabuľkách – znižuje riziko, že nájdeme rovnaké dáta v dvoch rozdielnych tabuľkách v rôznych formátoch.
- flexibilnejší návrh databázy – poskytuje ľahšiu modifikáciu dát v tabuľkách.
- lepšia možnosť zabezpečenia databázy – umožňuje efektívnu a ľahšiu prácu pre administrátora databázy, najmä v oblasti prístupových opatrení.

Normalizácia odstraňuje tieto tri anomálie objavujúce sa pri denormalizovaných databázach [21]:

- Anomália vkladania – nie je možné uložiť do databázy nové dáta, nakoľko existuje závislosť medzi stĺpcami.

Tab. 3 Anomália vkladania

id_z	Meno	Priezvisko	Pozícia	Platové rozmedzie na pozícii
1	Roman	Malý	analytik	1100-2100
2	Jana	Horváthová	učtovník	800-1400
3	Peter	Chorý	analytik	1100-2100
			programátor	

V Tab. 3 je zobrazená situácia, kedy firma otvorí novú pracovnú pozíciu programátora, ale zatiaľ táto pozícia nie je obsadená. Je nutné, aby sa v databáze uložila informácia, že takáto pozícia existuje vo firme, avšak v tejto situácii to nie je možné, nakoľko stĺpec Pozícia je závislý od ostatných stĺpcov a na pozícii zatiaľ nikto nepracuje.

- Anomália mazania – po odstránení záznamov stratíme detailnejšie informácie.

Tab. 4 Anomália mazania

id_z	Meno	Priezvisko	Pozícia	Platové rozmedzie na pozícii
1	Roman	Malý	analytik	1100-2100
~~2~~	~~Jana~~	~~Horváthová~~	~~učtovník~~	~~800-1400~~
3	Peter	Chorý	analytik	1100-2100

V Tab. 4 sa rozhodneme vymazať druhý záznam, nakoľko zamestnankyňa Jana Horváthová už nepracuje v spoločnosti. Bola jedinou pracovníčkou na pozícii účtovník a po vymazaní druhého riadka z tabuľky, stratíme informáciu o platovom rozmedzí na tejto pozícii.

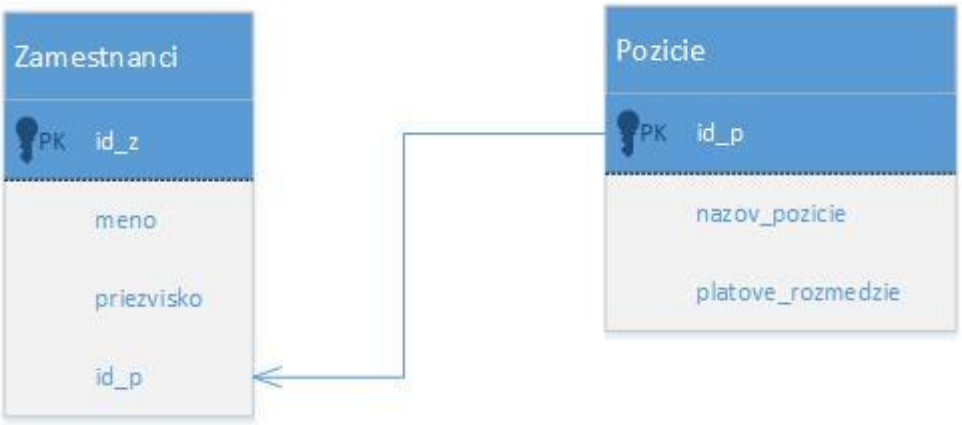
- Anomália modifikácie – aktualizácia jedného typu záznamu spôsobí aktualizáciu na viacerých miestach v databáze.

Tab. 5 Anomália vkladania

id_z	Meno	Priezvisko	Pozícia	Platové rozmedzie na pozícii
1	Roman	Malý	analytik	900-1800
2	Jana	Horváthová	učtovník	800-1400
3	Peter	Chorý	analytik	1100-2100

Po zmene platového rozmedzia na pozícii analytik je nutné aktualizovať každý údaj v tomto stĺpci, v rámci ktorého zamestnanec pracuje na tejto pozícii.

Riešením tohto problému by bolo rozbiť tabuľku na dve tabuľky, medzi ktorými by existoval vzťah.

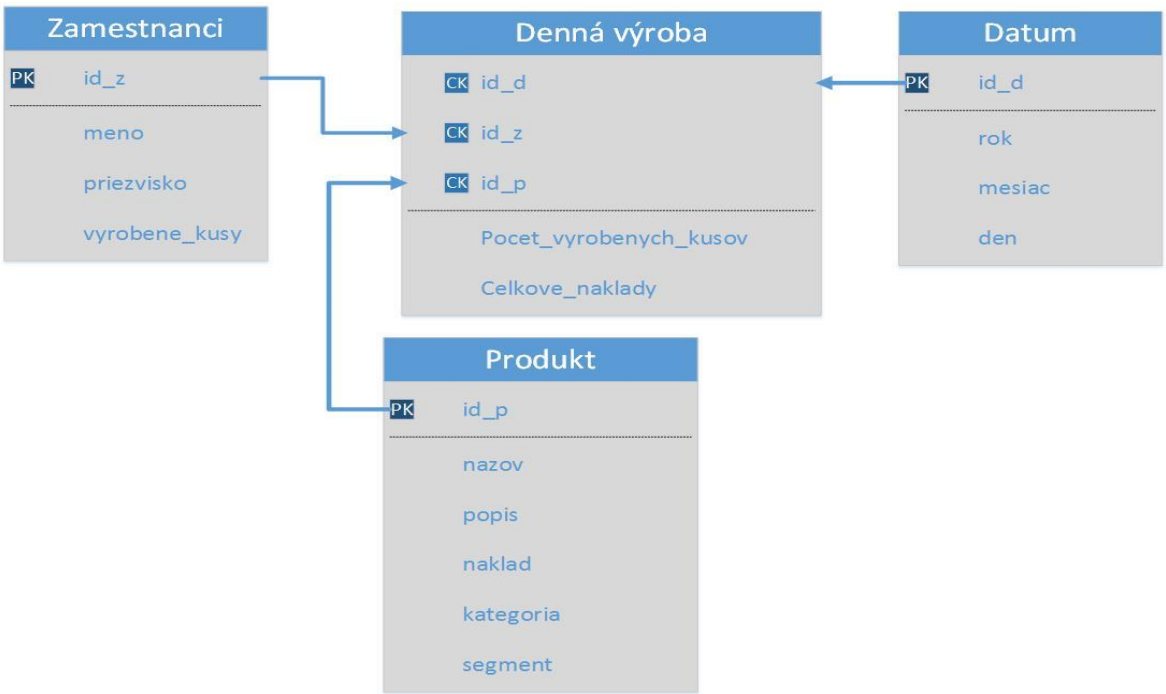


Obr. 3 Väzba dvoch tabuliek

Tabuľka zamestnanec obsahuje stĺpce identifikačný stĺpec zamestnanca, meno a priezvisko zamestnanca a identifikačné číslo zamestnanca, ktoré je v tejto tabuľke ako cudzí kľúč. Táto tabuľka má vzťah s tabuľkou Pozícia N:1, v ktorej sú informácie o identifikačnom čísle pozície, názvu pozície a platovom rozmedzí. Identifikačné číslo pozície (id_p) je v tejto tabuľke primárnymi kľúčom.

1.6.3 *Hviezdicová schéma*

Dátový model podľa tejto schémy je v tvare hviezdice, pozostávajúci z centrálnej tabuľky faktov a dimenzii. V dátovom modeli je tabuľka faktov zobrazená v strede a tabuľky dimenzii okolo. Je to spôsobené kvôli vzťahom, ktoré existujú len medzi tabuľkou faktov a jednotlivými dimenziami, nie medzi dimenziami navzájom. Tabuľka faktov obsahuje cudzie kľúče odkazujúce na identifikátory (primárne kľúče) v dimenziách a metriky zvyčajne číselné, ktoré vyjadrujú určitý štatistický ukazovateľ v podniku a počítajú sa na základe hodnôt v dimenziách. Dimenzie nie sú normalizované, čo má za následok, že často obsahujú rovnaké dáta na viacerých riadkov. Na Obr. 4 je zobrazená hviezdicová schéma jednoduchého dátového modelu [27]. Dátový model pozostáva z jednej tabuľky faktov: Denná výroba a troch dimenzii: Zamestnanci, Datum a Produkt. Každý riadok v tabuľke faktov má tri cudzie kľúče pridružené ku jednotlivým dimenziám. V dimenzii Zamestnanci je zoznam zamestnancov, ktorí vyrábajú produkty. V dimenzii Produkt je zoznam vyrábaných produktov a v poslednej dimenzii Datum sú roky, mesiace a dni, kedy prebiehala výroba produktov. V tabuľke faktov Denná výroba sú okrem cudzích kľúčov aj dve metriky: Počet vyrobených kusov a Celkový počet vyrobených kusov.



Obr. 4 Príklad hviezdicovej schémy

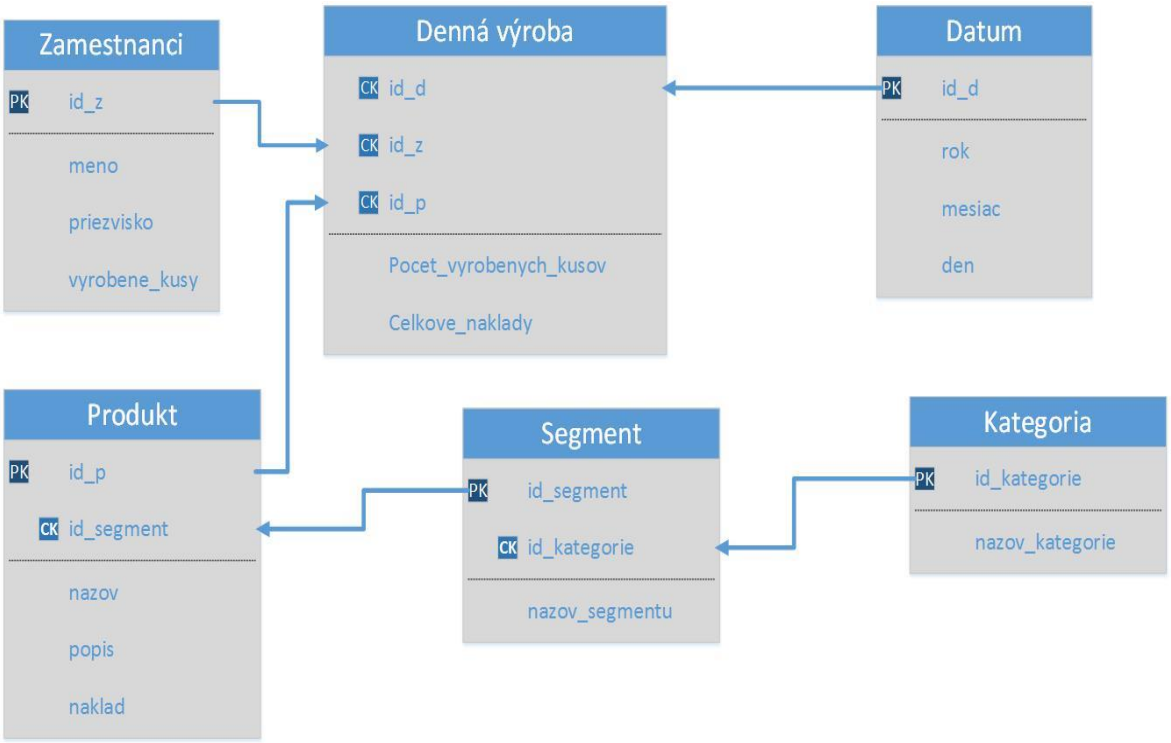
Granularita je na dennej úrovni pre časovú dimenziu, na úrovni jednotlivých zamestnancov v prípade zamestnaneckej dimenzie a na úrovni jednotlivých produktov

z pohľadu produktovej dimenzie. Nastavená granularita je na najnižšej možnej úrovni vo všetkých dimenziách, čím metriky vyjadrujú:

- koľko a aké typy produktov vyrobia jednotliví zamestnanci za vybraný deň, mesiac, rok alebo celé výrobné obdobie,
- koľko a aké typy produktov sa vyrobia celkovo za jeden deň, mesiac, rok alebo za celé výrobné obdobie,
- koľko produktov vyrobia jednotliví zamestnanci za jeden deň, mesiac, rok alebo za celé výrobné obdobie,
- koľko produktov sa vyrobí celkovo za jeden deň, mesiac, rok alebo za celé výrobné obdobie.

1.6.4 Snehová vločka

Schéma dátového modelu nazývaná snehová vločka je variantnou hviezdicovej schémy. Schémy sa líšia štruktúrou pri tabuľkách dimenzií, nakoľko každá dimenzia má nielen relačný vzťah s tabuľkou faktov, ale aj ďalšími tabuľkami, ktoré sú v tretej normálnej forme. Týmito tabuľkami je zabezpečený menší výskyt duplicitných dát a menší nárast veľkostí dát v dátovom modeli. Na Obr. 5 je dátový model z predošlej kapitoly, obsahujúci rovnaký počet dimenzií, no rozšírený o dve tabuľky: Segment a Kategoria.



Obr. 5 Príklad schémy snehovej vločky

Model rozšírený o tieto dve tabuľky, čím ponúka ďalšie možnosti pre tvorbu zostáv. Granulita v tomto modeli je na vyššej úrovni, nakoľko môžeme dostať ďalšie zaujímavé dáta, napríklad z akého segmentu najviac vyrábame produkty alebo z akej sú kategórie.

1.6.5 NOSQL prístup

NOSQL (Not only SQL) predstavuje prístup na ukladanie dát s nerelačným charakterom v distribuovaných prostrediach. Tento prístup nenahradzuje relačné databázy, ale zahŕňa postupy a technológie, ktoré nie sú postavené na relačnom modeli dát. NOSQL prístupy začali vznikať na začiatku 21. storočia, za účelom zachytávania veľkého množstva dát vo webových aplikáciách a v cloudoch [22]. Tieto technológie sú schopné pracovať so všetkými typmi dát (uvedené v kapitole 1.1). Úlohou týchto prístupov je uchovávať a načítať veľké množstvo dát. Nové webové aplikácie alebo systémy vyžadujú od dátových modelov dynamickosť, rýchlu odozvu a najmä dáta v reálnom čase.

Pri veľkých množstvách dát tieto potreby klasické relačné databázy nevedia splniť, a preto NOSQL prístupy sa stávajú dôležitou súčasťou dátovej architektúry pre firmy pracujúce s veľkým množstvom dát [22]:

- TESCO – použite NOSQL pre ecommerce alebo produktové katalógy
- Ryanair – na zvýšenie rýchlosti odozvy mobilnej aplikácie používanú viac ako tromi miliónmi používateľov
- Marriott - pre rezervačný systém, v ktorom sa účtuje 38 bilión dolárov ročne

NOSQL technológie sú vysoko výkonné, avšak neposkytujú tak veľa funkcií a operácií na prácu s dátami, ako klasické systémy založené na relačnom modeli dát.

2 Cieľ práce

V rámci tejto kapitoly definujeme ciele diplomovej práce, postupy a metódy, ktoré sme použili pri návrhu a aplikácii modelu.

Hlavným cieľom záverečnej práce je návrh modelu, pomocou ktorého dokážeme extrahovať rôzne typy dát z webových stránok a transformovať ich do podoby, kedy budú mať informačnú hodnotu. Následne tento model aplikovať v situácii, kedy chceme zistiť skúsenosti a názory užívateľov na internetové obchody.

Čiastkovými cieľmi predkladanej diplomovej práce v oblasti návrh modelu sú:

- Navrhnúť model, skladajúci sa z procesov, pomocou ktorých sa dostaneme ku kompletnému finálnemu riešeniu.
- Detailne definovať tieto procesy, v rámci ktorých budeme využívať rôzne technológie a metódy.
- Každá technológia a metóda je využívaná na špecifické činnosti, preto použite týchto technológii a metód bude vysvetlené na ukážkových príkladoch.

V oblasti aplikácie modelu máme stanovené tieto čiastkové ciele:

- Identifikovať a analyzovať webové zdroje obsahujúce dáta, ktoré potrebujeme na riešenie nášho problému.
- Extrahovať štruktúrované, neštruktúrované a semištruktúrované dáta z vybraných webových zdrojov.
- Transformovať tieto typy dát do vhodnej podoby pre pamäťové média tak, aby poskytovali cenné informácie.
- Uložiť transformované dáta do relačných databáz vo vhodnej forme pre budúce analyzovanie.

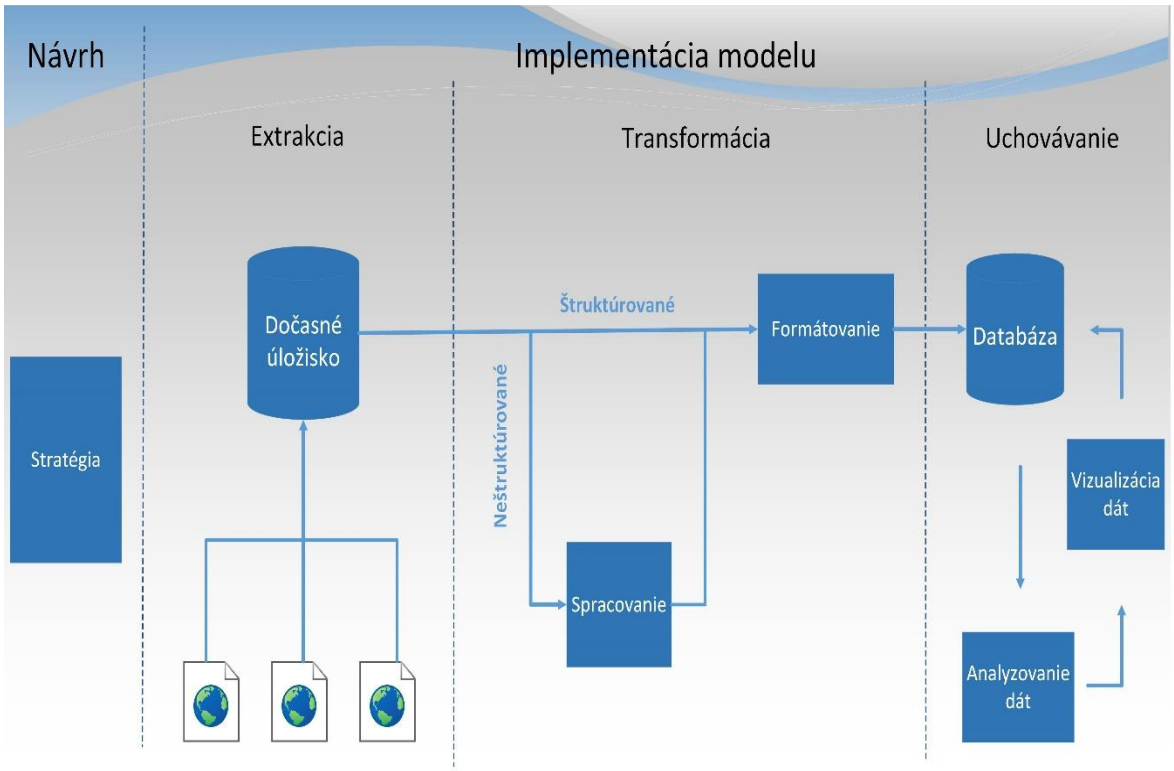
Pri návrhu a aplikovaní modelu sme čerpali zo skúseností a poznatkov, ktoré autor diplomovej práce nadobudol počas štúdia na Ekonomickej univerzite (Fakulta hospodárskej informatiky), semestrálneho študijného pobytu na Åbo Akademi (Fínsko), pracovných skúseností v tomto odbore a z príslušnej odbornej literatúry a internetových zdrojov.

Nástroje a technológie spomenuté a použité v ďalších dvoch kapitolách nemusia patriť medzi najlepšie na trhu, avšak na základe našich skúseností, vedomostí a vedomia ich používame v rámci nášho modelu na implementovanie funkcionalít a aktivít v modeli.

3 Návrh modelu

V rámci tejto kapitoly predstavíme náš navrhnutý model a popíšeme jednotlivé časti modelu.

Model pozostáva z fázy Návrhu a Implementácie modelu, ktorá pozostáva z troch častí: Extrakcia, Transformácia a Uchovávanie. Model je zobrazený na Obr. 6. Na tomto základnom obrázku nie sú zobrazené všetky procesy, aktivity a technológie, ktoré môžeme použiť v procesoch extrakcie, transformácie a analyzovania štruktúrovaných a neštruktúrovaných dát, ale obrázok slúži na predstavu fungovania nášho navrhovaného modelu.



Obr. 6 Model na extrakciu, transformáciu a analýzu dát

Prvá fáza modelu zahŕňa činnosti súvisiace s identifikáciou úloh, ktoré model bude vykonať a návrhov funkcionalít. Na konci tejto fázy by sme mali mať k dispozícii detailne popísané tri časti druhej fázy modelu. V časti Extrakcia získavame rôzne typy dátových súborov, webových dokumentov alebo dát a ukladáme ich do dočasného úložiska. V ďalšej časti najskôr pracujeme len s neštruktúrovanými dátami, ktoré pomocou Text Mining-ových metód transformujeme do štruktúrovanej podoby, vhodnej pre získavanie znalostí z dát a uchovávanie v relačných databázach. Keď máme všetky dáta v štruktúrovanej podobe

môžeme začať formátovať dáta, aby ich bolo možné ukladať do databázy. V poslednej časti po prenose dát do databázy analyzujeme naše dáta pomocou základných štatistických metód alebo Data Miningov-ových metód. Výsledky metód pre lepšie pochopenie vizualizujeme a ukladáme do databázy, aby sme ich mali k dispozícii a nemuseli aplikovať metódu znova na rovnaké dáta.

Na základe informácií v predchádzajúcom odseku a v ďalších častiach tejto kapitoly, v ktorých detailnejšie popíšeme jednotlivé časti definujeme základné ciele fázy Návrhu a jednotlivých častí Implementačnej fázy:

- Návrh – detailný popis funkcionalít jednotlivých častí vhodných pre dáta a definované ciele.
- Extrakcia – uložené dáta a súbory z rôznych dátových zdrojov v našom úložisku.
- Transformácia – štruktúrované dáta v jednotnom formáte vhodnom pre nami navrhnutú databázu a jej objekty.
- Uchovávanie – uskladnenie dát a výsledkov analýz, a vizualizácia výsledkov.

V ďalších častiach sa venujeme detailnejšie fázam modelu a možnostiam, aké metódy a technológie je vhodné použiť.

3.1 Návrh

Prvá časť modelu pozostáva z činností, ktorými by sme mali zistiť aké funkcie by mal mať náš model a či sme schopní model s takýmito funkciami implementovať v reálnom prostredí s reálnymi dátami. Táto fáza pozostáva z týchto 4 krokoch:

- identifikácia a definovanie cieľov,
- identifikácia a analýza dátových zdrojov,
- návrh procesov a technológii,
- návrh dátového modelu a implementácia databázy.

Pred návrhmi, ako realizovať model, je nutné venovať čas identifikácii cieľov. Pri identifikovaní cieľov berieme do úvahy celý rad faktorov, ako sú finančné, technologické alebo znalostné možnosti. Pri finančných musíme vedieť, aký rozpočet máme pre náš model. Technologické možnosti nám hovoria o tom, aké technológie máme k dispozícii pre model a znalostné, či sme schopní tieto technológie využiť a aké metódy a algoritmy budeme môcť

použiť v rámci nášho modelu. Po zistení našich možností by sme mali nájsť odpovede na otázky:

- Prečo sa má model použiť?
- Ako sa bude s modelom pracovať?
- Bude model spĺňať požiadavky užívateľov?

Následne sme schopní identifikovať naše ciele, pre ktoré model chceme aplikovať. Tieto ciele definujeme z viacerých pohľadov v závislosti od typov užívateľov, ktorí budú dáta z databázy a znalosti (výsledky analýzy v poslednej fáze modelu) využívať. Napríklad je možné, že dáta z modelu budú využívané zamestnancami z viacerých oddelení. Každé oddelenie môže mať inú požiadavku na model, a preto je dôležité pri tvorbe modelu brať do úvahy rôzne požiadavky.

V ďalšej fáze najskôr identifikujeme potenciálne dátové zdroje. Dáta, ktoré chceme analyzovať, môžu pochádzať z viacerých zdrojov a môžu byť dostupné v rôznej podobe (kapitoly 1.2,1.3 a 1.4). Dáta môžu pochádzať priamo z webových sídiel alebo súborov (napríklad textový, pdf, tabuľkový alebo databázový), ktoré sú voľne dostupné na webových sídlach. V mnohých prípadoch sa stáva, že na webe nájdeme súbor obsahujúci neúplné dáta, alebo prístup k dátam môžeme získať len so súhlasom majiteľa webového sídla alebo spoločnosti, ktorá vlastní dané webové sídlo. V prípade záujmu o tieto dáta je nevyhnutné kontaktovať majiteľa a dohodnúť sa s ním na poskytnutí dát. V tejto fáze by sme mali brať do úvahy aj dôveryhodnosť dát na webe. Vyberať by sa mali len tie dátové zdroje, ktoré poskytujú pravdivé a overené informácie. Pri identifikácii zdrojov je vhodné použiť metadáta, pokiaľ sú súčasťou dátového zdroja. Metadáta nám poskytujú informácie o dátovom zdroji a o dátach v nich. Použitím metadát, získame:

- lepšiu predstavu o dátach,
- výhodu pri rozhodovaní, ktoré dátové zdroje použiť,
- informácie využiteľné pri ďalších procesoch v modeli.

Po identifikovaní zdrojov a vybratí tých, z ktorých budeme extrahovať dáta, je dôležité analyzovať tieto webové sídla a súbory. Začíname analyzovaním jednotlivých zdrojov z pohľadu štruktúry a následne sa zameriavame na identifikovanie typov a formátov dát. Pri webových stránkach analyzujeme štruktúru zdrojového kódu stránky a dáta z pohľadu pozície v zdrojovom kóde, a zobrazenia na stránke. Pokiaľ spoločnosť dodržiava

zásady a normy pre weby stanovené organizáciou W3C, sme schopní na základe tágov vo webovom dokumente určiť vzťahy medzi dátami a typy dát. Pokiaľ sa dáta nachádzajú v databázovom súbore, je nutné analyzovať dátový model databázy, prostredníctvom ktorého získame prehľad o databáze, vzťahoch medzi dátami a typmi dát. V textových súborov sa vyskytujú najmä neštruktúrované dáta, preto dochádza k analyzovaniu dokumentu pre extrakciu dát, avšak nie je možné určovať vzťahy medzi dátami. Analyzovaním zdrojov získame prehľad a informácie o dátach užitočných pre extrakciu a analýzu dát v ďalších fázach modelu.

Získaním prehľadu o dátach, s ktorými budeme pracovať v modeli sme schopní začať plánovať aké technológie, metódy, techniky a doplňujúce procesy pomôžu extrahovať, transformovať a ukladať dáta. Na Obr. 6 sú zobrazené v druhej, tretej a štvrtej časti základné procesy, ktoré budú v ďalších častiach doplnené o ďalšie aktivity. Po dokončení týchto fázy by sme mali mať k dispozícii zoznam cieľov modelu a návrh, ako budeme model realizovať. Mali by sme byť schopní vysvetliť ako bude model fungovať a pre koho bude model využiteľný.

Navrhovať štruktúru relačnej databázy by sme mali vo fáze, keď máme prehľad, s akými dátami budeme v modeli pracovať a aké dáta plánujeme ukladať do databázy. Najskôr je nutné urobiť rozhodnutie, v akom type modelu budeme ukladať naše dáta. Typom dátového modelu myslíme, v akej dátovej štruktúre budú tabuľky rozmiestnené v modeli. Na výber máme z troch základných štruktúr:

- hviezdicová,
- snehová vločka,
- 3.normálna forma.

Každá z týchto štruktúr má svoje klady a zápory. Výber vhodnej štruktúry by mal závisieť najmä od konkrétneho použitia modelu. Dátový model môže byť navrhovaný a používaný pre dva základné účely: dolovanie dát a multidimenzionálnu analýzu [27].

Pre účely multidimenzionálnej analýzy využívame schémy: hviezdicovú alebo snehovú vločku. Týmito postupmi sledujeme konkrétne dáta, ktoré sú využívané v analýzach, ktorými sledujeme napríklad vývoj predaja výrobkov alebo využívania služieb. Pri dolovaní dát hľadáme znalosti v dátach, ktoré by nám pomohli pri rozhodovaní alebo hľadáme závislosti a šablóny medzi dátami. V týchto prípadoch je vhodné mať dátový model v tretej normálnej forme, ktorá je schopná zachytiť vzťahy medzi dátami a môžeme spájať dáta z rôznych oblastí.

Po rozhodnutí, aký typ dátového modelu budeme využívať, začíname navrhovať atribúty, ktoré budú zhromažďovať rovnaké typy dát. Po definícii atribútov začíname s tvorbou dátového modelu, ktorý bude pozostávať z tabuliek a vzťahov medzi nimi. Stĺpce tabuľky budú tvorené z jednotlivých atribútov, primárnych a cudzích kľúčov.

Na základe dátového modelu môžeme začať s tvorbou databázy, do ktorej budeme vkladať štruktúrované dáta, ktoré nám budú k dispozícii pre analýzy. Pred implementáciou dátového modelu sa rozhodujeme, v akom databázovom systéme bude fyzicky uložený náš model a dáta. Pri rozhodovaní by sme mali brať do úvahy:

- Finančné náklady spojené s nákupom a údržbou databázového systému – niektoré systémy sú spoplatnené pre komerčné použitie, iné sú voľne dostupné pre súkromné aj pre komerčné použitie.
- Naše znalosti a skúsenosti s databázovými systémami – každý systém je špecifický svojou syntaxou a doplnkovými nástrojmi.
- Kompatibilita s ďalšími aplikáciami a systémami využívanými v modeli.

Na základe rebríčka zostaveného organizáciou DB-ENGINES sú tri najpoužívanejšie databázové systémy: Oracle Database, MySQL a Microsoft SQL Server. Systém MySQL je jediným z týchto nástrojov voľne dostupný na Internete. Zvyčajne sa využíva na prácu s webovými technológiami. Databázový systém od firmy Microsoft sa vyznačuje jednoduchou integráciou s ostatnými nástrojmi od tejto firmy v oblasti dát. Databázu je možné zálohovať na cloudovom úložisku MS Azure. Posledný z týchto nástrojov od firmy Oracle je podľa rebríčka najpoužívanejší. Firma tiež ponúka cloud-ové riešenie na zálohovanie dát. Systém je možné používať aj s rôznymi ďalšími nástrojmi, pomocou ktorých je možné vytvárať databázové aplikácie [28].

Počas implementácie databázy vytvárame rôzne databázové objekty. Začíname tvorbou tabuliek a definovaním typov stĺpcov, ktorým určujeme aké typy dát budú v jednotlivých stĺpcoch. Máme možnosti vytvoriť aj ďalšie databázové objekty, ako sú indexy, pohľady alebo procedúry. Ich definovanie a používanie závisí od našich potrieb.

3.2 Zdroje

Po dokončení činností v predošlej časti máme predstavu o dátových zdrojoch a dátach v nich . Extrahovanie dát môže byť vykonané:

- vytvoreným programom/skriptom pre tento účel,

- zakúpeným softvérom alebo open-source softvérom,
- kombináciou predošlých typov.

Výber vhodného softwaru závisí od množstva kritérií, v ďalšej časti predstavíme niektoré z nich.

Tvorba programu/skriptu

Existuje množstvo programovacích jazykov, pomocou ktorých je možné vytvoriť skript alebo program na extrakciu dát z webových stránok. Avšak dva programovacie jazyky R a Python ponúkajú širokú škálu možností na prácu s rôznymi typmi dátových zdrojov a dát. Pomocou týchto jazykov je možné:

- "zavesiť" sa na webovú stránku a extrahovať dáta,
- analyzovať dáta a vizualizovať výsledky v rôznych grafických podobách,
- vytvoriť prepojenie medzi jazykom a rôznymi typmi databázových softvérov.

Jazyk R je programovací jazyk, určený najmä pre štatistické výpočty a vizualizáciu výsledkov týchto metód. V jazyku sme schopní pracovať s veľkým množstvom dát, ktoré môžeme importovať z rôznych súborov. Jazyk zahŕňa veľké množstvo metód na klasifikáciu, zhlukovanie alebo predikciu dát a štatistických funkcií akými sú ANOVA alebo Chi-square test. Program je neustále vyvíjaný, čím je zabezpečené, že v programe sú dostupné najnovšie metódy a technológie. Nevýhodami jazyka R sú zložitosť príkazov najmä pre začiatočníkov a nedokonalé balíky obsahujúce rôzne metódy a funkcie [24].

Programovací jazyk Python je definovaný ako objektovo-orientovaný jazyk s dynamickou štruktúrou, v ktorom sme schopní pracovať s rôznymi dátovými štruktúrami a typmi dát. V Pythone je možné využívať rôzne štýly programovania: objektovo-orientované, procedurálne a štrukturálne a je len na užívateľovi, ktorý štýl programovania si vyberie. Jazyk podporuje rôzne štatistické metódy, avšak v porovnaní s jazykom R neposkytuje tak veľa možností na vizualizáciu výsledkov [25].

Softvéry

Na trhu existuje veľký počet softvérov nazývaných Web Scraping nástroje, prostredníctvom ktorých je možné extrahovať dáta z webových stránok. Niektoré z nich sú voľne dostupné v základnej verzii a za rozšírenú verziu, ktorá ponúka viac možností, si musí

užívateľ zaplatiť, za používanie iných je treba zaplatiť, bez ohľadu na rozsah používaných služieb.

Príkladom týchto typov nástrojov sú [26]:

- Import.io – ponúka možnosť extrahovania dát z viac ako tisíc webových stránok a uložiť dáta CSV súboru, bez znalosti programovania.
- Webhose.io – pomocou špeciálnej techniky na prehľadávanie webových stránok extrahuje štruktúrované dáta do formátov XML, JSON a RSS.
- Spinn3r – umožňuje extrahovať aj neštruktúrované dáta z blogov alebo sociálnych médií a ukladá dáta do JSON formátu.

Tieto nástroje sú zvyčajne cenovo výhodné, avšak malé množstvo týchto typov nástrojov umožňuje ukladať dáta do relačných databáz. Dáta sú ukladané do CSV, JSON alebo XML formátov, ale ak chceme dáta ukladať do databáz, je nutné použiť ďalší nástroj na prenos dát, alebo vytvoriť program/skript s touto funkciou.

Ďalšou možnosťou je zakúpenie licencie na cloud-ové platformy, ako sú IBM WebSphere alebo Microsoft Azure. Tieto nástroje sa vyznačujú svojou komplexnosťou, nakoľko umožňujú nielen sťahovať dáta z rôznych typov zdrojov, ale je aj možné aplikovať rôzne štatistické metódy a poskytujú úložiská pre dáta. Súčasťou nástrojov býva aj vývojové prostredie, čím je zabezpečená možnosť dodatočného vývoja nových funkcií. Tieto nástroje však bývajú cenovo drahšie a integrácia s ďalšími používanými technológiami v organizácii je spojená s finančnými nákladmi.

Poslednou možnosťou je kombinácia zakúpeného nástroja na extrakciu a nami vytvoreného programu alebo skriptu, napríklad skript v systéme Unix alebo program v jazyku Cobol [27].

Výber správneho nástroja na extrakciu dát závisí nielen od finančných možností, ale aj od našich schopností a znalostí. Preto je pri voľbe vhodného nástroja zohľadňovať aj tieto skutočnosti.

Počas extrakcie dát je vhodné hľadať spôsoby triedenia sťahovaných webových dokumentov, dát alebo súborov, aby neprišlo k neprehľadnosti medzi nimi. Je dôležité separovať rôzne typy dát, aby neprišlo v budúcnosti zmätkom pri analýze dát. Minimálne by sa mali oddeliť štruktúrované a neštruktúrované typy dát, nakoľko neštruktúrované dáta je nutné ďalšími procesmi transformovať do štruktúrovanej formy. Stiahnuté dáta sú ukladané

do dočasného úložiska dát, v ktorom sa nachádzajú len aktuálne stiahnuté dáta. V dočasnom úložisku nenájdeme historické dáta a ak sú dáta použité v ďalších procesoch (Text Mining, Formátovanie), sú mazané z dočasného úložiska. Pri nastavovaní intervalov extrahovania resp. mazania dát do resp. z dočasného úložiska musíme brať do úvahy minimálne tieto faktory:

- ako často vznikajú nové dáta v našich zdrojoch,
- či chceme mať k dispozícii dáta, ktoré vznikli pred hodinou, dnes alebo nám stačia dáta zo včera,
- od pamäťového miesta na dočasnom úložisku,
- znížiť riziko straty v prípade, že by prišlo k nečakanej chybe v procesoch, ktorá by spôsobila poškodenie alebo stratu dát.

Posledný faktor výrazne ovplyvňuje čas, kedy budú dáta vymazané z dočasného úložiska. Tieto typy úložísk nemajú zvyčajne veľké množstvo pamäti, a preto je častá iniciatíva, aby dáta boli ihneď vymazané už počas procesov a prenechali miesto novým dátam. Tento krok môže spôsobiť, že dáta budeme mať k dispozícii rýchlejšie. Avšak podstupujeme riziko, že v prípade neočakávanej technickej alebo procesnej chyby, môžeme o dáta prísť. Možnosťou je vytvoriť ešte jedno dočasné úložisko, kam by sme ukladali dáta, s ktorými sa v tom čase pracuje. S týmto krokom sú však spojené finančné náklady aj väčšia zložitosť v architektúre modelu a previazanosti jednotlivých úložísk dát v modeli.

3.3 Transformácia dát

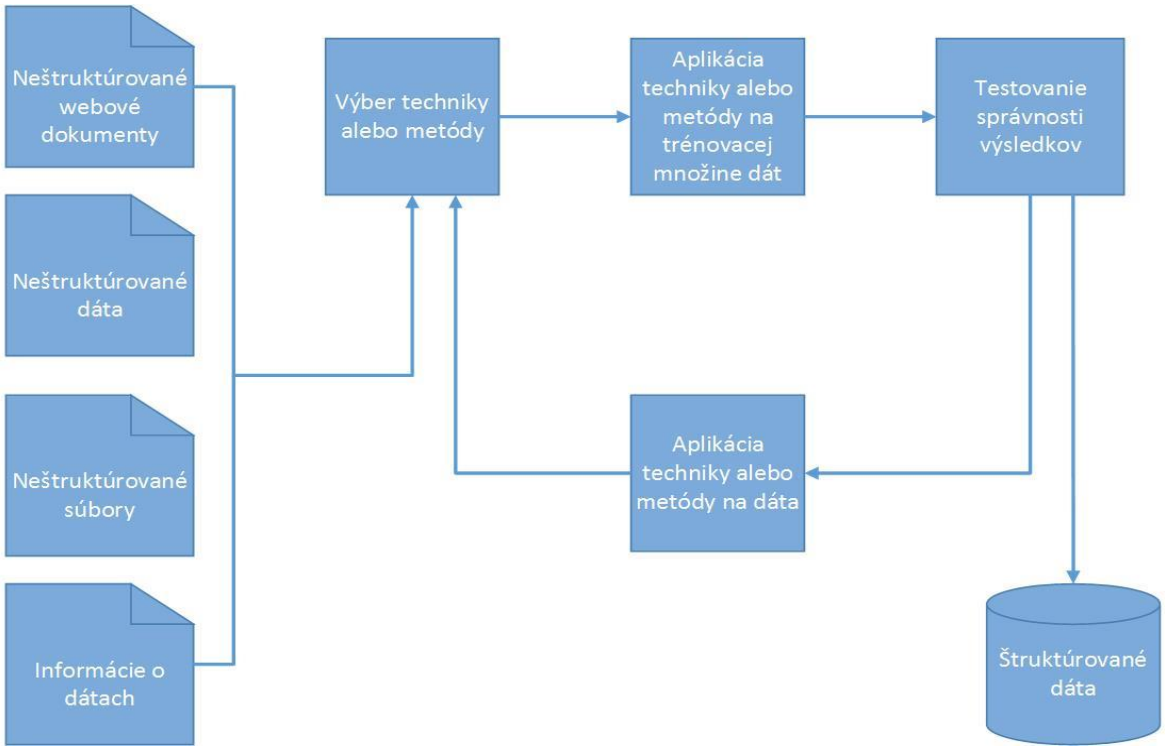
Po dokončení extrakcie dát z dátových zdrojov máme dáta roztriedené v dočasnom úložisku. Štruktúrované typy dát sme schopní transformovať do spoločných formátov. Druhou skupinou dát sú neštruktúrované dáta, z ktorých je treba vybrať len dáta, ktoré sú pre nás zaujímavé a majú informačnú hodnotu. Tieto dáta by bolo možné najskôr uložiť do relačných databáz spolu s ostatnými dátami a potom z nich extrahovať informácie a ukladať ich do databáz. Avšak v našom predkladanom modeli tieto dáta neukladáme do databáz, ale z nich extrahujeme informácie a ukladáme do databáz. Týmto postupom znížime čas dostupnosti dát v databázach a nezahltíme pamäť databázy dátami, s ktorými už v budúcnosti nebudeme pracovať. V prípade vyskytnutia sa chýb v procesoch extrahovania dát, sú neštruktúrované dáta v dočasnom úložisku do určitého času a môžeme z nich opakovane extrahovať dáta.

3.3.1 Spracovanie neštruktúrovaných dát

Aplikovaním Text Mining-ových techník a metód (podrobnejšie vysvetlené v kapitole 1.5) získame dáta vhodné pre analýzy a uloženie do relačných databáz. Neexistuje presný návod, ktoré techniky a metódy sú najlepšie alebo najvhodnejšie. Výber závisí od dát a môžeme povedať, že každá množina dát vyžaduje inú kombináciu techník a metód. V tejto časti modelu odporúčame najskôr pracovať s Text Mining-ovými technikami, prostredníctvom ktorých znížime počet dát a odstránime "zbytočné" dáta. Následne použijeme Text Mining-ové metódy, s ktorými sme schopní:

- triediť dokumenty klasifikačnými a zhlukovacími metódami,
- extrahovať kľúčové dáta,
- vizualizovať výsledky metód.

Nakoľko v druhej časti sme analyzovali aj neštruktúrované dátové zdroje, znalosti z tejto fázy môžu byť teraz pre nás užitočné. Na Obr. 7 je zobrazený proces, ako by sme mohli transformovať neštruktúrované dáta do štruktúrovanej podoby prostredníctvom Text Mining-u.



Obr. 7 Proces výberu a aplikovania Text Mining-ových techník

Na začiatku máme zhromaždené neštruktúrované dáta z predchádzajúcej fázy modelu. Najskôr vyberáme medzi Text Mining-ovými technikami. Vybranú techniku

použijeme na dáta a následne otestujeme výsledky použitej techniky. Je vhodné aplikovať najskôr techniky na trénovaciu množinu dát, aby sme boli schopní efektívnejšie a rýchlejšie overiť správnosť použitia metódy. Po zistení správnosti výsledkov, môžeme použiť techniku na všetky dáta. Ak zistíme chyby pri aplikácii, máme na výber dva prístupy:

a) Znovu použiť techniku tak, aby sa nezopakovali nájdené chyby z predošlej aplikácie.
b) Pri použití sa našli chyby, ktorým sa nejde vyhnúť a ukazujú, že daná technika je neefektívna pre naše dáta.

Týmto spôsobom postupujeme dovtedy, pokým nevyskúšame všetky techniky. Môžeme takto otestovať všetky techniky Text Mining-u, alebo vybrať len tie techniky, o ktorých sme presvedčení, že nám pomôžu.

Na zredukovanom počte dát použijeme Text Mining-ové metódy. Proces aplikovania týchto metód je rovnaký ako pri Text Mining-ových technikách. Najskôr otestujeme vybranú metódu na trénovacej množine dát. Následne overíme výsledky metódy a v prípade správnosti výsledkov aplikujeme metódu na všetky dáta. Vizualizáciu dát je možné použiť pre lepšie pochopenie výsledkov a dát, a tiež nám môže pomôcť pri rozhodovaní, či je vhodné použiť danú metódu.

3.3.2 Formátovanie dát

Po nasadení databázy do reálneho prostredia a tvorbe objektov, sme schopní transformovať štruktúrované dáta do formátu, vhodného pre našu databázu. V tejto fáze dochádza k čisteniu dát. Upravujeme všetky štruktúrované dáta, aj tie, ktoré boli extrahované z neštruktúrovaných dát. V rámci tohto procesu by sme mali vykonať tieto činnosti, aby dáta boli pripravené pre databázy:

- Obmedziť výskyt duplicitných dát - rovnaké dáta sa môžu nachádzať vo viacerých zdrojoch, preto je nutné tieto dáta zlúčiť a ukladať do databázy len jeden raz. Napríklad sa stane, že v zozname potenciálnych klientov nájdeme rovnakú firmu viackrát, nakoľko firma na webe alebo v zdrojovom súbore bola zmienená viackrát v rámci rôznych kategórii. My však nechceme typ firmy, ale len adresu. Preto uložíme identifikačný kód firmy, názov firmy a adresu do tabuľky len raz, aby sme našli informácie o firme.
- Transformovať dáta rovnakého typu do spoločného formátu – dáta pochádzajúce z rôznych zdrojov sú v rôznych formátoch. Takéto dáta treba upraviť do formátu definovanom v databáze. Napríklad v tabuľke sme definovali stĺpec váha produktu

a chceme záznamy uchovávať v kilogramoch. Avšak dáta o produktoch z jedného zdroja môžu byť kilogramoch, z druhého zdroja v gramoch a z tretieho v tonách. Preto všetky číselné údaje vyjadrujúce váhu produktu by mali byť do našej databázy ukladané v kilogramoch.

- Odstrániť alebo upraviť dáta, ktoré nespĺňajú štandardy – do databázy je nutné ukladať len úplne dáta. Napríklad chceme do databázy vložiť e-maily firiem. Môže nastať situácia, že niektoré emaily nebudú mať v adrese znak @. Tento prípad môžeme riešiť :
    a) Automatickým vkladaním znaku @ do emailovej adresy, ak tento znak sa v nej nenachádza. V tomto prípade musíme navrhnúť automat, ktorý bude schopný rozoznať nesprávnu adresu a vložiť @ na správne mesto v textovom reťazci.
    b) Odstránení takýchto adries a neukladaním do databázy.
    c) Tieto adresy dočasne uložíme do inej tabuľky, kde budú zhromažďované a ďalšími spôsobmi (napr. dodatočným osobným vyhľadávaním z iných zdrojov) zistiť správnu adresu.

  Každý z týchto prístupov má svoje klady aj zápory. Pri prvom prístupe musíme mať technické znalosti, aby sme boli schopní takýto automat navrhnúť. Pri druhom prístupe hrozí strata dát, najmä v situáciách, keď to je jediný možný kontaktný údaj na firmu. Pri treťom prístupe môže nastať problém, ak je takýchto adries niekoľko tisíc a museli by sme všetky manuálne vyhľadávať, čo je náročné časovo aj personálne.
- Rozbitie viacslovných dát – textové reťazce môžu obsahovať dáta, ktoré vyjadrujú a poskytujú rôzne informácie. Napríklad zo zdroja dostaneme adresu pozostávajúcu z ulice, mesta a PSČ. My však v tabuľke máme definované stĺpce ulica, mesto a PSČ. V tomto prípade je nutné oddeliť tieto dáta a vložiť do správnych stĺpcov.

Tieto činnosti by mali prispieť očisteniu dát do takej formy, aby sme boli schopní analyzovať dáta v databáze. Keď sú dáta v takej podobe, môžeme ich vložiť do predom definovaných a existujúcich tabuliek v databáze.

3.4 Uchovávanie dát

Jedným z hlavných cieľov prečo dáta uchovávame v databázach je, aby sme ich mali k dispozícii v budúcnosti, keď by nám mali pomôcť najmä v procese rozhodovania. Dát je však veľké množstvo v databázach a my chceme mať čo najkomplexnejšie informácie. Preto je nutné tieto dáta analyzovať a vizualizovať výsledky analýzy vo forme zrozumiteľnej pre

ľudí pôsobiacich na rôznych pozíciách. Proces, začínajúci od dát v databáze až po zobrazenia výsledkov analýz môže prebiehať pomocou BI nástroja alebo kombináciou činností, pri ktorých využijeme viacero nástrojov.

Na trhu je dostatok reporting-ových nástrojov, ktoré umožňujú získať dáta z databázy, aplikovať rôzne Data Mining-ové metódy na dáta a vizualizovať výsledky v podobe grafov alebo animácií. Výhodou týchto nástrojov je možnosť ich využívania bez znalosti programovacích jazykov a celý proces získania dát z databáz až po premenu na znalosti prebieha v jednom nástroji. Nevýhodou sú počiatočné investície do nástroja, či už finančné alebo časové, nakoľko integrácia nástroja s databázou a s ostatnými systémami je zložitá a často vyžaduje pomoc konzultantskej firmy. V mnohých prípadoch zamestnanci nemajú predchádzajúcu skúsenosť s týmito nástrojmi, nakoľko sa s nimi v ich predchádzajúcich zamestnaniach nestretli a ani na univerzitách nie sú časté kurzy, venované týmto typom programov. Takýmito nástrojmi sú napríklad Microsoft Power BI alebo QlikView.

Druhou možnosťou je zvoliť si proces, skladajúci sa z týchto činností:

- "Vytiahnuť" dáta z databázy v databázovom systéme prostredníctvom jazyka SQL.
- Analyzovať dáta prostredníctvom jednoduchých štatistických parametrov (priemer, medián, atď.) a Data Mining-ových metód.
- Vizualizovať výsledky metód.

Pri získaní dát z databáz SQL jazykom používame databázový systém, na ktorom "beží" aj naša databáza, čím nemusíme nakupovať dodatočné nástroje. Podmienkou sú znalosti a skúsenosti s týmto jazykom. Keď máme dáta z databázy stiahnuté, môžeme ich analyzovať a vizualizovať výsledky pomocou:

- programovacích jazykov ako R alebo Python,
- nástrojov ako EXCEL alebo RapidMiner,

ktoré umožňujú aplikovať rôzne Data Mining-ové metódy a následne vizualizovať výsledky metód. Úlohou obidvoch prístupov je zmeniť dáta na znalosti a aby boli k dispozícii čo najskôr a v čo najzrozumiteľnejšej podobe.

4 Aplikácia modelu na riešenie konkrétneho problému

V poslednej kapitole sme použili navrhnutý model z predošlej kapitoly na konkrétne riešenie úlohy. Ak spoločnosť vlastniaca internetový obchod (ďalej len obchod) chce zistiť, ako sú jej zákazníci spokojní s ponúkanými tovarmi a službami má len dve možnosti. Prvá možnosť je pýtať sa zákazníkov na ich názory prostredníctvom telefonátov alebo emailov. Táto forma však nemusí byť ideálna pre zákazníka, nakoľko môže to brať ako obťažovanie, ak mu zavoláme v jeho pracovnom čase alebo v čase, keď nemá čas odpovedať na naše otázky alebo poslaný mail môže byť zaradený do spamu. Druhou možnosťou sú sociálne siete a webové stránky ponúkajúce možnosť zdieľať svoje skúsenosti s obchodom. Zvyčajne tieto informácie hľadajú potencionálny zákazníci obchodu, avšak môžu byť užitočné aj pre obchod, ktorého sa týkajú recenzie. Spoločnosť môže z týchto dát zistiť napríklad:

- aké oblasti smerom k zákazníkom by mala vylepšiť,
- či navrhnuté procesy zlepšili spokojnosť zákazníkov.

Tieto komentáre sú na stránke ako neštruktúrované dáta. Zákazníci nevypĺňajú anketu, kde by len zaškrtli odpoveď na otázky, ale píšu komentáre vo vetách a preto sme ich označili ako neštruktúrované. Avšak pre spoločnosť by bolo veľmi náročné spracovať tieto komentáre manuálne zamestnancami. Napríklad len na stránke spoločnosti Heureka.sk, kde nájdeme recenzie rôznych obchodov, na spoločnosť MALL.SK je 73200 recenzií [30]. Preto sme sa rozhodli na túto úlohu použiť náš model, pomocou ktorého dokážeme extrahovať tieto komentáre automaticky, čím sa dosiahne, že výsledky budeme mať k dispozícii za pár minút. Po získaní recenzií je z týchto komentárov ťažké vyhodnocovať, čo zákazníkom obchodu prekáža a naopak čo je devízou spoločnosti vlastniacej obchod. Tieto komentáre sú často dlhé, v rôznych formátoch a obsahujú veľa "zbytočných" dát. Transformáciou tieto problémy odstránime a výsledky budeme ukladať pre ďalšie použitia.

4.1 Stratégia modelu

V prvej fáze modelu sme sa zamerali na:

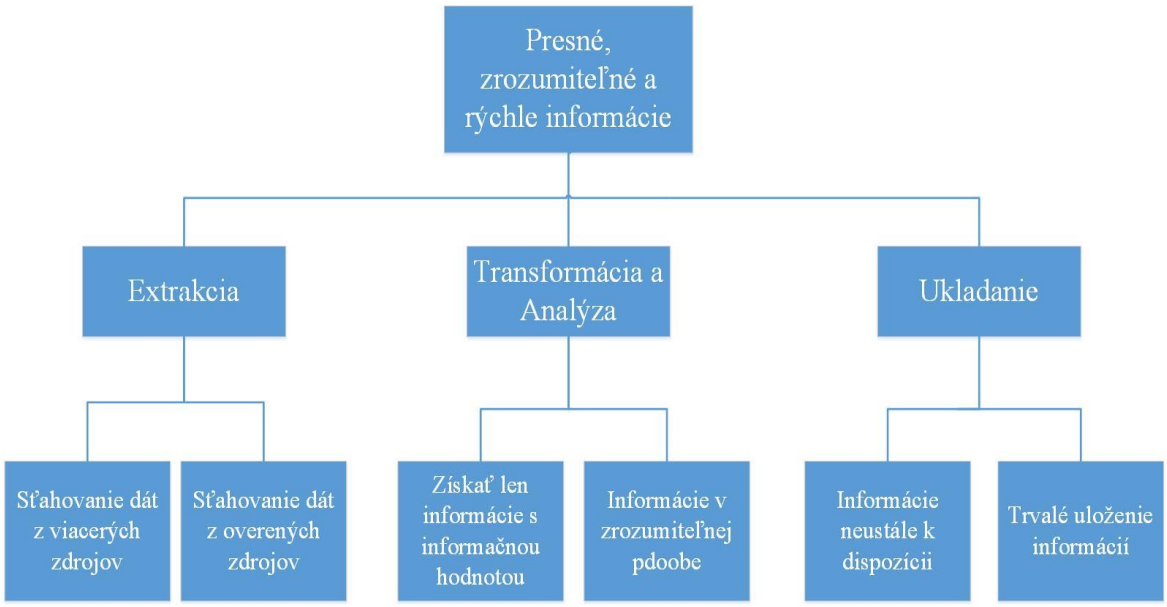
- Identifikáciu cieľov, ktoré chceme dosiahnuť naším modelom a definovaniu cieľov do hierarchickej štruktúry.
- Identifikovať a analyzovať vybraté webové stránky.
- Návrh procesov a výber vhodných technológii v modeli.
- Návrh dátových modelov a ich implementácia.

4.1.1 Identifikácia cieľov modelu

Pri identifikácii cieľov sme museli brať do úvahy naše finančné a znalostné možnosti. Finančné ovplyvňovali v našom prípade najmä technológie, ktoré môžeme použiť v modeli. Pri identifikácii nám pomohli typy otázok spomenutých v kapitole 3.1. Na prvú otázku z akého dôvodu sa bude model používať, sme odpovedali v predošlej kapitole. Na druhu a tretiu otázku, ako bude model využitý a či model bude spĺňať požiadavky sme boli schopní odpovedať v tejto časti. Model bude využitý na extrakciu dát z webových stránok, na ktorých nájdeme recenzie o obchodoch a informácie o zákazníkoch.

Medzi naše základné ciele sme zaradili extrakciu dát. Extrahovať dáta sme chceli z čo najviac dátových zdrojov, avšak chceli sme extrahovať dáta len z webových stránok, na ktorých neprebiehajú nekalé praktiky, ako napríklad mazanie negatívnych recenzií. Ďalej by sme chceli modelom získavať z recenzií len tie informácie, ktoré budú užitočné pre spoločnosť. Nechceli sme transformáciou informácií získať z recenzií veľké množstvo dát, ktoré nebudú spoločnosti poskytovať informácie v zrozumiteľnej podobe. Informácie sme chceli mať v čo v najkratšom čase, aby boli k dispozícii spoločnosti. Keďže sme chceli, aby informácie boli k dispozícii 24 hodín 7 dní v týždni a aby boli dostupné aj za viacero rokov, bolo nutné vytvoriť takéto dátové úložisko.

Po identifikácii sme zostrojili hierarchický diagram, pozostávajúci z definovaných cieľov modelu, Obr. 8.



Obr. 8 Hierarchický diagram cieľov modelu

Hlavný cieľ nášho modelu je poskytovať presné, zrozumiteľné a rýchle informácie. Vedľajšie ciele sme roztriedili do troch kategórií: Extrakcia, Transformácia a Analýza, a Ukladanie. Do kategórie Extrakcia patria ciele, ktorými sme chceli dosiahnuť, aby sme mali k dispozícii, v čo najväčšej miere správne dáta. V kategórii Transformácia a Analýza sme definovali ciele, splnením ktorých by spoločnosti mali informácie, ktoré im pomôžu definovať nedostatky, pokroky alebo úpadky v jednotlivých oblastiach. Ciele v poslednej kategórii nám mali pomôcť dosiahnuť efektívne a bezpečné uchovávanie informácií, a rýchly prístup k dátam.

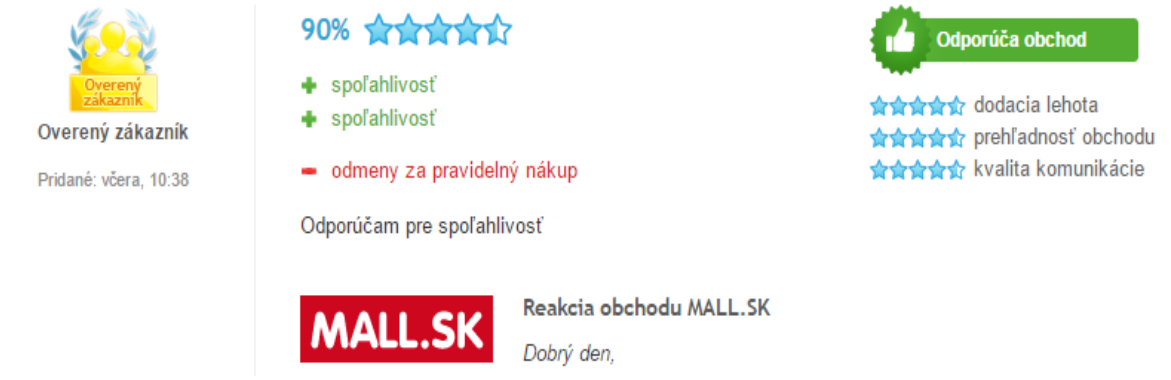
4.1.2 Analýza dátových zdrojov

Po definovaní cieľov sme vybrali dátové zdroje, ktoré poskytujú dáta zaujímavé pre náš model. Zamerali sme sa na webové sídla, ktoré poskytujú priestor pre zákazníkov hodnotiť obchody. Podarilo sa nám nájsť tri webové sídla, na ktorých je možné vyjadriť svoje skúsenosti s obchodom alebo nájsť informácie, čo sa ľudom páči a nepáči na obchode:

- https://www.heureka.sk/ (ďalej len heureka),
- https://www.najnakup.sk / (ďalej len najnakup),
- https://new-webdizajn.sk.

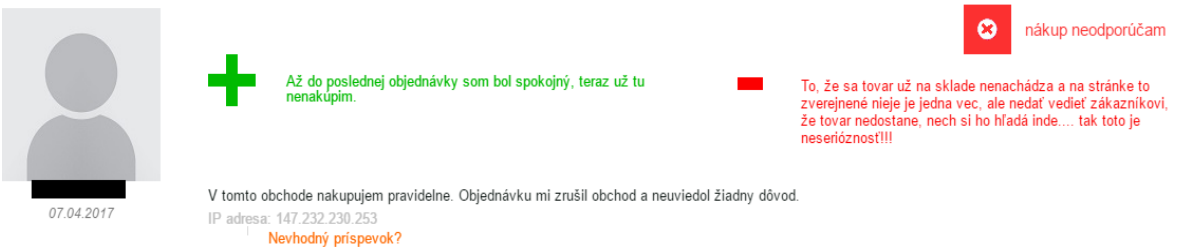
Po vybratí týchto zdrojov sme mohli začať s analyzovaním, či webové sídla poskytujú aktuálne a pravdivé dáta. Zamerali sme sa, či webové sídlo poskytuje dostatok aktuálnych dát a či nie sú na niektorom z nich konané nekalé praktiky, ako napríklad mazanie negatívnych komentárov, aby mal obchod lepšie skóre. Prvé dve webové sídla spĺňali naše kritéria, nakoľko sa objavuje na nich dostatok aj pozitívnych aj negatívnych recenzií. Na treťom webovom sídle https://new-webdizajn.sk sú dva roky staré dáta, ktoré pre nás neboli zaujímavé, pretože za dva roky sa mohla kvalita tovarov a služieb v rámci obchodov diametrálne zmeniť. Preto sme sa rozhodli vylúčiť z našich dátových zdrojov posledné webové sídlo z nášho zoznamu.

Po vybratí dvoch webových sídiel sme začali s analyzovaním štruktúry webových stránok jednotlivých webových sídiel a analyzovaním, aké typy dát budeme extrahovať z týchto stránok. Na Obr. 9 je zobrazená recenzia na obchod MALL.sk [31].



Obr. 9 Príklad recenzie z heureky

Recenzia pozostáva z viacerých častí. Na ľavej strane stránky je zobrazený užívateľ, ktorý pridal túto recenziu. Heureka ponúka možnosť anonymizovať užívateľa, čím nemôžeme analyzovať recenzie na základe veku alebo pohlavia, nakoľko niektorí užívatelia nie sú ochotní pri recenzii zverejniť svoju identitu. V strede stránky sú tri typy recenzií: plusy, zápory a možnosť vyjadriť svoj názor a tento komentár nebude triedený do plusov alebo mínusov. Všetky tri typy komentárov sú pridávané formou okna, do ktorého sa užívatelia môžu bez obmedzenia vyjadriť (heslovito alebo vo vetách). Z tohto dôvodu sme mohli označiť tieto dáta za neštruktúrované. Napravo od komentárov je informácia, či odporúča užívateľ obchod a jeho hodnotenie na tovary a služby obchodu. Pod komentármi je vyhradený priestor pre vyjadrenie spoločnosti na recenziu. Na stránkach najnakup sú dáta jednej recenzie zobrazené v tomto formáte [32].



Obr. 10 Príklad recenzie z najnakup

Recenzie sú zobrazované v podobnom formáte. V ľavej časti je zobrazené prihlasovacie meno užívateľa. V strede sú komentáre, naľavo pozitívne a napravo negatívne. Tieto komentáre sú rovnako neštruktúrované dáta, nakoľko užívatelia pri nich nie sú

obmedzovaní a môžu písať vo vetách alebo heslovito. Nad negatívnymi komentármi je informácia, či užívateľ odporúča obchod. V spodnej časti recenzie je priestor pre ďalšie vyjadrenie užívateľa, avšak tieto komentáre nie sú písane užívateľom, ale užívateľ si vyberie formou checkboxov, s akými tvrdeniami súhlasí a tieto sú zobrazované na stránke. Tieto komentáre sme preto označili za štruktúrované dáta. Po preskúmaní zdrojových kódov webových stránok sme boli schopní prehlásiť, že spĺňali štandardy W3C spoločnosti. Testovanie stránok sme realizovali aj formou, či sa dostaneme k dátam cez html tágy, čo sa nám podarilo vo všetkých prípadoch.

Po zistení s akými typmi dát budeme pracovať a štruktúre dátových zdrojov sme definovali procesy a technológie použité v modeli. Základné procesy v modeli sme definovali v predchádzajúcej kapitole (Obr. 6), teraz sme však tieto procesy prispôsobili na našu úlohu a navrhovali v rámci nich vhodné nástroje, techniky a metódy. Model pozostáva z troch častí: Extrakcia, Transformácia a Ukladanie.

4.1.3 Návrh procesov a funkcionalít

Pri návrhu procesov a funkcionalít sme definovali ako model má fungovať a tok dát v modeli v jednotlivých častiach implementačnej fázy.

Extrakcia

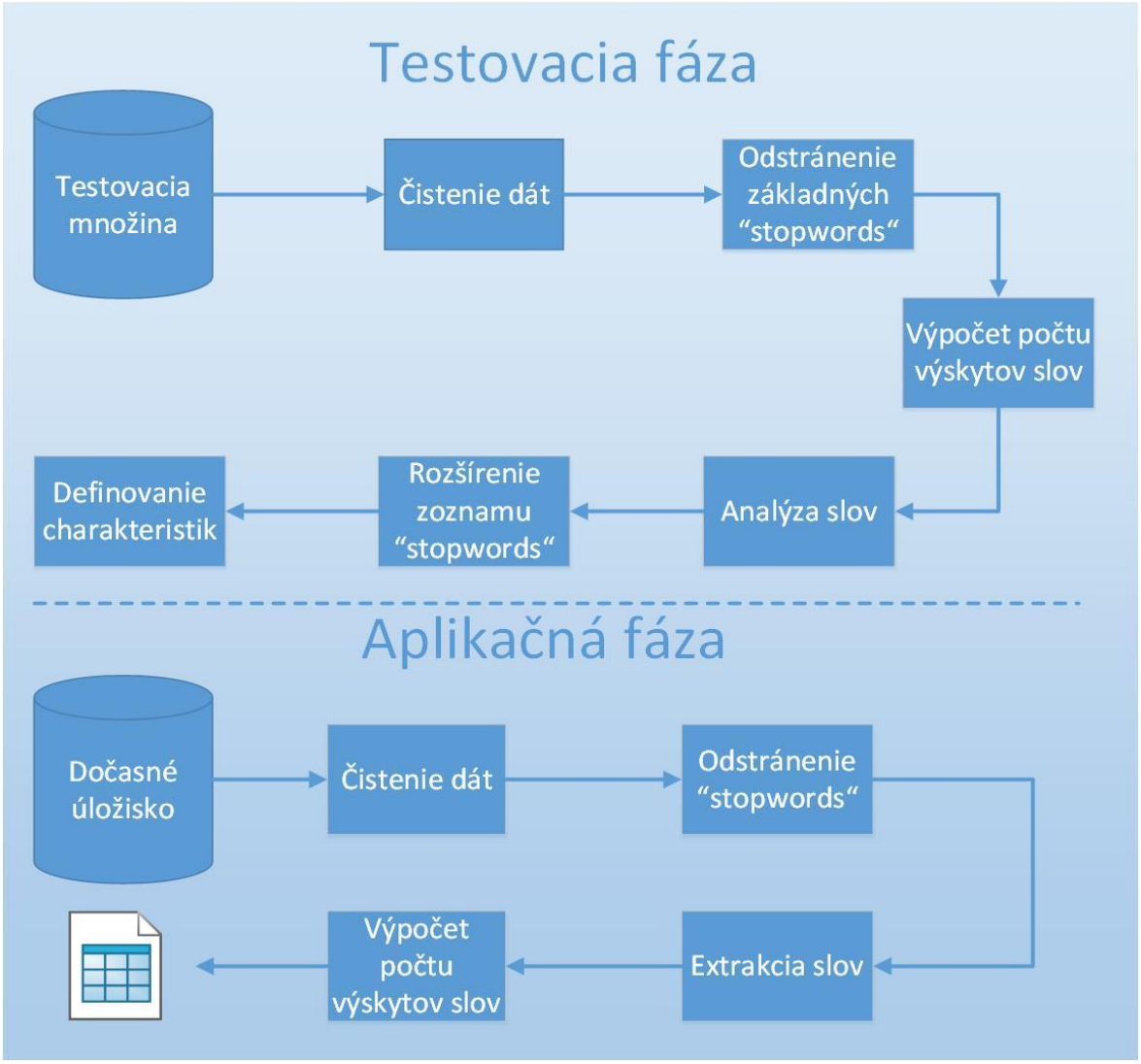
Automatizovane sme chceli sťahovať dáta z dvoch webových stránok. Po analýze dát sme sa rozhodli extrahovať komentáre užívateľov na webových stránkach, ktoré majú definovaný typ, či sú pozitívne alebo negatívne. Na základe analýzy štruktúry webových stránok sme sa rozhodli vytvárať web wrapper, prostredníctvom ktorého dokážeme sťahovať webové dokumenty. Úlohou web wrappera bude nielen sťahovať tieto dokumenty, ale aj ukladať dáta z dokumentov do dočasného úložiska. Extrahovať a ukladať sme chceli len tie dáta, s ktorými budeme neskôr pracovať. Na extrakciu dát z dokumentov sme sa rozohdli využiť techniku XPath definovanú v kapitole 1.2. Stiahnuté webové dokumenty spĺňajú štandardy W3C spoločnosti, čo umožňuje, aby technika bola použitá efektívne. Pomocou techniky dokážeme v krátkom časovom úseku extrahovať vybraté dáta na základe zvolenej cesty pozostávajúcej z HTML tágov. Avšak museli sme počítať s možnosťou, že štruktúry webových stránok jedného z webových sídiel budú zmenené a my budeme musieť definovať novú cestu k dátam.

Miesto pre dočasné úložisko dát sme sa rozhodli vytvoriť samostatnú databázu, v ktorej budeme uchovávať stiahnuté komentáre v neštruktúrovanej podobe 6 mesiacov od

vloženia do databázy. Pre tento prístup sme sa rozhodli, nakoľko stiahnuté komentáre budeme transformovať do štruktúrovanej podoby tak, že budeme extrahovať len dáta, ktoré budú informovať o kvalite tovarov a služieb obchodu. Počas transformácie však môže dôjsť k výpadku alebo chybe v transformácii, na ktorú prídeme až po niekoľkých hodinách alebo dňoch. Táto databáza nám umožní opäť získať tieto komentáre v neštruktúrovanej podobe a opakovať proces transformácie v prípade neočakávanej chyby alebo výpadku.

Transformácia

V tejto fáze sme najskôr pracovali s neštruktúrovanými dátami. Výsledkom tejto fázy dostaneme zoznam slov, ktoré budú charakterizovať obchod a ich výskyt. Následne tieto dáta upravíme do formátu vhodného pre uloženie do databázy. Popis procesu spracovania neštruktúrovaných dát je zobrazený na Obr. 11, ktorý pozostáva z dvoch fáz: testovacej a aplikačnej.



Obr. 11 Spracovanie komentárov

V testovacej fáze budeme pracovať s množinou náhodne vybraných komentárov o rôznych obchodoch, aby sme mali k dispozícii rôznorodé dáta. Na komentáre sme najskôr navrhli aplikovať techniky na čistenie dát. Rozhodli sme sa využiť textové operácie ako:

- odstránenie veľkých písmen,
- odstránenie interpunkčných znamienok,
- odstránenie mäkčeňov a dĺžňov,
- odstránenie číslic,
- nahradenie dlhých a mäkkých slabík za krátke,
- lemitizácia.

V testovacej fáze chceme postupovať podľa Obr. 7. Po aplikovaní každej techniky skontrolujeme, či technika je správne aplikovaná, nakoľko v aplikačnej fáze už nebude možné overovanie správnej funkcionality každej techniky a bolo mi komplikovanejšie testovať nové techniky. Po vyčistení dát odstránime z komentárov slová, ktoré neposkytujú informačnú hodnotu nazývané "stopwords". Tento zoznam sme získali z webovej stránky http://text.fiit.stuba.sk [33]. Zoznam slov však budeme musieť rozšíriť o nespisovné slová a skratky, ktoré využívajú často užívatelia pri písaní recenzií. Definitívny zoznam týchto slov získame v testovacej fáze pri analýze slov, kedy štatistiky zistíme, ako často sa jednotlivé slová vyskytujú v textoch. Tieto štatistiky získame metódami na výpočet výskytu slov v texte. Pomocou týchto štatistík identifikujeme zoznam slov, ktoré definujú napríklad predávaný tovar alebo službu, kvalitu služieb pri doručení alebo spokojnosť so zamestnancami. Nakoľko v slovenskom jazyku existujú synonymá, vytvoríme pre naše účely krátky synonymický zoznam, v ktorom ku každej hlavnej charakteristike budú priradené ďalšie slová s rovnakým významom.

Výsledkom aplikačnej fázy bude zoznam našich charakteristík a ich výskyt v komentároch o jednom obchode. Rovnako ako v testovacej fáze, budeme aplikovať techniky na čistenie dát a odstránime slová bez informačnej hodnoty. Následne extrahujeme z komentárov len tie slová, ktoré sme vybrali v testovacej fáze.

Posledný krok pred ukladaním dát do databáz je ich formátovanie do tvaru vhodného pre našu navrhnutú databázu. Rozhodli sme sa, že dáta o obchodoch (názvy, kontaktné údaje, atď.) a sledované charakteristiky budeme ukladať predom definovaných tabuliek, nakoľko ich potrebujeme mať k dispozícii pri aplikačnej fáze. Z tohto dôvodu budeme formátovať

výsledky spracovania recenzií a to na základe formátu tabuľky, ktorú navrhneme v ďalšom kroku.

Uchovávanie

Dáta budeme mať k dispozícii v databáze, odkiaľ si ich budeme môcť hocikedy stiahnuť a spracovať. Dáta musia byť k dispozícii vo forme vhodnej pre analýzu a vizualizáciu. Úlohou vizualizácie je pomôcť lepšie pochopenie výsledkov. Pri vizualizácii sme sa chceli zamerať na:

- porovnanie charakteristík služieb a tovarov v rámci kladných a negatívnych komentárov,
- vývoj vybratých charakteristík v čase.

Rozhodli sme sa implementovať náš model vo vývojovom prostredí R-Studio a databázovom systéme MS SQL Server (ďalej MS SQL). V R-studio je využívaný programovací jazyk R a v tomto prostredí zimplementujeme prvú, druhú fázu modelu a vizualizáciu dát v rámci tretej fázy. Dočasné úložisko dát a navrhnutý dátový model budeme implementovať do databázového systému MS SQL formou dvoch databáz.

V ďalšej časti navrhneme štruktúru dočasného úložiska a databázy, v ktorej budeme ukladať štatistiky zo spracovania recenzií a informácií súvisiacimi s týmito dátami.

4.1.4 Návrh dátových modelov a implementácia

V rámci nášho modelu sme sa rozhodli navrhnúť dva dátové modely pozostávajúce z tabuliek a atribútov. Prvý dátový model Monitoring bude pozostávať z atribútov obsahujúcich informácie o sledovaných obchodoch, charakteristikách a dátových zdrojoch. V ďalších troch tabuľkách sú zobrazené atribúty z jednotlivých oblastí a ich stručný popis:

- sledované obchody,

Tab. 6 Zoznam atribútov pre sledované obchody

Popis	Názov atribútu
Identifikačné číslo obchodu	idObchodu
Obchodný názov spoločnosti	Nazov
Adresa sídla spoločnosti	Sidlo
Identifikačné číslo internej kategórie	idKategorie
Typy výrobkov ponúkané v obchode	Kategoria

- charakteristiky,

Tab. 7 Zoznam atribútov pre sledované obchody

Popis	Názov atribútu
Identifikačné číslo skupiny zahŕňajúca významovo rovnaké charakteristiky	idSkupiny
Hlavná charakteristika zastupujúca ostatné charakteristiky s rovnakým významom	zaklCharakteristika
Identifikačné číslo charakteristiky	idCharakteristiky
Slovo pomenúvajúce tovary, služby alebo aktivity súvisiace s obchodom	Charakteristika
Identifikačné číslo typu komentára	idHodnotenia
Typ komentáru [plusy/mínusy]	Typ
Počet výskytov danej charakteristiky	Pocet
Dátum sťahovania dát	Datum

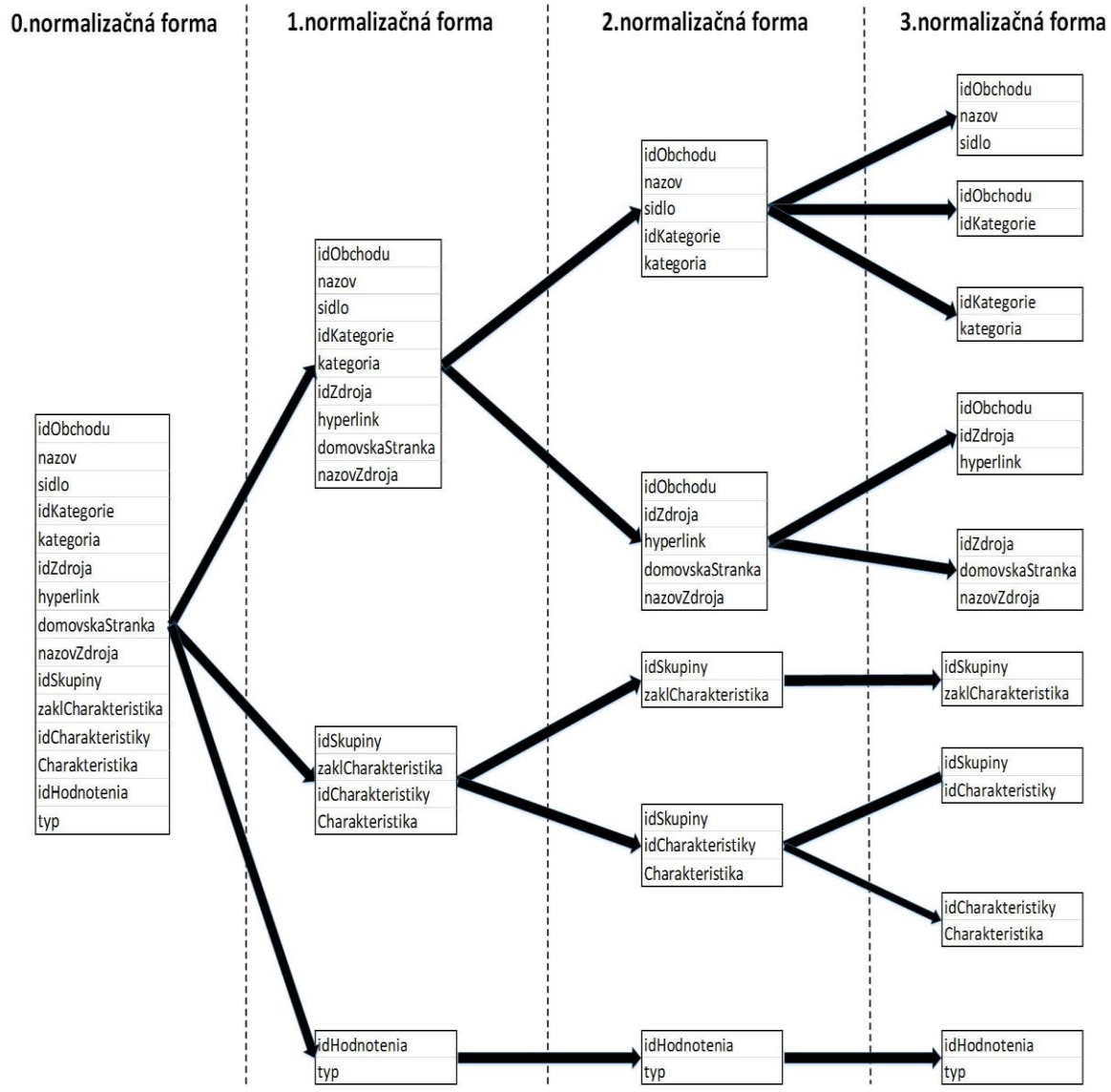
- dátové zdroje.

Tab. 8 Zoznam atribútov pre sledované obchody

Popis	Názov atribútu
Identifikačné číslo zdroja	idZdroja
Webová adresa na prvú stránku, kde sa nachádzajú recenzie užívateľov na vybraných obchod	hyperlink
Domovská webová adresa spoločnosti, poskytujúca recenzie užívateľov	domovskaStranka
Názov spoločnosti poskytujúca recenzie	nazovZdroja

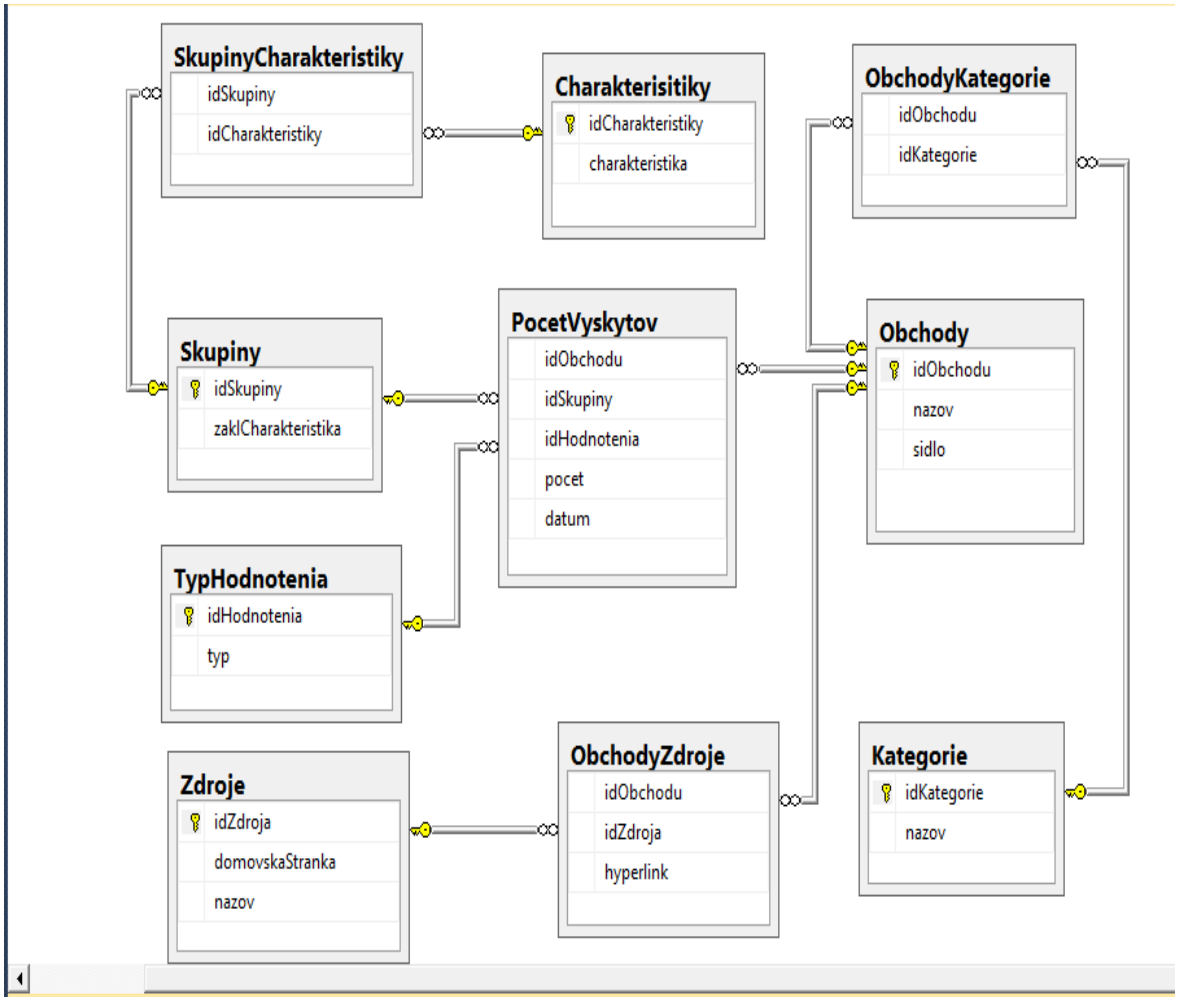
Rozhodli sme sa použiť ako schému dátového modelu snehovú vločku, nakoľko v databáze budeme uchovávať nielen metriku (atribút pocet), ale aj informácie o charakteristikách, obchodoch a dátových zdrojoch. Tabuľky obsahujúce tieto typy dát sme znormalizovali, aby sme sa vyhli duplicitným dátam v našej databáze. Faktová tabuľka bude pozostávať z piatich atribútov: idObchodu, idSkupiny, idHodnotenia, pocet a datum. Ostatné atribúty sme použili pri procese normalizácie. V nultej normalizačnej forme sme

zobrazili všetky atribúty do jednej tabuľky. Do prvej normalizačnej formy sme sa dostali rozdelením atribútov do troch tabuliek v závislosti od toho, na aký identifikačný kľúč sa viazali. Zvolili sme tri identifikačné kľúče (idObchodu, idSkupiny a idHodnotenia), ktoré sú vo faktovej tabuľke. Do druhej normalizačnej formy sme sa dostali vytvorením tabuliek pre atribúty, ktoré boli len čiastočne závislé na primárnom kľúči. Prvú tabuľku sme rozdelili, nakoľko posledné štyri atribúty sa úplne neviažu na idObchodu. Druhú tabuľku sme rozdelili z rovnakého dôvodu pre posledné dva atribúty. Vytvorením nových tabuliek pre atribúty, ktoré neboli závislé na primárnom kľúči tabuľky, sme získali dátový model v tretej normalizačnej forme. Z druhej normalizačnej formy sme rozbili prvú tabuľku, pretože atribút kategória je závislý len na idKategórii. Rovnako sme postupovali pri všetkých tabuľkách a normalizovaný dátový model je zobrazený na Obr. 12.



Obr. 12 Normalizácia dátového modelu

Po návrhu sme implementovali dátový model do prostredia MS SQL, v ktorom sme vytvorili databázu s názvom Monitoring. Pri tvorbe databázy sme definovali názvy jednotlivých tabuliek, typy atribútov a definujeme databázové objekty pre jednotlivé atribúty.



Obr. 13 Implementovaná databáza

Pri návrhu dátového modelu Dočasného úložiska sme použili atribúty idKomentara a komentar. Atribút idKomentara označuje identifikačné číslo každého stiahnutého komentára z webovej stránky a do atribútu komentar ukladáme jednotlivé stiahnuté komentáre. Dočasné úložisko bude pozostávať z jedného typu tabuliek. Nakoľko sme nechceli miešať v jednej tabuľke komentáre rôzneho typu (kladné a záporne), týkajúce sa rôznych obchodoch a stiahnuté v rôznych dátumoch, budeme na základe týchto kritérií komentáre ukladať do existujúcich tabuliek alebo vytvárať nové. Názov tabuľky pozostávajú z identifikačného čísla obchodu, z identifikačného čísla typu hodnotenia a dátumu, kedy sa

tieto komentáre stiahli. Medzi jednotlivými identifikačnými údajmi bude znak podčiarkovník: „-“. Pre tento krok sme sa rozhodli, pretože nevieme presný počet komentárov uložených do tabuliek a chceli sme sa vyhnúť situácii, kedy by v jednej tabuľke bolo uložených veľké množstvo dát.

4.2 Implementácia modelu

V nasledujúcej kapitole ukážeme prostredníctvom fragmentov programov, ako sme implementovali navrhnuté funkcionality modelu. Fragmenty pozostávajú z časti kódov, ktoré sú súčasťou zdrojových kódov, uvedené v prílohe predkladanej diplomovej práce. Zdrojové kódy v prílohe sú obsiahlejšie a zahŕňajú viac funkcionalít, cieľom tejto kapitoly je vysvetliť časti kódu, ktorými sme realizovali hlavné návrhy a myšlienky z predošlej kapitole. Implementačnú fázu modelu sme realizovali vo vybraných vývojových prostrediach R-Studio a MS SQL, v ktorých sme zrealizovali navrhnuté funkcionality a procesy z predošlej časti. Dáta uložené v databáze potrebujeme pri extrakcii aj transformácii dát v R-Studio a transformované dáta z R-Studia budeme vkladať do databázy, preto sme na začiatku vytvorili spojenie medzi nimi.

```
> library(RODBC)
> db_Monitoring  <- odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server;Server=PC;
Database=Monitoring;Uid=; Pwd=; trusted_connection=yes")
> db_DocasneUlozisko  <- odbcDriverConnect("Driver=ODBC Driver 11 for SQL Server;Server
=PC; Database=Docasne_ulozisko;Uid=; Pwd=; trusted_connection=yes")
```

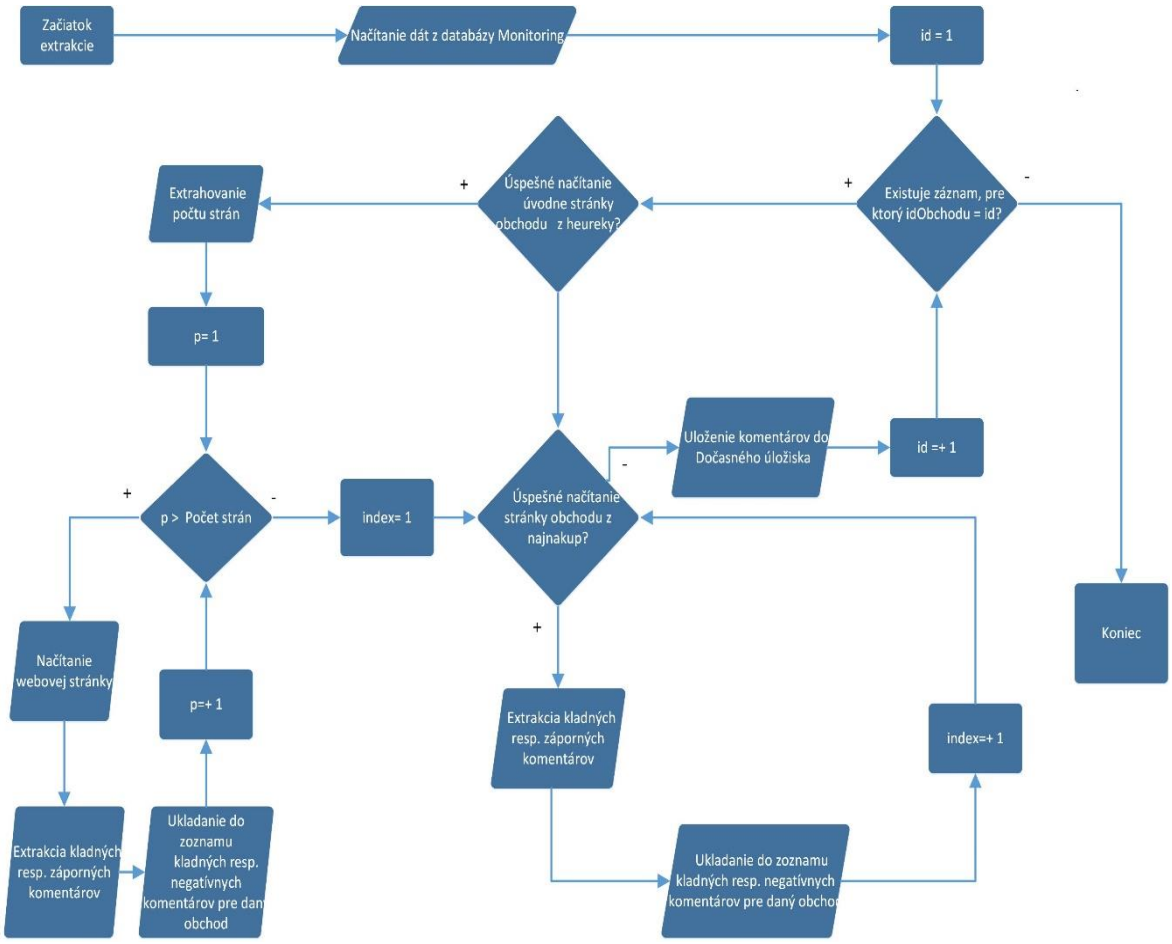
Fragment 1 Vytvorenie spojenia medzi vývojovými prostrediami

Na vytvorenie spojenia sme použili package RODBC a spojenia uložili do premenných “db_Monitoring“ a “db_DocasneUlozisko“, použitím ktorých dokážeme presunúť dáta z jedného prostredia do druhého.

4.2.1 Extrakcia

Pri extrakcii dát sme využili znalosti z analýzy dátových zdrojov. Proces extrakcie je zobrazený prostredníctvom vývojového diagramu, Obr.14. Vývojový diagram poskytuje návod, akými krokmi a postupmi získame dáta z vybratých dvoch webových sídiel. Nakoľko dáta z týchto dvoch zdrojov pochádzajú z viacerých webových stránok, ktoré sú súčasťou webových sídiel, museli sme nastaviť postupy spôsobom, aby sme boli schopní extrahovať všetky dostupné dáta. Diagram pozostáva z viacerých činnosti, ktoré na seba nadväzujú.

V rámci diagramu sme pracovali s dvomi typmi premennými. Prvým typom boli premenné uchovávajúce dáta (kladné a negatívne komentáre, informácie o obchodoch). Do druhej skupiny patrili premenné, do ktorých sme ukladali číselný údaj použitý pre cykly vo vývojom diagrame.



Obr. 14 Vývojový diagram extrakcie dát

V tejto časti ukážeme ako sme niektoré časti vývojového diagramu aplikovali v RStudio:

Načítanie dát z databázy - z databázy Monitoring sme chceli vytiahnuť informácie o obchodoch a url adresy. Pomocou jazyka SQL sme získali tieto dáta z databázy a uložili ich v R prostredí do dátového typu data frame, aby sme mohli pracovať s týmito dátami, ako keby boli v tabuľke.

```
> obchody_df <- data.frame(sqlQuery(db_Monitoring,
        'SELECT   o.idObchodu,
                  o.nazov,
                  ozh.hyperlink as url_heureka,
                  ozn.hyperlink as url_najnakup
          FROM    Obchody o
          JOIN    ObchodyZdroje ozh ON o.idObchodu = ozh.idObchodu AND ozh.idZdroja = 1
          JOIN    ObchodyZdroje ozn ON o.idObchodu = ozn.idObchodu AND ozn.idZdroja = 2'))
```

Fragment 2 Aplikáciu SQL jazyka v R-Studio

Podmienka na overenie existencie obchodu - na to aby sme začali extrahovať dáta z webových stránok museli sme mať obchody, ktoré chceme sledovať uložené v databáze a k nim aj potrebné údaje. Z data framu sme zistili, či sme mali taký obchod a ak ich je viac, pre každý z nich sťahneme dáta postupne.

```
> id <- 1
> while(length(obchody_df$nazov[obchody_df$idObchodu == id]) == 1) {..... id <- id + 1}
```

Fragment 3 Cyklus na overenie existencie ďalšieho obchodu na extrakciu

Načítanie prvej stránky na heureke o vybranom obchode - z data framu "obchody_df" sme získali url adresu pre vybratý obchod a stiahneme si prvú stránku, ktorá obsahuje aj komentáre.

```
> url <- levels(droplevels(obchody_df$url_heureka[obchody_df$idObchodu == id]))
> stranka <- read_html(url)
```

Fragment 4 Získanie url adresy a načítanie zdrojového kódu stránky

Získanie počtu strán - nakoľko recenzie bývajú na viacerých stránkach, museli sme vytvoriť cyklus, ktorý bude extrahovať webové stránky podľa toho, na koľkých sú recenzie o danom obchode. Túto informáciu sme získali extrakciou z premennej "stranka" pomocou definovania cesty k tomuto údaju.

```
> text <- stranka %>% html_nodes(".bot") %>% html_text()
> strany <- strapply(text, "\\d+", as.numeric, simplify = TRUE)
> pocet_stran <- strany[3,1]
> library(plyr)
> pocet_stran<-round_any(pocet_stran/10,1,f=ceiling)
```

Fragment 5 Získanie počtu strán

Extrakcia kladných a záporných komentárov - na základe štruktúry webových stránok na heureke sme určili cestu, ako extrahovať kladné a záporné komentáre. Komentáre boli ukladané do premenných typu vektor, v ktorej boli uložené v 1-dimenzionálne tabuľke.

```
> plusy_e <- stranka %>% html_nodes(".plus li") %>% html_text()
> minusy_e <- stranka %>% html_nodes(".minus li") %>% html_text()
```

Fragment 6 Extrakcia kladných a záporných komentárov

Načítanie úvodnej stránky pre daný obchod na najnakup - pri tomto webovom sídle sme museli extrahovať recenzie z jednotlivých webových stránok iným spôsobom, nakoľko na úvodnej stránke nie je informáciu o počte stránok, na ktorých budú recenzie. Avšak ak sa prihlásime na stránku na ktorej už nie sú recenzie, zobrazí sa webová stránka s nápisom "STRÁNKA UŽ NEEXISTUJE". Opäť sme definovali cyklus, v ktorom sa budú vykonávať príkazy, kým sa na stránke neobjaví spomínaný nadpis.

```
> index <- 1
> url <- levels(droplevels(obchody_df$url_najnakup[obchody_df$idObchodu == id]))
> while (tryCatch(read_html(url),error = function(e) {print("NEEXISTUJE STRÁNKA")}) !=
  "NEEXISTUJE STRÁNKA") {....index <- index + 1}
```

Fragment 7 Cyklus na extrakciu dát z najnakup

Uloženie kladných a záporných komentárov do Dočasného úložiska - zoznam kladných komentárov bolo uložených vo vektore, avšak keď sme chceli ukladať dáta do databázy, museli sme transformovať premennú "plusy" na dátový typ data frame. Následne sme si vytvorili stĺpce, ktoré budú v tabuľke a definujeme ich typy. Na uloženie tabuľky v MS SQL sme si vygenerovali jej názov pozostávajúci z idObchodu, typu komentov a dnešného dátumu. Použitím príkazu "sqlSave" tabuľka bola uložená do databázy.

```
> nazov_tabulky <- paste(id,1,gsub("-","",Sys.Date()),sep="_")
> plusy <- data.frame(plusy)
> typy_stlpcov <- list(idKomentu="int", komentar="NVarchar(MAX)")
> sqlSave(db_DocasneUlozisko, plusy,tablename = nazov_tabulky, colnames=
c("idKomentu","komentar") ,varTypes = typy_stlpcov)
```

Fragment 8 Uloženie kladných komentárov do Dočasného úložiska

Dáta po tejto fáze boli uložené v Dočasnom úložisku a stále aj v R-Studio, kde boli pripravené na druhú časť Transformáciu

4.2.2 Transformácia

Podľa Obr. 11 sme sa najskôr zamerali na Testovaciu fázu, po ktorej by sme mali mať k dispozícii definovaný zoznam techník na čistenie dát, zoznamy slov bez informačnej hodnoty a sledované charakteristiky. Postupným testovaním techník na čistenie dát sme dospeli k záveru, že najlepšie je použiť techniky v tomto poradí, Fragment 9.

```
> komenty <-  chartr("áäÁÄ","aaaa",komenty)
> komenty <-  chartr("éÉ","ee",komenty)
> komenty <-  chartr("íÍ","ii",komenty)
> komenty <-  chartr("ĺľĹĽ","llll",komenty)
> komenty <-  chartr("óôÓÔ","oooo",komenty)
> komenty <-  chartr("ŕŔ","rr",komenty)
> komenty <-  chartr("úÚ","uu",komenty)
> komenty <-  chartr("ýÝ","yy",komenty)
> komenty <-  chartr("čČ","cc",komenty)
> komenty <-  chartr("ďĎ","dd",komenty)
> komenty <-  chartr("ňŇ","nn",komenty)
> komenty <-  chartr("šŠ","ss",komenty)
> komenty <-  chartr("ťŤ","tt",komenty)
> komenty <-  chartr("žŽ","zz",komenty)
> komenty <- iconv(komenty, "UTF-8", "ASCII", sub = "")
> komenty <- removePunctuation(komenty)
> komenty <- removeNumbers(komenty)
> komenty <- tolower(komenty)
> komenty <- removeWords(komenty, stopwords_)
> komenty <- stripWhitespace(komenty)
```

Fragment 9 Techniky na čistenie dát

V testovacej fáze sme pracovali s množinou 250 000 komentárov, uložených v premennej "komenty". Najskôr funkciou "chartr" sme nahradili dlhé samohlásky a jednu dvojhlásku (ô) za krátke samohlásky a mäkké spoluhlásky za tvrdé. Pomocou funkcie "iconv" sme odstránili z komentárov znaky, ktorými sa tvoria rôzne grafické vizualizácie v textoch a sú často využívané užívateľmi na webových stránkach. Rozhodli sme sa najskôr použiť funkciou "chartr", nakoľko funkcia "iconv" odstraňuje zo slov aj mäkké a tvrdé hlásky, čím by sme stratili význam slov. Pomocou ďalších dvoch funkcií "removePunctuations" a "removeNumbers" sme odstránili interpunkčné znamienka a čísla.

Funkciou “tolower“ sme upravili komentáre do jedného formátu, zmenením všetkých veľkých písmen na malé. Z komentárov funkcia “removeWords“ odstránila slová definované v premennej “stopwords_“ a použitím funkcie “stripWhitespace“ sme odstránili prázdne miesta medzi slovami v komentároch, ktoré zostali po odstránení slov v predchádzajúcom kroku.

Očistené dáta boli pripravené vo formáte vhodnom pre analyzy. Najskôr komentáre z dátovej štruktúry vektor sme vložili do korpusu, kde každý komentár bol uložený ako dokument. Definovaním a použitím funkcie “tokenizer“ sme extrahovali z jednotlivých dokumentov korpusu jedno a dvojslovné pomenovania a výsledky funkcie sme uložili do premennej “komenty_tdm“ typu TermDocumentMatrix. V tejto premennej bol uvedený počet slov alebo slovných spojení v jednotlivých dokumentoch korpusu. Pomocou funkcie “findFreqTerms“ sme analyzovali výsledky z predošlých aktivít.

```
> komenty_corpus <- VCorpus(VectorSource(komenty))
> tokenizer <- function(x)
+ { NGramTokenizer(x, Weka_control(min = 1, max = 2)) }
>
> komenty_corpus <- VCorpus(VectorSource(komenty))
> tokenizer <- function(x)
+ { NGramTokenizer(x, Weka_control(min = 1, max = 2)) }
> komenty_tdm <- TermDocumentMatrix(komenty_corpus, control = list(tokenize = tokenizer))
> findFreqTerms(komenty_tdm,3000)
```

Fragment 10 Proces extrakcie slov z komentárov

Analyzovaním slov a dvojslovných spojení sme rozšírili zoznam slov bez informačnej hodnoty o ďalšie slová, ktoré sú využívané v recenziách užívateľmi, ako napríklad: na, neviem, atď. Zoznam sme rozšírili o 19 slov, ktoré budú využité v Aplikačnej časti. Analýzou sme tiež získali charakteristiky, ktoré pomenúvajú kvalitu tovarov alebo služieb obchodu. Najskôr sme si jednotlivé slová vypísali, ktoré sme chceli sledovať a každé z nich zastupuje jednu skupinu. Po skončení získavania slov, sme mali k dispozícii zoznam, kde každé slovo zastupuje jednu skupinu. Nakoľko v zozname sme mali slová so spoločným významom, tieto slová boli zoskupené do jednej skupiny. Skupiny boli rozšírene ešte o ďalšie slova s rovnakým významom, pomocou online synonymického slovníka [34].

Tab. 9 Zoznam prvých 10 skupín charakteristík

id_skupiny	zaklCharakteristika	chrakteristiky
1	cena	cena
2	promptnost	pohotovost,promptnost
3	personal	personal,zamestnanec,zamestnanc
4	web	stranka,eshop,web
5	ponuka	ponuka,návrh,vyber
6	ochota	ochota, pozornost,obetavost,serioznost
7	posta	posta,postovne
8	tovar	tovar,komodita, produkt,sortiment
9	prehladnost	prehladnost,informacnost
10	dodanie	dodanie,dorucenie,distribucia

Obidva zoznamy sme uložili do databázy Monitoring, kde budú vo forme tabuliek k dispozícii a môžeme ich kedykoľvek aktualizovať.

V Aplikačnej fáze boli dispozícii kladné a záporné komentáre, s ktorými sme pracovali separátne v závislosti od toho, na aký obchod sa viazali. Najskôr sme použili techniky na čistenie dát, Fragment 9. Z komentárov boli extrahované slová, ktoré sme uložili do premennej typu TermDocumentMatrix, do ktorej sa ukladali aj údaje o počte výskytov jednotlivých slov v textoch. Použili sme rovnaké príkazy ako vo Fragment 10, avšak funkcia "tokenizer" vyhľadala len slová, nie dvojslovné pomenovania. Slová, ktoré sa vyskytli v kladných komentároch resp. záporných komentároch sme ukladali do premennej "plusy_tdm" resp. "minusy_tdm". Funkciou "rowSums" sme sčítali celkový výskyt jednotlivých slov v kladných a záporných komentároch, ktoré predtým boli uložené do dátového typu matrix, pre lepšiu prácu z dátami a použitie funkcie "rowSums".

```
> plusy_m <- as.matrix(plusy_tdm)
> plusy_pocet <- rowSums(plusy_m)
> minusy_m <- as.matrix(minusy_tdm)
> minusy_pocet <- rowSums(minusy_m)
```

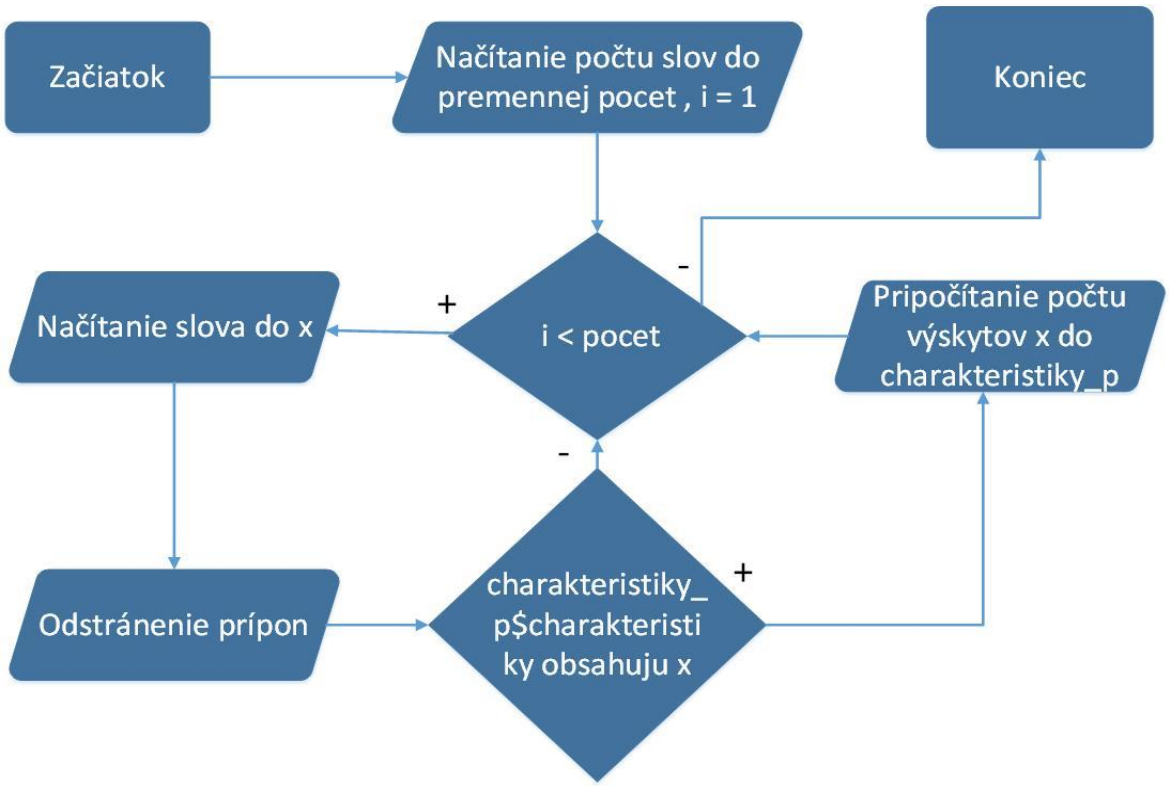
Fragment 11 Výpočet výskytu jednotlivých slov v kladných a záporných recenziách

V R-studiu sme vytvorili premenné "charakteristiky_p" a "charakteristiky_m" typu data frame, ktoré sa skladali z troch stĺpcov:

- zaklad – hlavné slovo charakterizujúce skupinu slov,

- charakteristiky – slovo alebo skupina slov s rovnakým významom pomenúvajúce určitý tovar alebo službu obchodu,
- vyskyt – počet výskytov charakteristík v komentároch, na začiatku sú v tomto stĺpci hodnoty na 0.

Do posledného stĺpca vyskyt sme sa rozhodli pripočítavať počty výskytov jednotlivých slov, na základe toho, v ktorej skupine boli definované. Predtým sme však museli ešte zo slov odstrániť prípony, aby sme boli schopní automatizovane slová priraďovať k jednoslovným definovaným charakteristikám. Rozhodli sme sa neodstraňovať predpony, nakoľko odstránením predpony zo slov by sme zmenili význam slova [35]. Proces odstránenia prípon zo slov a výpočtu výskytov jednotlivých charakteristík zobrazíme na vývojovom diagrame, Obr. 15.



Obr. 15 Výpočet počtu výskytov sledovaných charakteristík

V tomto odstavci vysvetlíme časti kódu z R-Studia, ktorými sme realizovali niektoré aktivity z vývojového diagramu na Obr. 15:

- Odstránenie prípon – bolo realizované na základe podmienok, ktoré boli vyhodnocvané prostredníctvom funkcie “substr“, aká prípona je v slove použitá. Ak jedna z podmienok

rozozná príponu, následne je odstránená zo slova. Vo Fragment 12 je zobrazená časť kódu, pomocou ktorej sme odstránili zo slov prípony pozostávajúce zo štyroch písmen.

```
> if (str_sub(x,-4) == "ence" || str_sub(x,-4) == "tami" || str_sub(x,-4) == "tach"
|| str_sub(x,-4) == "iami" || str_sub(x,-4) == "iach" || str_sub(x,-4) == "ovia" ||
str_sub(x,-4) == "tach")
+ {x <- str_sub(x,0,-5) }
```

Fragment 12 Odstránenie prípon zo slov

- Podmienka, na overenie, či slovo patrí do niektorej zo skupín – na overenie, či dané slovo patrí do určitej skupiny sme použili funkciu "grep", ktorá ak našla zhodu slova a určitej charakteristiky v skupine, vrátila hodnotu 1 a ak nenašla zhodu, vrátila číslo 0. Pred vykonaním podmienky sme vytvorili premennú "slovo", ktorá pozostávala z vyhľadávacieho znaku „^" a základu slova uloženého v premennej "x".

```
> slovo <- paste("^" , x , sep = "")
  if  (length (charakteristiky_p[grep(slovo,charakteristiky_p$charakter
  istiky), 1] )  !=  0 )
```

Fragment 13 Podmienka na overenie, či dané slovo sa nachádza v množine definovaných charakteristík

- Pripočítanie počtu výskytov slova do priradenej skupiny charakteristík – do premennej "zaklad_syn" sme vložili základné slovo reprezentujúce skupinu charakteristík, na základe toho, ku ktorej skupine bolo extrahované slovo z komentárov priradené. Do premennej "pocet_vyskytov" sme uložili hodnotu, koľkokrát sme dané slovo našli v komentároch. V poslednom kroku sme aktualizovali hodnotu stĺpca výskyt pre vybratú skupinu sčítaním aktuálnej hodnoty v stĺpci a hodnoty uloženej v premennej "pocet_vyskytov".

```
> zaklad_syn <- charakteristiky_p$zaklCharakteristika[grep (slovo,charakteristiky_p$ charakteristiky)]
  pocet_vyskytov <- plusy_pocet[i]
  charakteristiky_p$vyskyt [charakteristiky_p$zaklCharakteristika  == zaklad_syn] <- pocet_vyskytov +
  charakteristiky_p$vyskyt [charakteristiky_p$zaklCharakteristika  == zaklad_syn ]
```

Fragment 14 Priraďovanie slova do skupín charakteristík a aktualizovanie počtu výskytov pre danú skupinu

Rovnako boli spracované slová aj z negatívnych komentárov, avšak výsledky sme ukladali do inej premennej. Výsledky päť skupín pre kladné komentáre sú na Obr. 16.

	zaklad	charakteristiky	vyskyt
1	cena	cena	827
2	komunikacia	komunikacia,dorozumievanie,jednanie	320
3	promptnost	pohotovost,promptnost	3
4	dodanie	dodanie,dorucenie,distribucia	440
5	tovar	tovar,komodita, produkt,sortiment	556

Obr. 16 Výsledok prvých 5 skupín charakteristík pre obchod DOMOSS

Výpočty charakteristík pre kladné aj záporné komentáre boli vykonané aj pre ďalšie dva obchody, ktoré boli uložené do databázy.

Formátovanie dát pre uloženie do databázy nevyžadovali od nás použitie veľa techník, nakoľko okrem komentárov stiahnutých z webov, sme pracovali z dátami, ktoré boli stiahnuté priamo z databázy. Výsledky spracovania neštruktúrovaných dát sme uložili do tabuľky "PocetVyskytov" uloženú v databáze Monitoring. Táto tabuľka pozostávala z piatich stĺpcov: idObchodu, idSkupiny, idHodnotenia, pocet,datum. Pre prvé tri stĺpce máli hodnoty uložené v premenných a tieto hodnoty museli byť celočíselné, keďže tieto stĺpce boli predtým pri implementácii definované typom integer. Štvrtý stĺpec v tabuľke bol tiež definovaný ako celočíselný a počet výskytov sme exportovali z premenných "charakteristiky_p" (pre negatívne komentáre "charakteristiky_m"), zo stĺpca vyskyt. Posledný údaj musel pozostávať z dátumu, ktorý sme vygenerovali prostredníctvom funkcie "Sys.Date()" v RStudio.

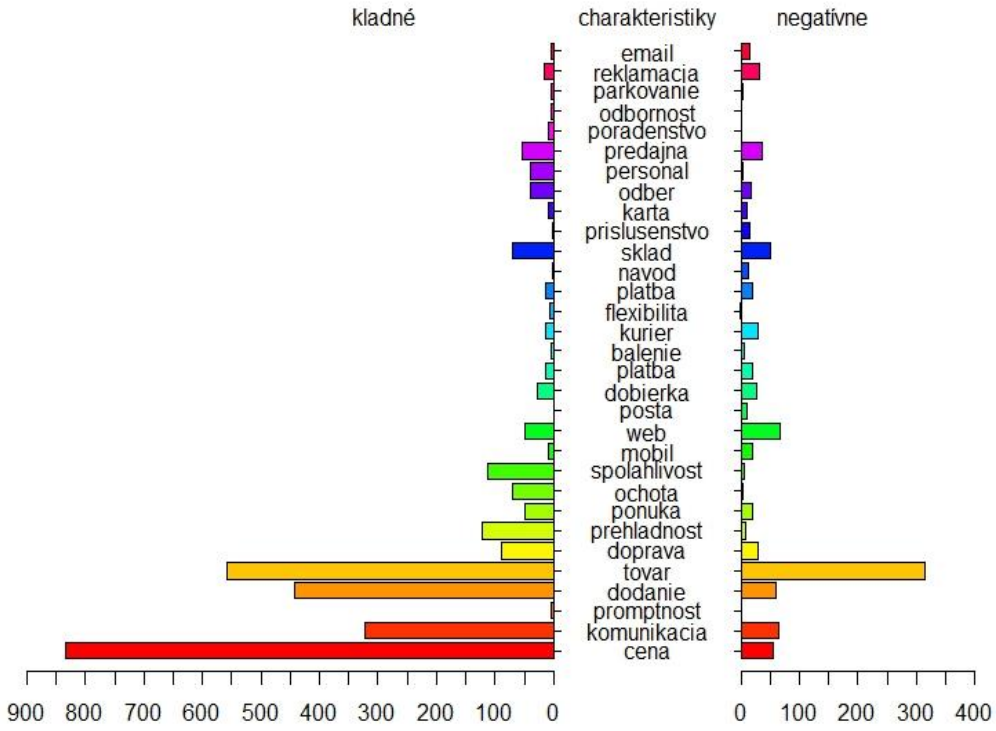
4.2.3 Uchovávanie

Pre uloženie výsledkov do tabuľky "PocetVyskytov" v databáze Monitoring sme najskôr museli výsledky uložiť do Dočasného úložiska rovnakým spôsobom ako v časti Extrakcia. Jediný rozdiel bol v spôsobe definovania tabuliek. Nemohli sme mať v rámci jednej databázy dve tabuľky s rovnakým názvom, preto názvy tabuliek museli pozostávať aj zo slova "Vysledky". Následne sme v MS SQL vytvorili procedúru, ktorá použitím príkazu "insert into" uložila dáta do existujúcej tabuľky "PocetVyskytov".

Výsledky sledovaných charakteristík pre vybrané obchody a informácie o týchto obchodoch boli k dispozícii v databáze Monitoring. Pre ďalšie analýzy bolo možné použiť

reporting-ový nástroj, ktorým je možné získať dáta z MS SQL a prostredníctvom tohto nástroja vizualizovať skúmané dáta. RStudio nám tiež ponúka veľké množstvo možností na vizualizáciu dát. Cez vytvorené spojenie medzi prostrediami sme boli schopní získať dáta z databázy Monitoring.

Pomocou grafov sme sa najskôr pokúsili zobraziť porovnanie počtu jednotlivých charakteristík pre obchod DOMOSS v rámci kladných aj negatívnych komentárov. Pre potrebu tejto analýzy sme potrebovali názvy základných charakteristík a hodnotu ich výskytov v kladných aj negatívnych komentárov. Dáta sme stiahli funkciou "sqlQuery" (Fragment 1). Bolo potrebné použiť taký typ grafov, v ktorom dokážeme zobraziť dva druhy číselných údajov, ktoré patria jednotlivým charakteristikám. Použitím pyramídového grafu sme boli schopní tieto požiadavky splniť.

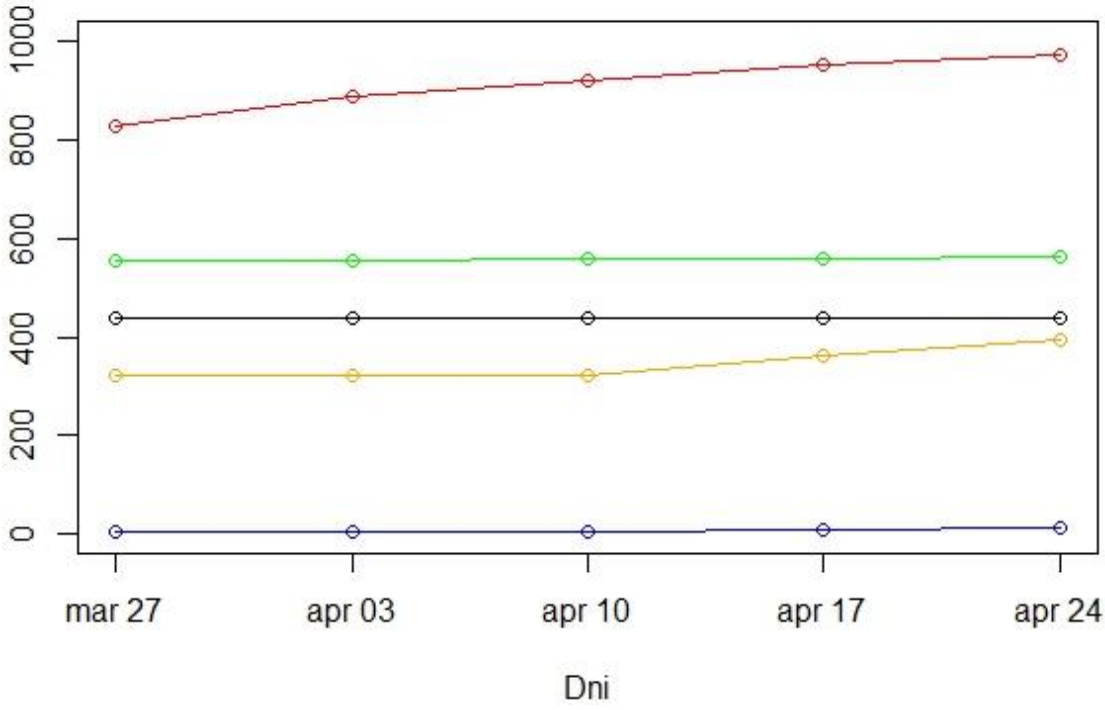


Obr. 17 Pyramídový graf zobrazujúci porovnanie počtu výskytov jednotlivých charakteristík v kladných a negatívnych komentárov

Na vytvorenie grafu sme použili funkciu "pyramid.plot". Na ľavej strane je graficky v podobe farebných stĺpcov zobrazený počet výskytov charakteristík v kladných komentároch a na pravej strane v negatívnych komentároch. Medzi stĺpcami sú zobrazené jednotlivé charakteristiky. Pomocou tohto typu grafu a dát zobrazených v ňom dokáže

manažér alebo vedúci pracovník obchodu zistiť, v čom by sa firma mala zlepšiť a z akými službami sú ich zákazníci spokojní.

Zámerom ďalšej vizualizácie bolo zachytiť vývoj päť charakteristík (cena, promptnosť, tovar, dodanie a komunikácia) v čase. Vyberali sme graf, v ktorom je možné zobrazovať časový vývoj hodnôt (počet výskytov) pre jednotlivé premenné (charakteristiky). Rozhodli sme sa pre čiarový typ grafu, v ktorom je možné zobraziť viacero čiar.



Obr. 18 Spojnicový graf zachytávajúci vývoj počtu výskytov jednotlivých charakteristík v čase

Na osi x sú zobrazené dni, kedy boli údaje zachytené a na osi y zase počty výskytov charakteristík. Každej charakteristike v grafe patrí jedna čiara (cena = červená, promptnosť = čierna, tovar = zelená, dodanie = čierna, komunikácia = oranžová). Tento typ grafu môže byť užitočný v situáciách, kedy sa obchod rozhodne zlepšiť určitú službu. Prostredníctvom tohto grafu je možné zistiť, či zavedenie nových opatrení alebo inovácii spôsobilo u zákazníkov väčšiu spokojnosť v tejto oblasti.

5 Diskusia

V práci sme sa zamerali najmä na neštruktúrované dáta týkajúce sa recenzií užívateľov na internetových stránkach. Náš navrhnutý model sme aplikovali na úlohu, ktorej cieľom bola extrakcia recenzií z dôveryhodných webových sídiel, transformácia týchto recenzií do formy, aby nám poskytovali informácie zachytávajúce kvalitné a slabé stránky obchodov a zabezpečiť uloženie dát.

Použili sme na implementáciu modelu dve vývojové prostredia, ktoré dokážu medzi sebou posielať dáta. RStudio sa ukázalo ako prostredie, v ktorom dokážeme použiť rôzne techniky a metódy na prácu s rôznymi typmi dát. MS SQL sme použili na implementáciu dátového modelu a bezpečné ukladanie dát. V rámci prvej fázy modely sme neimplementovali žiadne riešenia, ale zamerali sme sa na definíciu cieľov a funkcionalít, a potom sme prešli k návrhu a implementácii databázy. Pri implementácii modelu sme vyžili návrhy a analýzy prvej fázy, čo nám potvrdilo dôležitosť tejto fázy a skutočnosť, že až po úplnom dokončení prvej fázy by sa malo prejsť na druhú fázu modelu, ktorou je Implementácia.

Druhá fáza modelu pozostávala najskôr z extrakcie dát z vybratých a analyzovaných webových sídiel. Osvedčila sa nám použitá technika Xpath na extrakciu dát z webových stránok. Rozhodnutím navrhnúť a implementovať Dočasné úložisko sme sa snažili zamedziť, aby v prípade nečakanej chybe došlo k strate dát a opätovne by sme museli dáta sťahovať z webových stránok. Zaoberali sme sa myšlienkou, že Dočasné úložisko nebudeme implementovať, nakoľko recenzie je možné hocikedy stiahnuť z webových stránok. Avšak sme sa rozhodli ukladať stiahnuté dáta do Dočasného úložiska, pretože je rýchlejšie získať dáta z tohto zdroja, než sa pripojiť na stránku a opätovne sťahovať dáta. Vo fáze Transformácia sme najskôr spracovali tieto dáta. Rozhodli sme sa najskôr zaviesť Testovaciu fázu, v ktorej sme vyskúšali rôzne Text Mining-ové techniky a metódy a vybrali tie, ktoré budú užitočné pre našu úlohu. V Aplikačnej fáze sme použili tieto techniky už na dátach, ktoré sa týkali len jedného obchodu a boli jedného typu (kladné alebo záporné). Na konci tejto fázy sme priraďovali jednotlivé slová do skupín pozostávajúcich zo slov s rovnakým významom. Extrahované slová z komentárov boli rôznych pádoch a stáli sme pred otázkou, či budeme odstraňovať prípony zo slov alebo vytvoríme zoznam, v ktorom každému slovu v základnom tvare budú priradené formy tohto slova v rôznych pádoch. Rozhodli sme sa pre prvú možnosť, ktorá nám prišla rýchlejšia a dosahovali sme v nej vysokú úspešnosť priradenia. Výsledky sme v poslednej fáze uložili do implementovanej

relačnej databázy, spôsobom aby boli zachované predom definované vzťahy medzi tabuľkami a stĺpcami vo výslednej tabuľke. Uložené dáta sme následne vizualizovali na dvoch úlohách, ako by mohli byť dáta spracované v modeli využité.

Navrhované a implementované riešenie by bolo možné rozšíriť o ďalšie funkcionality, ktorými by sme mohli napríklad zvýšiť počet dát, s ktorými sa pracuje v modeli. Na webovom sídle heureky sú okrem kladných a negatívnych komentárov ešte komentáre, ktoré nie sú zaradené do týchto dvoch kategórii. My sme sa rozhodli tieto komentáre neextrahovať z webových stránok. Avšak aj tieto komentáre by bolo možné zatriediť do týchto dvoch kategórií prostredníctvom automatu, ktorý by vedel tieto texty triediť pomocou Text Mining-ových metód. Automat by mohol najskôr použiť Text Mining-ové techniky na očistenie dát, ktoré sme aj my využili v procese spracovania neštruktúrovaných dát a následne výberom správnej Text Mining-ovej metódy zaraďovali tieto komentáre do jednej z dvoch kategórii. Rovnakým spôsobom by sme mohli získavať dáta aj zo sociálnych sieti, kde je tiež možné pridať recenziu na obchody. Tu však okrem použitia automatu na rozoznávanie komentárov by bolo nutné dostať schválenie od jednotlivých majiteľov obchodov, ktorí vlastnia stránky na sociálnych sieťach. Bez súhlasu majiteľ neobdŕžania prihlasovacích údajov nie je možné tieto dáta extrahovať. Dáta sme ukladali do tabuliek, ktoré sú súčasťou relačných databáz. V modely sme navrhli a implementovali dve relačné databázy. Bolo by zaujímavé sledovať, či pri väčšom množstve dát, by bolo možné s databázami pracovať rýchlo a efektívne. V prvej kapitole sme definovali okrem relačných databáz, aj NOSQL prístup, ktorý by bolo možné použiť v tejto situácii. V prípade objavenia problému súvisiaceho s množstvom, by mohla časť dát, napríklad dáta ukladané do Dočasného úložiska a výsledky spracovania neštruktúrovaných dát (výsledky počtu výskytov jednotlivých skupín charakteristík), byť uložená do formátu JSON alebo XML. Bolo by zaujímavé zistiť či by sa zmenšil alebo odstránil problém s veľkým počtom dát a či by bolo možné použiť dva rozdielne prístupy uchovávania dát.

Výsledky modelu poskytujú žiadané informácie pre obchody, z ktorých je možé zistiť, s čím zákazníci boli spokojný a čo zákazníkom prekáža na obchode. Ak obchod zavedie zlepšenia, môže prostredníctvom dát z modelu zistiť, či zmeny pomohli odstrániť tieto nedostatky alebo vnímané problémy u zákazníkov pretrváva.

Záver

V diplomovej práci sme sa zamerali na návrh a implementáciu modelu na spracovanie, triedenie a analyzovanie štruktúrovaných aj neštruktúrovaných dát z webového prostredia.

Prvá kapitola pozostáva z analýz možností, aké dáta môžeme extrahovať z webového prostredia, ako môžeme spracovať neštruktúrované dáta a následne, ako tieto dáta uložiť. Existuje mnoho prístupov, ktoré sa zameriavajú na prístup s neštruktúrovanými dátami. My sme sa v práci zamerali na definovanie a použitie Text Mining-u. V závere prvej kapitoly sme vymedzili v pojmy v oblasti relačných databáz a NOSQL, ktorý poskytuje alternatívu k tradičným databázam.

V tretej kapitole sme navrhli všeobecne model, ktorým by sme mali byť schopní extrahovať, triediť a analyzovať rôzne typy dát. Na začiatku sme definovali základy modelu, z akých fáz by mal pozostávať. Potom sme definovali podrobnejšie jednotlivé fázy modelu.

V záverečnej kapitole sme implementovali navrhnutý model na riešenie konkrétneho problému, ktorý súvisel s extrakciou recenzií z webových sídiel, spracovaním neštruktúrovaných dát a uložením výsledkov relačných databáz. Z výsledkov vyplynulo, že model by bolo možné použiť na takéto typy úloh. Výsledkom modelu pre toto riešenie bol zoznam slov charakterizujúce určitý tovar, službu alebo atribút obchodu, s ktorými sa zákazníci obchodu stretávajú.

Zoznam použitej literatúry

[1] Analýza velkých dat. [online]. [cit. 5.1. 2017]. Dostupné na internete: <https://www.gaussalgo.cz/analyza-velkych-dat/>.

[2] Herschel, Richard T., and Nory E. Jones. 2005. Knowledge management and business intelligence: The importance of integration. Journal of Knowledge Management1367-3270 (Aug 1,)9: 45 – 55.

[3] W3Techs - Web Technology Surveys. W3Techs - world wide web technology surveys. [online]. [cit.7.1. 2017]. Dostupné na internete: <https://w3techs.com>.

[4] Prof. PhDr. Soňa Makulová, PhD. 2010. Informačná architektúra sieťových informačných zdrojov a médií. Bratislava: ELET, s.r.o.978-80-88812-21-0.

[5] Mencl Michal. Základní kurz 4: Základy syntaxe. [online]. [cit.8.1. 2017]. Dostupné na internete: <http://www.pehapko.cz/zakladni-kurz/4-zaklady-syntaxe>.

[6] Srivastava, T., P. Desikan, and V. Kumar. 2005. Web mining – concepts, applications and research directions. In Foundations and advances in data mining. Vol. 180. Berlin, Heidelberg: Springer Berlin Heidelberg, 275-3073540250573.

[7] Srivastava Jaideep. Web mining : Accomplishments & future directions. [online]. [cit.28.1. 2017]. Dostupné na internete: <http://ieeexplore.ieee.org/document/5452608>.

[8] Ferrara, Emilio, Pasquale De Meo, Giacomo Fiumara, and Robert Baumgartner. 2014. Web data extraction, applications and techniques: A survey. Knowledge-Based Systems0950-7051 (Nov 1,)70: 301-23. <http://arxiv.org/abs/1207.0246>.

[9] Mazal, Zdeněk, Ondřej Morský, and Lucie Fojtová. 2011. Extrakce textových dat z internetových stránek.Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. [online]. [cit.28.1. 2017]. Dostupné na internete: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=40352>.

[10] Firat, Aykut, Stuart Madnick, Nor Yahaya, Choo Kuan, and Stéphane Bressan. 2005. Information aggregation using the caméléon# web wrapper. In E-commerce and web technologies. Vol. 3590. Berlin, Heidelberg: Springer Berlin Heidelberg, 76-869783540284673.

[11] W3C. XML path language (XPath) version 1.0. [online]. [cit.30.1. 2017]. Dostupné na internete: <https://www.w3.org/TR/xpath/>,

[12] Divéky Marko. 2009. Triedenie a zoraďovanie (HITS). [online]. [cit.31.1. 2017]. Dostupné na internete: <http://vi.ikt.ui.sav.sk/@api/deki/files/466/=projekt.pdf>.

[13] Kopáčková, Hana, and Máchová, Renáta. 2006. Manažérské rozhodování za využití metód pro zpracování dokumentů. [online]. [cit.5.2. 2017]. Dostupné na internete: <http://hdl.handle.net/10195/35140>

[14] Kolek Stefan, a Kirmaci Oezkan. 2006. Web mining& clickstream analysis University of Fribourg (31.05.). [online]. [cit.7.2. 2017]. Dostupné na internete: <http://diuf.unifr.ch/is/studentprojects/pdf/reports/CRM_SS06_Web_Mining_And_Clickstream_Analysis_(StefanKolek_OezkanKirmaci).pdf>

[15] Hotho Andreas, Nurnberger Andreas, and Paaß Gerhard. 2005. A brief survey of text mining (May 13,). [online]. [cit.8.2. 2017]. Dostupné na internete: <http://www.kde.cs.uni-kassel.de/hotho/pub/2005/hotho05TextMining.pdf>.

[16] Jovanovic Jelena. Introduction to text mining. [online]. [cit.10.2. 2017]. Dostupné na internete: <http://ai.fon.bg.ac.rs/wp-content/uploads/2015/04/Intro-to-TM.pdf>.

[17] Paralič Ján 2003: Objavovanie znalostí v databázach. Košice: Elfa, 2003.ISBN 80-89066-60-7.

[18] Ing. Matúš Jurečka PhD. Metódy rozpoznávania reči. [online]. [cit.12.2. 2017]. Dostupné na internete: <http://frtk.fri.uniza.sk/jurecka/hmm.pdf>.

[19] Modelování databází. [online]. [cit.17.2. 2017]. Dostupné na internete: <https://www.root.cz/clanky/modelovani-databazi/>.

[20] Stephens Ryan, Plew Ron, Jones D. Arie. Brno: Computer Press,a.s., 2010. ISBN 978-80-251-2700-1.

[21] Johnstone High School. Data anomalies. [online]. [cit.27.2. 2017]. Dostupné na internete: <http://www.jhigh.co.uk/Higher/dbases/anomalies.html>.

[22] Rouse Margaret. NoSQL (not only SQL database). [online]. [cit.2.3. 2017]. Dostupné na internete: <http://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>.

[23] Why NoSQL database? [online]. [cit.5.3. 2017]. Dostupné na internete: <https://www.couchbase.com/nosql-resources/why-nosql>.

[24] R-manual.[online]. [cit.7.3. 2017]. Dostupné na internete: <http://rmanual.fri.uniza.sk/?page_id=20>.

[25] What is python? executive summary. [online]. [cit.7.3. 2017]. Dostupné na <internete:https://www.python.org/doc/essays/blurb/>.

[26] Ashutosh, K. S. 10 web scraping tools to extract online data. [online]. [cit.15.3.2017]. Dostupné na internete: <http://www.hongkiat.com/blog/web-scraping-tools/>.

[27] Laberge, Robert. 2011. Dátove sklady: Agilní metódy a business intelligence. New York [u.a.]: McGraw-Hill0071745327. ISBN 978-80-251-3729-1.

[28] DB-Engines. DB-engines ranking. [online]. [cit.16.3. 2017]. Dostupné na internete: <http://db-engines.com/en/ranking>.

[29] Čermák Miroslav. Vícevrstvá architektura: Tenký, tlustý a chytrý klient. [online]. [cit.16.3. 2017]. Dostupné na internete: <http://www.cleverandsmart.cz/vicevrstva-architektura-tenky-tlusty-a-chytry-klient/>.

[30] [online]. [cit.20.3. 2017]. Dostupné na internete: <https://obchody.heureka.sk>.

[31] [online]. [cit.21.3. 2017]. Dostupné na internete: <https://obchody.heureka.sk/mall-sk/recenze/?f=4>.

[32] [online]. [cit.21.3. 2017]. Dostupné na internete: <https://www.najnakup.sk/mall-sk>.

[33] Zoznam stop slov. 2016. [online]. [cit.22.3. 2017]. Dostupné na internete: <http://text.fiit.stuba.sk/zoznam_stop_slov.php#focus>.

[34] Synonymicky slovník slovenčiny. 2017. [online]. [cit.23.3. 2017]. Dostupné na internete: <http://www.juls.savba.sk/synonymicky_slovnik.html>.

[35] Predpona. 2015. [online]. [cit.23.3. 2017]. Dostupné na internete: <http://dai.fmph.uniba.sk/~filit/fvp/predpona.html>.

Zoznam príloh

DVD obsahuje nasledovné skripty a jeden Excel súbor:

- databazy,
- extrakcia,
- testovaciaFaza,
- aplikacnaFaza,
- stopwords_slova,
- vizualizacia.