Crawlovanie a extrakcia relevantných častí webových portálov

Rudolf Pavel

3Ib, 2016 - 2017

Abstrakt. Práca je súčasťou školského projektu kapsa. Zaoberá sa crawlovaním webových portálov internetových obchodov a následnou extrakciou dát z týchto portálov. Kľúčovou časťou práce je zabezpečiť extrakciu relevantných častí webových portálov za pomoci automatického orezávania prehľadávania webového portálu prostredníctvom analýzy úspešnosti predchádzajúcich vetiev.

Kľúčové slová: crawlovanie, extrakcia dát, relevantný, analýza prehľadávania

1. Úvod

Projekt kapsa je vedecký projekt, ktorého cieľom je získavanie, extrakcia a následné spracovávanie dát z webových portálov internetových obchodov. Projekt kapsa sa skladá z viacerých častí. Táto práca sa venuje hlavne časti získavania dát.

Crawlovanie je proces, pri ktorom crawler vyhľadáva všetky linky na webovej stránke. Tieto linky môžu smerovať na ľubovoľné iné stránky a môže ich byť nespočetne veľa. Zároveň môžu viesť na stránky, ktoré sme už predtým crawlovali. To sú javy ktorým sa snažíme predchádzať a vyhýbať sa im. Cieľom je nasledovať iba odkazy, ktoré vedú na časti stránok crawlovaného internetového obchodu, ktoré sa nazývajú detailové stránky produkov. Detailová stránka produktu je webová stránka na nejakom internetovom obchode, ktorá obsahuje špecifické detaily a údaje o danom produkte internetového obchodu.

Samotnému procesu crawlovania predchádza anotácia. Anotácia prebieha v prehliadači na datailovej stránke produktu internetového obchodu za pomoci anotačného nástroja. My využívame anotačný nástroj Exago. Na detailovej stránke produktu daného e-shopu sa výberú atribúty, pomocou ktorých sa pri crawlovaní určuje, ktorá stránka je a ktorá nie je detailová. Takisto rozhodovanie o tom, ktorá stránka je relevantná pre crawler a ktorá nie, prebieha za pomoci regulárneho výrazu, ktorý sa vytvorí počas anotácie na danom internetovom obchode. Na základe týchto dát sa vytvoria pravidlá, ktoré sa využijú pri samotnom crawlovaní. Následne sa z týchto pravidiel vytvorí wrapper, do ktorého sa tieto pravidla uložia. Tento wrapper je súbor typu JSON, ktorý sa uloží do databázy. A teda podľa pravidiel z wrappera sa pri samotnom crawlovaní extrahujú vybrané atribúty, ktoré sú ukladané do databázy.

2. Návrh riešenia

2.1. Proces crawlovania

V práci sa využíva pre potreby crawlovania web crawler **Crawler4j.** Je to open source web crawler, určený pre javu a umožňuje multi threaded riešenie crawlovania. Pred samotným crawlovaním sa z databázy načíta wrapper, ktorý obsahuje údaje a atribúty z anotácie detailovej stránky produktu. Pre potreby simulovania preklikov na webovej stánke, využívame nástroj **Selenium,** ktorý umožňuje integráciu s webovým prehľadávačom Firefox a parsovanie HTML kódu. Ďalej využívame knižnice Jsoup a Xsoup, ktoré umožňujú prácu s HTML dokumentom v Jave. Z ich pomocou vieme z HTML kódu stránky získať jednotlivé elementy a následne pre tieto elementy vygenerovať XPAthy.

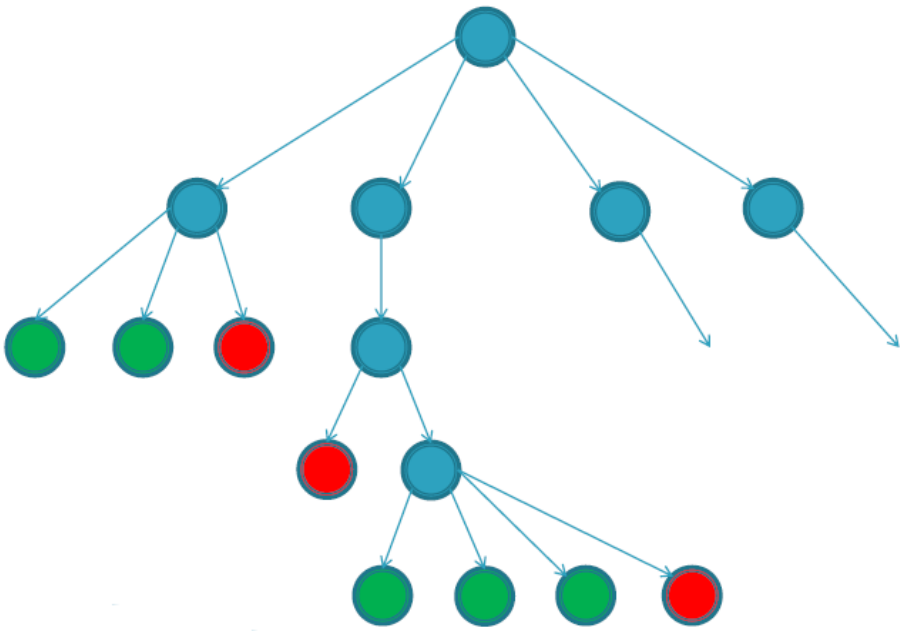
Bežný prístup pri procese crawlovania je ten, že crawler nasleduje všetky linky, ktoré nevedú mimo stránok internetového obchodu. To však stále predstavuje veľkú množinu stránok, ktoré nie sú relevantné pre extrakciu dát, pretože nevedú na detailové stránky produktov. Takisto je možné dostať sa na tú istú detailovú stránku produktu rôznymi cestami, s rôznymi dĺžkami ciest prehľadávania. Strom prehľadávania v takomto prípade bude obsahovať viacero slepých vetiev. Je to z toho dôvodu, že každý internetový obchod obsahuje okrem detailových stránok produktov aj ostatné stránky, ako napríklad stránka prihlásenia alebo registrácie a podobne. To je samozrejme situácia, ktorá je nežiaduca a chceme jej predchádzať. Naším hlavným problémom je nájsť všetky detailové stránky produktov čo najefektívnejšie.

Hlavným cieľom je teda nájsť všetky detailové stránky produktov a zároveň minimalizovať počet tých ostatných. A teda vyhnúť sa neúspešným vetvám v strome prehľadávania, ale sústrediť sa na tie relevantné a zároveň nevynechávať a vyhľadávať nové úspešné vetvy v kategóriách a produktoch internetového obchodu. K tomu je potrebné vytvoriť algoritmus pre nájdenie a označenie týchto slepých vetiev.

2.2. Crawlovanie eshopu

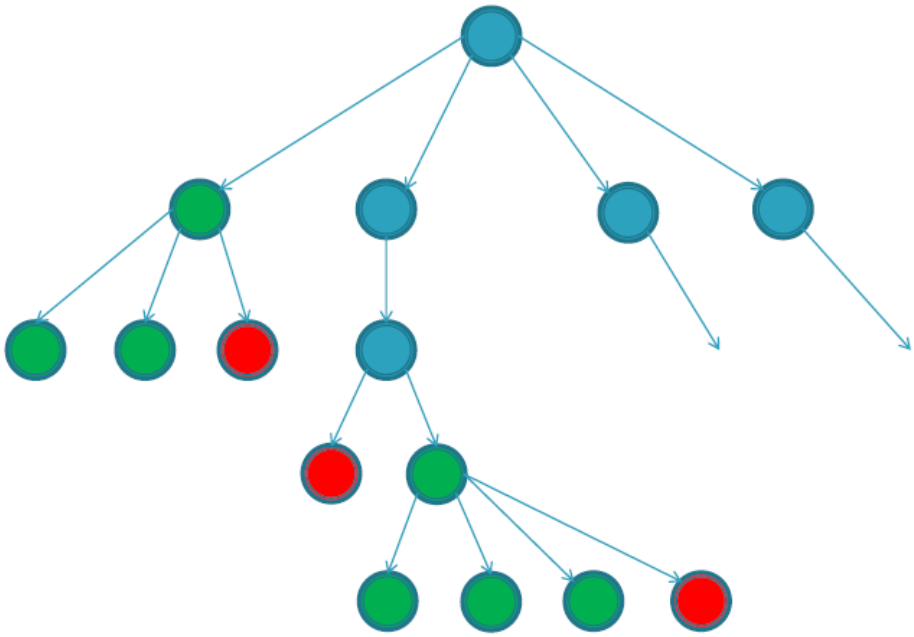
Predstavme si crawlovanie ľubovoľného eshopu. Crawlovanie sa začne na stránke eshopu ktorú zadáme crawleru ako vstup. Tento vstup bude url adresa, ktorú označujeme ako seedurl. Táto seedurl stránka predstavuje v strome prehľadávania koreň. Všetky linky, ktoré crawler na tejto stránke nájde, budú predstavovať deti koreňa stromu. Crawler potom prejde všetky deti koreňa a nájdené linky budú predstavovať ich deti. Tento proces sa zopakuje až pokým crawler neprejde všetky stránky eshopu.

Na obrázku 1. je zobrazený model stromu prehľadávania ktorý vznikne crawlovaním eshopu.



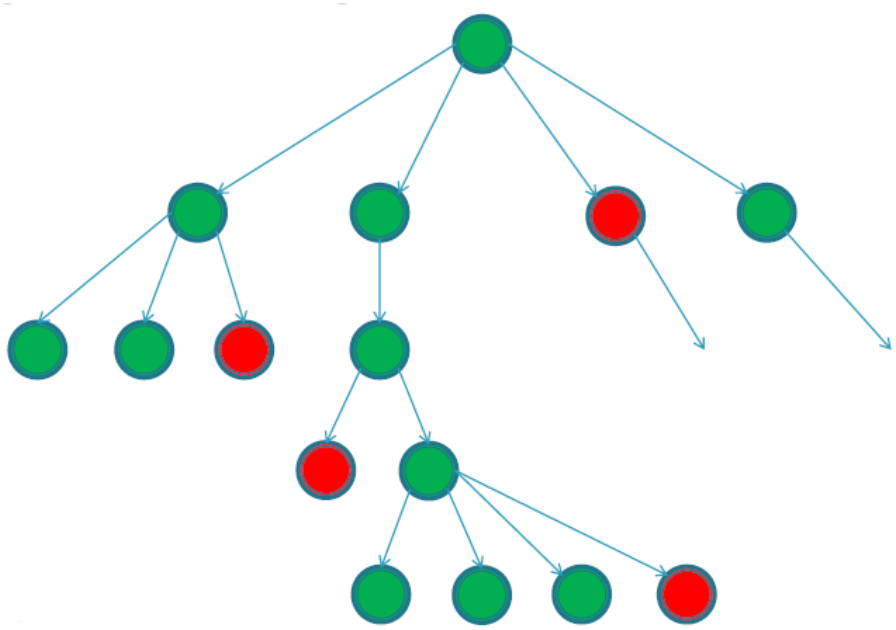
Obrázok 1. Model časti stromu prehľadávania – detailové stránky.

V tomto strome sú všetky listy vyfarbené na zeleno alebo na červeno. Zelenou farbou sú označené všetky listy, ktoré predstavujú detailové stránky produktov. Červenou farbou sú naopak označené všetky stránky, ktoré nie sú detailové.



Obrázok 2. Model časti stromu prehľadávania – zoznamové stránky.

Použitím metódy prechádzania stromu bottom – up označíme všetkých rodičov listových uzlov. V prípade ak sa z rodičovského uzla vieme priamo dostať k detailovým stránkam produktov alebo k zoznamom, ktoré k datailovým stránkam vedú, tak tento uzol označíme zelenou farbou. V opačnom prípade ho označíme červenou farbou. Tento proces opakujeme až pokým neoznačíme všetky uzly po koreň.

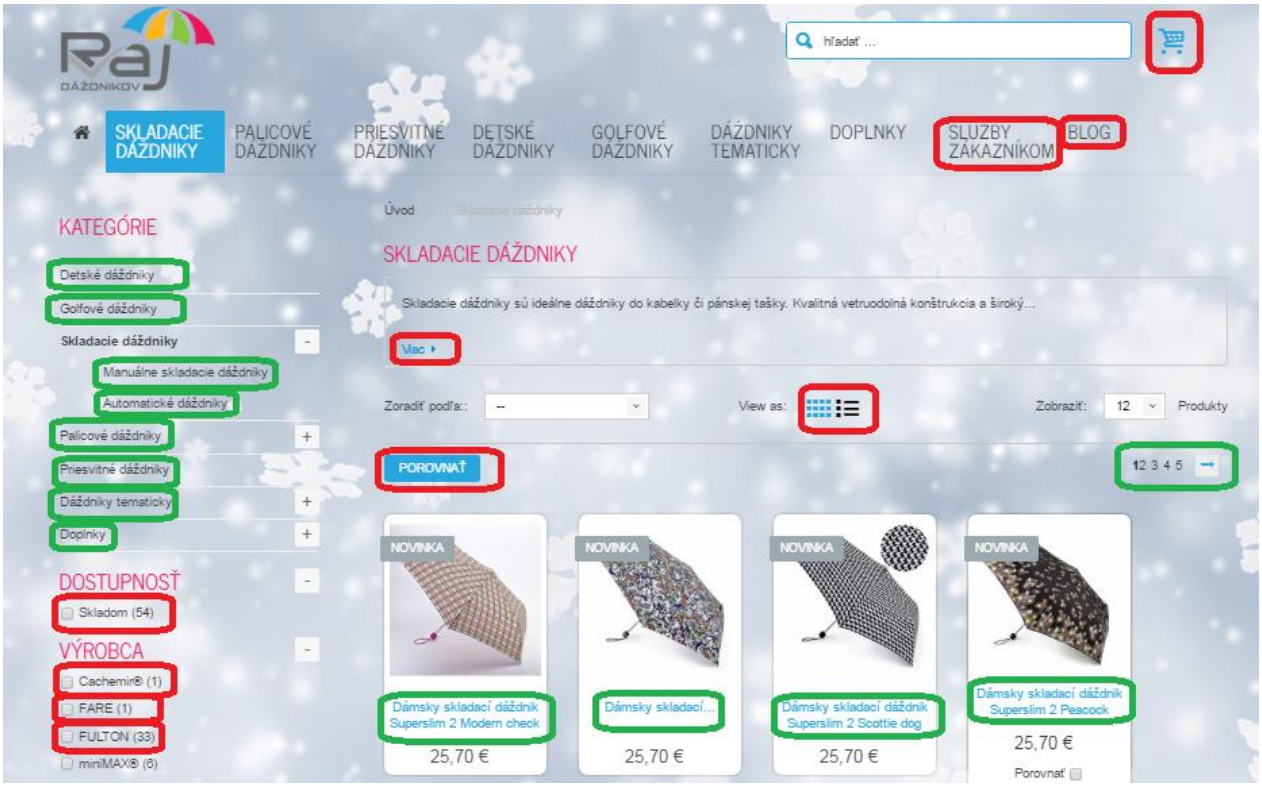


Obrázok 3. Model časti stromu prehľadávania – úspešné vetvy.

Označením celého stromu dostaneme pre každý rodičovský uzol zoznam úspešných potomkov teda úspešných vetiev. Potom následne použitím metódy top – down pri crawlovaní vieme crawleru ponúknuť množinu úspešných vetiev ktoré sa majú crawlovať.

2.3. Generalizácia XPathov

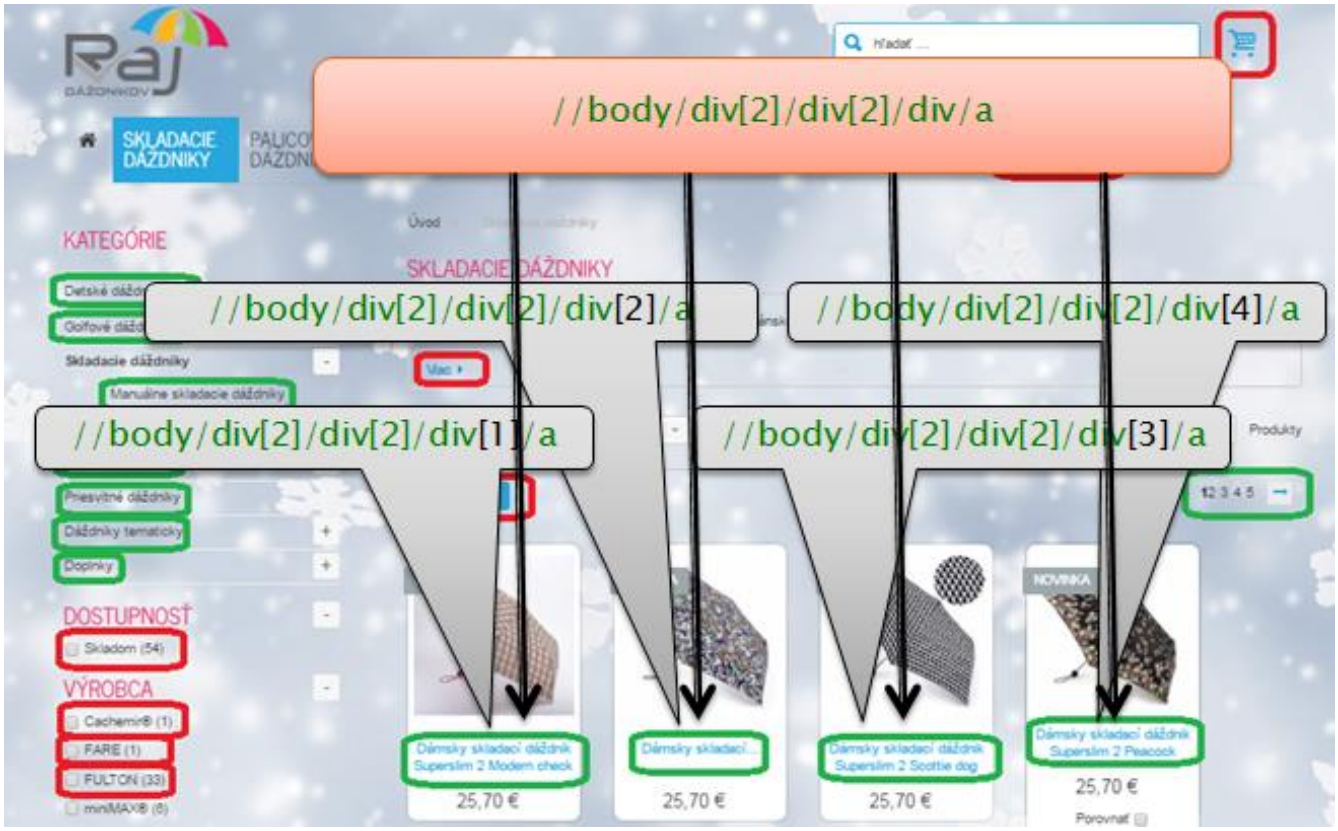
Ak sa zameriame strom prehľadávania z obrázku 3. a vyberieme nejaký zoznamový uzol. Tento zoznamový uzol bude vyzerať ako zoznamová stránka na nejakom eshope. Na príklade na obrázku 4. si ukážeme zoznamovú stránku z eshopu www.rajdazdnikov.sk.



Obrázok 4. www.rajdazdnikov.sk – zoznamová stránka.

Na obrázku 4. môžeme vidieť linky na ďalšie stránky eshopu, ktoré sú zakrúžkované na zeleno alebo na červeno. Zelenou farbou sú zakrúžkované linky, ktoré vedú na ďalšie zoznamové stránky alebo priamo na detailové stránky produktov. Naopak červenou farbou sú znázornené linky, ktoré predstavujú slepé vetvy v strome prehľadávania.

Pozrieme sa bližšie na štyri produkty v spodnej časti obrázku 4. Zistíme z HTML kódu elementy daných produktov a ich XPathy. Tieto XPathy môžeme vidieť na obrázku 5.



Obrázok 5. www.rajdazdnikov.sk – XPathy.

Na základe analýzy štyroch XPathov na obrázku 5., ktoré sú súčasťou úspešných vetiev, vieme zistiť určité podobnosti. Pozorujeme že tieto xPathy úspešných liniek z jednej stránky, ktoré na stránke patria do nejakého zoznamu produktov, sú si navzájom veľmi podobné. Tieto xPathy majú na zeleno zvýraznené časti rovnaké, aj keď sú to štyri rôzne produkty. Tieto XPathy sa líšia iba v jednom bode a to v tomto prípade v hranatej zátvorke pri poslednom div. Teda môžeme pozorovať, že xPath umožňuje adresovať nie iba konkrétny element, ale aj viacero elementov. Túto vlastnosť sme využili pri algoritme generalizácie XPathov.

Pri generalizácii XPathov sa nájde daný bod v ktorom sa XPathy líšia a na základe toho sa vytvorí generalizovaný XPath. Generalizovaný XPath pre naše štyri produkty je na obrázku 5. v červenej bubline. Ak takýto XPath zavoláme nad HTML dokumentom danej webovej stránky, dostaneme nie len tieto štyri produkty na obrázku 5., ale všetky produkty zo zoznamu produktov, ktorých XPathy spĺňajú danú podmienku.

2.4. Analýza prehľadávania

Aby sme mohli implementovať algoritmus nájdenia a označenia slepých vetiev, potrebujeme najprv poznať samotné dáta pri crawlovaní. Tieto dáta získame počas samotného procesu crawlovania. Zistíme kompletnú množinu liniek na danom internetovom obchode. Na základe tejto množiny crawlujeme jednotlivé webové stránky a získavame údaje ktoré sa ukladajú do databázovej tabuľky. Táto databázová tabuľka (**Tabuľka 1**.) obsahuje url adresu každej linky na danom internetovom obchode, ktorú crawlujeme. Ďalej obsahuje jednoznačný identifikátor rodiča danej url adresy v tejto tabuľke. Za pomoci tohto id vieme určiť rodiča pre danú url adresu. To využijeme pri rekonštrukcii stromu prehľadávania. Dôležitým údajom v tabuľke je xPathId. Toto id odkazuje na množinu XPathov v tabuľke 2. Samotný xPath získavame za pomoci url adresy. V HTML kóde pomocou url adresy nájdeme element, ktorý obsahuje hľadanú linku. Následne z tohto elementu vygenerujeme xPath k nemu v HTML kóde. Stĺpec isDetailPage nám označuje v akej hĺbke sa od koreňa nachádzame a že je to relevantná vetva. Hodnota 0 znamená že ide o slepú vetvu. V stĺpci generalized XPathId sa nachádza hash množiny generalizovaných XPathov, ktorý nám v tabuľke 3. určí, ktorý generalizovaný XPath patrí ktorej url adrese.

Táto **tabuľka 1**. v reáli predstavuje kompletný strom prehľadávania pri crawlovani internetového obchodu. Teda obsahuje všetky url adresy, ktoré crawler spracovával. Na základe týchto údajov z tabuľky analyzujeme kompletný strom prehľadávania a zisťujeme, ktoré vetvy v strome sú úspešné. Čo znamená že vedú k detailovým stránkam produktov.

Tabuľka 1. Tabuľka s údajmi samotného crawlovania.

id	ParentId	url	XPath_Id	isDetailPage	generalizedXPathId
1	0	http://penazenkyshop.sk/	0	3	1709448165
2	1	http://penazenkyshop.sk/cart/	1	0	0
3	1	http://penazenkyshop.sk/eshop/login	2	0	0
4	1	http://penazenkyshop.sk/kozene-opasky/c1022	3	2	7365840
5	4	http://penazenkyshop.sk/kozeny-opasok-sh-hnedy-10h/p665947c1023	4	1	2541916
6	4	http://penazenkyshop.sk/kozeny-opasok-sh-2016-chrom/p666233c1024	5	1	-1485511120
7	4	https://penazenkyshop.sk/registration/	6	0	0

id	idURL	XPath
1	1	//body/div[2]/div[1]
2	2	//body/div[2]/div[3]/div[2]
3	3	//body/div[2]/div[4]/div[3]
4	3	//body/div[2]/div[4]/div[4]

Tabuľka 2. Tabuľka s údajmi o XPathoch.

id	My Hash	idDownload	Generalized_Xpath
1	7365840	358	//body/div[2]/div[2]/div/a
2	7365840	358	//body/div[2]/div[2]/div[3]/div/a
3	2541916	358	//body/div[3]/div/section/div/div[3]/a

Tabuľka 2. Tabuľka s údajmi o generalizovaných XPathoch.

3. Záver

Na základe týchto vlastností v XPathoch, sme vytvorili algoritmus na hľadanie elementov, ktoré sú si podobné v XPathe. A teda tieto elementy budú obsahovať linky, ktoré vedú na podobné stránky. Za pomoci tohto algoritmu môžeme rozoznať slepé vetvy a následne ich označiť.

Pri ďalšej iterácii crawlovania sa teda množina crawlovaných liniek bude skladať len s relevantných, ktoré vedú čo najefektívnejšie k produktovým stránkam produktov na internetovom obchode. A teda samotné vetvy, ktoré boli označené ako slepé, už nebudú v ďalších iteráciách používané pri crawlovaní.

Zatiaľ sa v práci podarilo implementovať analýzu klikov pri crawlovaní. Vytvorenie a naplnenie databázovej tabuľky, ktorá obsahuje údaje o jednotlivých preklikoch počas crawlovania. Podarilo sa implementovať aj získavanie elementov z HTML kódu a následne ich xPathov pre jednotlivé prekliky pri crawlovaní . Ďalej je implementovaná generalizácia XPathov a označovanie slepých vetiev.

Ďalej je v pláne implementácia tohto riešenia do crawlera, aby pri opakovanom crawlovaní toho istého eshopu crawloval iba relevantné stránky. A teda efektívne nasledoval linky, ktoré vedú k detailovým stránkam produktov.

Následne nás čaká implementovanie identifikácie produktov na viacerých stránkach. Ďalej zavedenie politness a paralelného prehľadávania. A v neposlednom rade otestovanie na viacerých eshopoch a vylepšenie funkcionality.

Poďakovanie. Ďakujem RNDr. Petrovi Gurskému, PhD., za cenné rady pri tvorbe práce.

4. Literatúra

1. Selenium Documentation. Dostupné na internete: http://www.seleniumhq.org/docs/

2. Súčasný stav v metodológiách pre poloautomatické získavanie dát zo služieb či stránok webu, ich anotácia a konverzia do štruktúrovanej podoby a mapovanie na objekty z aplikačnej domény. Finálna správa projektu CeZIS. Košice. 2015

3. Návrh a popis metód pre poloautomatické získavanie dát zo služieb či stránok webu, ich anotácia a konverzia do štruktúrovanej podoby a mapovanie na objekty z aplikačnej domény. Finálna správa projektu CeZIS. Košice. 2015