Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Aljaž Srša 63120233 in Gregor Sušnik 63120102

Zajemanje vsebine spletnih strani

Poročilo druge seminarske naloge pri predmetu Iskanje in ekstrakcija podatkov s spleta

Asistent: asist. prof. dr. Slavko Žitnik

**Povzetek**

Zajemanje zanimivih podatkov spletnih strani <insert\_some\_more\_intro\_here>. V seminarski nalogi implementirava zajemanje podatkov na tri različne načine . Vsak način opiševa in pokomentirava težave, na katere sva naletela pri implementaciji. Implementirane metode preizkusiva na določenih parih spletnih strani – overstock, rtvslo.si in <INSERT\_CUSTOM\_SITE\_HERE>. Dobljene rezultate primerjava in pokomentirava v poglavju rezultati.

1. **Uvod**

Ročno zajemanje podatkov iz spletnih strani je precej zahtevno in ne praktično. Še posebej, če imamo opravka z veliko količino podatkov. Zajemanje večje količine podatkov poskušamo avtomatizirati, saj s tem prihranimo veliko časa in dela, ki bi ga opravljali ročno.

V drugi seminarski nalogi smo se lotili procesa zajemanja podatkov. Ideja je bila implementirati na tri različne načine za pridobivanje podatkov, in sicer: uporaba regularnih izrazov, uporaba XPath izrazov in uporaba RoadRunner pristopa. Vsak način smo na koncu preizkusili na treh različnih tipih strani. Vsak tip strani je vseboval dve strani. Nad vsemi tremi pari stranmi – skupaj torej šest strani, so se preizkusili vsi trije načini za pridobivanje podatkov in med posameznima stranema v paru se je primerjalo pridobljene podatke.

1. **Metode pridobivanja podatkov**

Seminarsko nalogo sva izdelala v jeziku Python. Program se začne izvajati v glavni datoteki main.py, v kateri se prebere HTML datoteka. Prebrana datoteka se najprej obdela s pomočjo Seleniuma, nato pa se sva s pomočjo knjižnice BeautifulSoup celotni HTML še olepšala v smislu optimizacije, zaključka značk, itp. Tako pripravljen HTML je vhod za vse tri načine pridobivanja podatkov. Vsak način je implementiran kot funkcija v ločeni datoteki. Kateri tip spletne strani pa ureja spremenljivka pageType. Tip je lahko nastavljen na 0 (Overstock), 1 (rtvslo.si) ali 2 (<INSERT\_CUSTOM\_SITE\_HERE>)

* 1. **Uporaba regularnih izrazov**

Funkcija sprejme obdelan HTML string in tip spletne strani. Na podlagi podatka o tipu spletne strani se izvede koda. Za zajem podatkov poskrbijo spodnji regularni izrazi:

**Overstock**

regex = r'<b>([0-9].+)<**\/**b>.\***\n**.\***\n**.\*<s>([**\$**0-9,.]+).\***\n**.\*<b>([**\$**0-9.]+).\***\n**.\*>([**\$**0-9.,]+)\s.([0-9]+.).\***\n**.\***\n**.\*<span class=**\"**normal**\"**>(.\*|.\***\n**.\***\n**.\*|.\***\n**.\***\n**.\***\n**.\*)<br/><a'

Razlaga regularnega izraza:

* <b>([0-9].+)<**\/**b> poišče naziv izdelka, ki pa se vedno prične z neko številko
* .\***\n**.\***\n**.\* Izpusti nekaj vsebine, ki je med nazivom izdelka in cenami
* ([**\$**0-9,.]+) Zajame vsebino 'listPrice'
* .\***\n**.\* Izpusti nekaj vsebine
* <b>([**\$**0-9.]+) Zajame vsebino 'price'
* .\***\n**.\* Izpusti nekaj vsebine
* >([**\$**0-9.,]+)\s.([0-9]+.).\***\n**.\***\n**.\* Zajame 'saving' in 'savingPercent'
* <span class=**\"**normal**\"**>(.\*|.\***\n**.\***\n**.\*|.\***\n**.\***\n**.\***\n**.\*)<br/><a' Zajame vsebino opisa izdela. Pri čemer ima opis lahko več vrstic opisa.

**Rtvslo.si**

**<CUSTOM\_SITE>**

Zajem podatkov se shrani v slovar objekt, ta pa se s pomočjo knjižnice json preoblikuje v json objekt. Rezultat funkcij je json dump.

* 1. **Uporaba XPath**

Funkcija sprejme obdelan HTML string in tip spletne strani. Na podlagi podatka o tipu spletne strani se izvede koda. Za zajem podatkov poskrbijo spodnji XPath izrazi:

**Overstock**

Najprej poiščemo del v HTML, kjer se nahajajo željeni podatki. Iščejo se vsi takšni <TR> nodi, ki imajo atribut bgcolor nastavljen na določeno vrednost pri tem pa mora omenjen node imeti še točno dva otroka tipa <TD>

*objects = tree.xpath('//tbody/tr[(contains(@bgcolor, "#ffffff") or contains(@bgcolor, "#dddddd")) and count(td[@valign="top"]) = 2]*

Nato se nad trenutno lokacijo pridobivajo podatki o vsebini, ki nas zanimakar v for zanki.

for obj in objects:

*title = obj.xpath('string(a/b/text())')*

*listPrice = obj.xpath('string(table/tbody/tr/td[1]/table/tbody/tr[1]/td[2]/s/text())')*

*price = obj.xpath('string(table/tbody/tr/td[1]/table/tbody/tr[2]/td[2]/span/b/text())')*

*saving = obj.xpath('substring-before(table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span/text(), " ")')*

*savingPercent = obj.xpath('substring(substring-after(table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span/text(), " "),2 ,3)')*

*content = obj.xpath('string(table/tbody/tr/td[2]/span/text())')*

Tudi tukaj se izvede json dump, kise vrne v main kot rezultat funkcije.

* 1. **Uporaba RoadRunner algoritma**

1. **Rezultati**

**Literatura**