Fulltextové vyhľadávanie so štruktúrovaným výsledkom

Patrik Sedlák

3Ib, 2016 - 2017

Abstrakt. Práca sa zaoberá návrhom a vytvorením metódy fulltextového vyhľadávania nad produktmi internetových obchodov, ktorej výsledkom je okrem zoznamu produktov, ktoré najviac zodpovedajú vstupnému dopytu, aj štruktúrovaná informácia o úspešnosti vyhľadávania podľa domén a atribútov. Navrhnutá metóda má byť následne implementovaná ako modul v projekte Kapsa.

Kľúčové slová: fulltextové vyhľadávanie, štruktúrované dáta, Elasticsearch, internetový obchod, Kapsa, Wicket, JSON, expandovanie dopytu, query

1 Úvod

V dnešnej uponáhľanej dobe azda niet človeka, ktorý by si nerád urýchlil a uľahčil činnosti každodenného života. Medzi takéto činnosti patrí aj nakupovanie, ktoré vďaka Internetu nadobudlo novú podobu. Prostredníctvom internetových obchodov si už dnes vieme zakúpiť množstvo vecí a tak, keď človek hľadá na webe produkty pre seba, môže sa v tej záplave internetových obchodov a rôznych produktov strácať.

Naša práca sa preto zaoberá návrhom takej vyhľadávacej metódy, ktorá zákazníkovi internetového obchodu uľahčí vyhľadávanie medzi produktmi a poskytne mu aj informáciu o tom, ktoré produkty najviac vyhovujú jeho dopytu. Súčasne chceme, aby zákazník tiež o produkte vedel, ku akej doméne (elektronika, tablety, bicykle, ...) patrí, a pri ktorých atribútoch sa našla nejaká zhoda s dopytom.

Cieľom našej práce je vytvoriť vyhľadávanie nad viacerými atribútmi a kategóriami produktov, pričom výsledky tohto vyhľadávania budú štruktúrované a tiež utriedené podľa toho, na koľko vyhovujú dopytu.

1.1 Fulltextové vyhľadávanie

Pri fulltextovom vyhľadávaní vyhľadávací engin kontroluje všetky slová uložené v danom indexe a snaží sa nájsť zhodu s vyhľadávacími kritériami, ktoré boli zadané používateľom (takýmto kritériom môže byť napríklad text zadaný používateľom). Fulltextové vyhľadávanie v minulom storočí našlo využitie hlavne v knižných

databázach, v súčasnosti ponúka možnosť fulltextového vyhľadávania množstvo webových stránok či aplikácií.

V prípade, že fulltextové vyhľadávanie využívame pri malom množstve dokumentov, je možné priamo porovnať obsah dokumentov s dopytom. Avšak ak je množina dokumentov na prehľadávanie potenciálne veľká, je dobré vyhľadávanie rozdeliť na dve úlohy, a to indexovanie a vyhľadávanie. Fulltextové vyhľadávače pracujú s takou podobou dokumentov, akú mali v čase posledného indexovania. Index je preto potrebné po istom čase aktualizovať, no aktualizácia môže byť pri väčšom objeme dát časovo náročná a preto je potrebné dobre si premyslieť kedy aktualizovať. Keďže Kapsa indexuje produkty z internetových obchodov, tak v tomto prípade ide o veľké množstvo produktov, ktoré je potrebné indexovať, avšak platí, že počet indexovaných produktov je menší, ako počet extrahovaných produktov, pretože sa dá ľahko predpokladať, že niektoré produkty sú ponúkané viacerými obchodmi.

Keď hovoríme o fulltextovom vyhľadávaní, je dôležité spomenúť aj jeho vlastnosti, ktorými sú presnosť a úplnosť vyhľadávania. Úplnosť popisuje kvantitu výsledkov, ktoré nám vyhľadávanie vráti. Presnosť na druhú stranu popisuje kvalitu vrátených výsledkov. Úplnosť určíme ako pomer vrátených relevantných výsledkov a všetkých relevantných výsledkov v indexe. Presnosť určíme ako pomer relevantných vrátených výsledkov a všetkých vrátených výsledkov.

1.2 Elasticsearch

Projekt Kapsa pre fulltextové vyhľadávanie využíva vyhľadávací engin Elastichsearch. Tento engin vie vyhľadávať nad dokumentmi vo formáte JSON. Preto je potrebné naše dáta najprv previesť do tohto formátu. Následne ako výsledok po vykonaní dopytu nám Elasticsearch opäť vracia štruktúrovaný výsledok v podobe JSON dokumentu, čo je výhodou v našej bakalárskej práci.

JSON je skutočne jednoduchý formát zrozumiteľný pre bežného človeka. Opíšme si teda tento formát jeho štruktúru v skratke. JSON je vo svojej podstate obyčajný text, čo z neho robí formát nezávislý od jazyka, keďže s textom vie pracovať každý programovací jazyk. Dáta v JSON sú písané ako dvojica kľúč (názov atribútu) a jeho hodnota oddelen0 dvojbodkou. Dáta sú od seba navzájom oddelené čiarkami a jednotlivé objekty sú ohraničené zloženými zátvorkami. JSON musí mať ako kľúč vždy string ohraničený úvodzovkami, zatiaľ čo hodnoty môžu byť typu string, číslo, boolean, objekt (JSON objekt), pole (to je ohraničené hranatými zátvorkami) alebo null. Objekt JSON môže mať napríklad takýto tvar:

```
{
      "meno" : "Jozef",
      "vek" : 25,
      "jazyky" : ["slovensky", "anglicky", "rusky"]
}
```

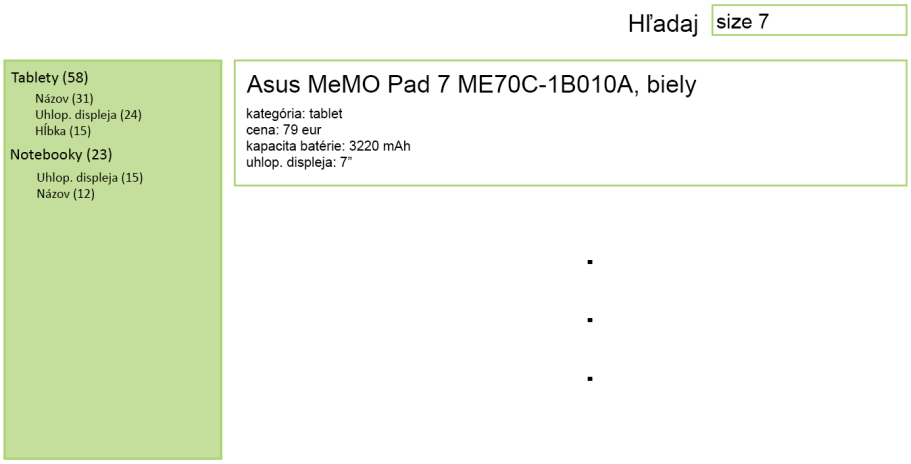
2 Praktická časť

2.1 Nedostatky vyhľadávania a možnosti ich riešenia

Problémom, s ktorým sa stretávame u rôznych internetových obchodov je, že pri zadaní nášho textového dopytu do vyhľadávača, sú nám vrátené produkty, ktoré sme na výstupe neočakávali, respektíve nechceli. Vezmime si ako príklad textový dopyt *„Samsung 7“ “*. Užívateľ očakáva, že mu vyhľadávanie vráti také zariadenia značky Samsung, ktoré majú veľkosť uhlopriečky sedem palcov. Zvyčajne však vyhľadávanie v internetových obchodoch vráti produkt, ktorý v názve obsahuje reťazce *„Samsung“* a *„7“*, pričom sa veľkosť displeja odignoruje.

Ďalším problémom, ktorý sa u internetových obchodov vyskytuje a v projekte Kapsa by sme sa mu chceli vyhnúť je, že ako výsledok vyhľadávania sú nám zobrazené len konkrétne produkty, a chýba nám informácia o tom, z ktorej kategórie (tablety, elektronika, bicykle, ...) sú tieto zobrazené produkty. Vráťme sa k nášmu dopytu *„Samsung 7“ “*. Po vykonaní tohto dopytu by sme chceli vidieť okrem konkrétnych produktov aj to, v akých kategóriách (doménach), prípadne podkategóriách (subdoménach) sú zaradené. Výsledky nášho dopytu by mohli obsahovať kategóriu *„elektronika“* a podkategóriu *„tablety“*. Ku kategórii by sa malo tiež zobraziť číslo, vyjadrujúce počet produktov vyhovujúcich dopytu v tej kategórii.

Zobrazovanie kategórií však nie je problém u všetkých internetových obchodov. Niektoré obchody tieto informácie zobrazujú. Čo by sme ale tiež uvítali ako užitočnú informáciu pre používateľa, zákazníka internetového obchodu, je dôvod, prečo nám vyhľadávanie vrátilo produkty, ktoré vrátilo. Teda ak po dopyte „Samsung 7“ “ sú mi vrátené produkty *Samsung Galaxy Tab A 7.0 WiFi čierny* a *Samsung Galaxy S7 biely*, tak o prvom z nich vieme, že bol vrátený preto, lebo zhoda sa našla v názve, vo výrobcovi (Samsung) ale aj vo veľkosti displeja (7 palcov). O druhom vieme, že zhoda bola v názve a výrobcovi. Atribúty spolu s početnosťou zhôd chceme teda rovnako ako domény zobraziť po vykonaní vyhľadávania. Výsledky vyhľadávania by preto mohli vyzerať tak, ako to vidíme na obrázku (Obr. 1)



Obr.1. Jednoduchý návrh možného výstupu vyhľadávania. V centrálnej časti vidíme konkrétny produkt nájdený vyhľadávaním, v zelenom paneli naľavo vidíme strom obsahujúci kategórie a atribúty spolu s početnosťou produktov, ktoré v danej kategórii a atribúte vyhovujú dopytu.

2.2 Dáta Kapsy v Elasticsearch

Dáta Kapsy, ktoré sú uložené v Elasticsearch majú formát JSON. Ako sme už spomínali v časti *1.2 Elasticsearch*, tak dáta v JSON sú písané ako dvojica kľúč a hodnota, pričom kľúč má vždy podobu stringu, zatiaľ čo hodnoty môžu byť viacerých typov. Naše kľúče, okrem rôznych identifikátorov (id_object, id_domain, id_source) sú domain, kde sa v hodnote nachádza pole hodnôt, ktoré opisujú príslušné domény a subdomény, ku ktorým bol produkt priradený (napr. tablety), ďalej sú to kľúče, ktoré majú tvar *"attribute_xy"* (namiesto xy sa v kľúči nachádza nejaké prirodzené číslo – identifikátor atribútu) a tieto kľúče zodpovedajú jednotlivým atribútom. Z kľúča *"attribute_4"* však nevieme veľmi dobre určiť, o aký atribút produktu ide. Na to aby sme ku kľúču vedeli priradiť konkrétny atribút, musíme sa pozrieť na tabuľku v MySQL databáze, ktorá obsahuje popis jednotlivých atribútov. Ku každému atribútu je v tejto tabuľke pridelené aj identifikačné číslo (viď. Obr.3 na konci tejto časti). Na základe tejto tabuľky teda už vieme, že ak v Elasticsearch vidíme pri produkte kľúč *"attribute_4"*, tak ide o atribút ceny produktu. Ku každému kľúču v Elasticsearch je samozrejme priradená hodnota (väčšinou ako string).

Ďalším kľúčom, ktorý môžeme vidieť v Elasticsearch, je *"content"*. Jeho hodnotou je dlhší text, ktorý obsahuje všetky dôležité slová súvisiace s daným produktom. V podstate sa tu nachádzajú názvy všetkých atribútov, ktoré pre daný produkt poznáme, a tiež hodnoty priradené týmto atribútom. Ak teda vykonávame dopyt, tak sa môžeme pozerať práve na obsah (hodnotu) priradený k tomuto kľúču. Následne si

vieme v tomto obsahu vyznačiť zhody, ktoré sa našli so zadaným dopytom, a potom nájsť atribúty, v ktorých sa tieto zhody vyskytujú. Na základe toho vieme dosiahnuť to, že vo výsledku vyhľadávania sa na stránke zobrazia atribúty, v ktorých sa našli zhody s našim dopytom. Určiť domény produktov, ktoré nám vyhľadávanie vrátilo je jednoduchšie, keďže každý produkt má priradené pole domén, ku ktorým patrí.

Keď hovoríme o podobe dát Kapsy v Elasticsearch, je nutné spomenúť, že podoba týchto dát nie je striktne daná, ale v prípade potreby je možné dáta meniť. Súčasnú podobu dát je možné vidieť na Obr. 2 nižšie.


```
"_source": {
      "id_object": 68660,
      "id_domain": 4,
      "id_source": 17,
      "domain": [
            "Tablety",
            null
      ],
      "attribute_4": "109 eur",
      "attribute_36": "295 g",
      "attribute_37": "3910 mAh",
      "attribute_38": "7 \"",
      "attribute_39": "16 GB",
      "attribute_24": "42318",
      "attribute_40": "1 GB",
      "attribute_41": "áno",
      "attribute_42": "áno",
      "attribute_44": "áno",
      "attribute_32": "189.3 mm",
      "attribute_33": "7 mm",
      "attribute_34": "6 mm",
      "attribute_35": "7950.6 mm3",
      "attribute_26": "Asus MeMO Pad 7 ME176CX-1B046A White",
      "content": "cena 109 eur Hmotnosť Weight 295 g Battery capacity
            Kapacita batérie 3910 mAh Uhlopriečka displeja Display size 7 \"
            Storage Kapacita úložného priestoru 16 GB Local identifier
            Lokálny identifikátor 42318 Memory RAM Pamäť RAM 1 GB GPS áno
            Bluetooth áno WiFi áno Výška 189.3 mm Šírka 7 mm Hĺbka 6 mm Objem
            7950.6 mm3 Názov Name Asus MeMO Pad 7 ME176CX-1B046A White"
}
```


Obr. 2. Dáta Kapsy v Elasticsearch (JSON formát). Na obrázku su dáta popisujúce jeden produkt.

	id	id_attribute	id_language	name	description
▶	1	1	1	karoseria	typ karoserie
	2	2	1	pocet dveri	pocet dveri karoserie
	3	3	1	rok vyroby	rok vyroby automobilu
	4	4	1	cena	predajna cena
	5	5	1	stav	stav - havarovanost
	6	6	1	pocet kilometrov	pocet najazdenych kilometrov
	7	7	1	palivo	typ paliva
	8	8	1	pocet rychlosti	pocet rychlosti prevodovky
	9	9	1	objem motora	zdvihovy objem motora
	10	10	1	vykon	vykon motora
	11	11	1	pocet airbagov	pocet nainstalovanych airba...
	12	12	1	model	znacka a model automobilu

Obr. 3. MySQL tabuľka, na základe ktorej (okrem iného) vieme k identifikátoru atribútu priradiť názov atribútu

2.3 Návrh riešenia

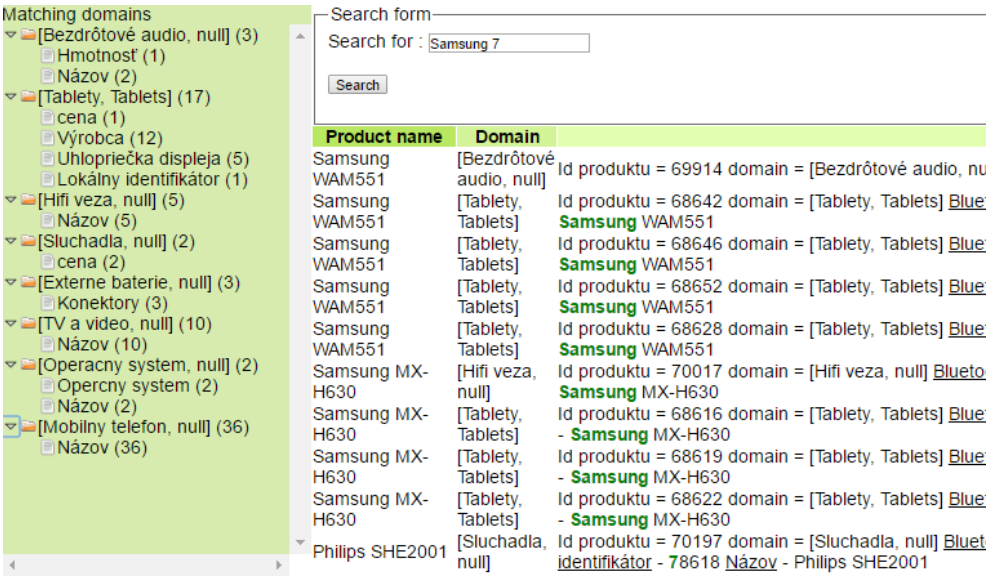
Po tom, ako užívateľ zadá do vyhľadávania vstupnú otázku, my vykonáme dopyt zodpovedajúci vstupnej otázky v Elasticsearch nad poľom *"content",* ktorého obsah je typu string (ako sme uvádzali v predchádzajúcej časti). Na to aby naše vyhľadávanie vedelo reagovať aj na možné preklepy a podobne, nastavíme dopytu parameter fuzziness. Taktiež pri vytváraní dopytu uvedieme, že chceme nájdené zhody, respektíve reťazce, kvôli ktorým nám Elasticsearch vrátil daný objekt, odchytiť do poľa *"highlight"*. V tomto poli nám Elasticsearch pri každom objekte vyznačí reťazce, ktoré vyhodnotil v danom poli (v našom prípade *"content"*) ako vyhovujúce dopytu. Tieto reťazce sú v poli vyznačené html tagmi (predvolený tag je **).

Následne tento výsledok dopytu, ktorého súčasťou je nielen objekt Elasticsearch, ako sme ho popisovali v minulej kapitole, ale aj pole *"highlight"*, spracujeme v Jave. Z poľa *"highlight"* si vyextrahujeme len slová vyznačené html tagmi. Teraz vieme, ktoré výrazy Elasticsearch vyhodnotil ako vyhovujúce (kvôli nastavenému parametru fuzziness, Elasticsearch vyhodnotí napríklad slovo *size* ako vyhovujúce pre slovo *side* v dopyte). My však ešte chceme vedieť, v ktorých atribútoch sa tieto výrazy nachádzajú. Preto prejdeme opäť našim výsledkom a zistíme, ktoré atribúty obsahujú tieto výrazy, pričom už aj vypočítame pre každý atribút počet výskytov týchto výrazov. Na to aby sme ešte každému atribútu dali názov a nezobrazovali ho v tvare, aký má v Elasticsearch, pripojíme sa na MySQL databázu, odkiaľ meno tohto atribútu získame. Tak už máme informáciu o tom, ktoré objekty (produkty) vyhovujú vstupnej otázke, v ktorých atribútoch sa našla zhoda a prakticky zadarmo máme aj informáciu

o tom, v akých doménach sú výsledné produkty, keďže táto informácia je uchovaná v Elasticsearch pri každom objekte v poli *"domain"*.

Na základe už získaných informácií chceme používateľovi zobraziť výsledky vyhľadávania v používateľskom rozhraní. Po vykonaní dopytu sú preto používateľovi zobrazené v používateľskom rozhraní produkty, ktoré vyhoveli jeho vstupnej otázke. Produkty sú utriedené na základe relevancie vypočítanej enginom Elasticsearch, pričom túto relevanciu ovplyvňujú faktory ako je počet výskytov hľadaného termínu v danom poli (v našom prípade v poli *"content"*), ako často sa hľadaný termín vyskytuje v celom indexe (veľa výskytov daného termínu spôsobí, že tento termín nebude mať až takú veľkú váhu pre zvýšenie relevancie daného dokumentu), a na záver je to aj dĺžka poľa, nad ktorým vyhľadávame (tu platí, že krátky termín v dlhom poli nemá až takú veľkú váhu pre relevanciu). Produkty a informácie o nich sú zobrazované tak, že sú (zelenou farbou) zvýraznené slová, na základe ktorých bol konkrétny produkt používateľovi vrátený ako vyhovujúci (viď Obr.4.).

Okrem zoznamu vyhovujúcich produktov je používateľovi zobrazený aj strom popisujúci výsledky vyhľadávania z pohľadu početností v jednotlivých doménach a atribútoch (viď Obr.4). Po kliknutí na niektorú položku v strome sa zobrazené produkty prefiltrujú podľa toho, kde v strome používateľ klikol. Ak klikol na položku s názvom domény, java opäť nadviaže spojenie s Elasticsearch a spustí v ňom dopyt s rovnakou vstupnou otázkou, akú používateľ naposledy zadal, avšak teraz už aj s požiadavkou, aby boli vrátené iba produkty, ktoré sú v takej doméne, na akú bolo kliknuté. Výsledky sú opäť zobrazené používateľovi a utriedené podľa relevancie vypočítanej Elasticsearchom. V prípade, že používateľ klikol v strome na položku s názvom atribútu prislúchajúcu k určitej doméne, vykoná sa podobná úloha ako pri kliknutí na doménu. Java spustí v Elasticsearch dopyt s rovnakou vstupnou otázkou, akú používateľ naposledy zadal a požiadavkou, aby výsledky boli produkty z domény, ku ktorej prislúcha atribút, na ktorý sa kliklo, a tiež aby výsledky obsahovali zhodu s dopytom v danom atribúte. Znova sú používateľovi zobrazené vyhovujúce produkty.



Obr.4. Náhľad na nami vytvorené používateľské rozhranie. Vľavo sa nachádza strom domén a atribútov, vpravo hore je formulár pre vyhľadávanie a pod ním sú zobrazené výsledky.

2.4 Analýza riešenia

Po návrhu implementácii metódy fulltextového vyhľadávania je potrebné naše riešenie ešte analyzovať. Analýzou chceme určiť úplnosť a presnosť nášho riešenia (o tom, ako tieto dve hodnoty určíme, sme písali v časti *1.1 Fulltextové vyhľadávanie*). Pre testovanie sme si vytvorili sadu 15 dopytov (viď Obr.5 a Obr.6). Tieto dopyty sme vytvárali tak, aby sme dostali čo najpestrejšie výsledky vzhľadom na výskyt zhôd v jednotlivých doménach a atribútoch.

Ku každému dopytu budeme musieť určiť pre vyhodnotenie úplnosti počet relevantných produktov v databáze a počet relevantných produktov vrátených našim vyhľadávaním. Pre vyhodnotenie presnosti budeme musieť určiť počet všetkých produktov vrátených našim vyhľadávaním a počet relevantných vrátených produktov. Čísla zapíšeme do tabuľky (viď Obr.5 a Obr.6).

	Dopyt	počet relevantných v DB	počet relevantných vrátených	Úplnosť
1	Samsung	67	67	100%
2	Samsung 7"			
3	WiFi áno Bluetooth áno			
4	Kapacita baterie 2000mAh			
5	Samsung Lenovo			
6	Lenovo nie			
7	Sony bluetooth áno			
8	Farba biela			
9	Slúchadlá jack			
10	Lenobo widi amo			

Obr.5. Dopyty pre testovanie (tabuľka úplnosti)

	Dopyt	počet všetkých vrátených	počet relevantných vrátených	Presnosť
1	Samsung	67	67	100%
2	Samsung 7"			
3	WiFi áno Bluetooth áno			
4	Kapacita baterie 2000mAh			
5	Samsung Lenovo			
6	Lenovo size 7			
7	Sony bluetooth áno			
8	Farba biela			
9	Slúchadlá jack			
10	Lenobo niee widi			

Obr.6. Dopyty pre testovanie (tabuľka presnosti)

Doposiaľ bolo otestovaných viacero jednoslovných dopytov a ukázalo sa, že pre jednoslovné dopyty dosahuje vyhľadávanie 100% úplnosť a 100% presnosť. Aktuálne však testujeme ďalšie (viacslovné) dopyty, ku ktorým presné výsledky nemáme avšak vyhľadávanie už neposkytuje takú dobrú presnosť, avšak úplnosť ostáva stále na vysokej úrovni.

3 Záver

Doposiaľ sme sa oboznámili s prácou v Elasticsearch a jeho Java API. Tiež sme sa naučili pracovať s Wicket frameworkom. Vytvorili sme jednoduché vyhľadávanie, ktoré vie nájsť produkty, ktoré nejako vyhovujú dopytu a tiež naše riešenie používateľa informuje o výsledkoch vyhľadávania v jednotlivých doménach a atribútoch. Máme vyriešené filtrovanie výsledkov podľa domén alebo domén a atribútov. Pre vyhľadávanie sme vytvorili jednoduché grafické rozhranie. Taktiež sme vytvorili testovaciu sadu, na základe ktorej budeme analyzovať navrhnutú metódu fulltextového vyhľadávania a začali sme už so samotným testovaním. Ostáva nám už len dokončiť fázu testovania a z výsledkov testovania vyvodiť záver o úspešnosti metódy z hľadiska úplnosti a presnosti.

Poďakovanie. Ďakujem RNDr. Petrovi Gurskému, PhD., vedúcemu mojej bakalárskej práce za rady a pomoc pri jej tvorbe.

Literatúra

1. Carstens, Carola ; Womser-Hacker, Christ: Ontology Based Query Expansion - Retrieval Support for the Domain of Educational Research. 2012 *[online] Dostupné na <http://d-nb.info/1023809400>*.
2. Baeza-Yates, Ricardo A. and Ribeiro-Neto, Berthier: Modern Information Retrieval, 1999 ISBN 020139829
3. Tvarožek, M., Bieliková, M. Bridging Semantic and Legacy Web Exploration: Orientation, Revisitation and Result Exploration Support. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. *[online] Dostupné na <http://acadmedia.wku.edu/Zhuhadar/WI-IAT%202010/Data/4191c326.pdf>*
4. Jozef Kuper: Unifikácia metadát o produktoch internetových obchodov. Diplomová práca, PF UPJŠ, 2015
5. Kowalski, G., Information Retrieval Architecture and Algorithms, 2011 ISBN 978-1-4419-7715-1