UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

ISKANJE IN EKSTRAKCIJA PODATKOV S SPLETA

SEMINARSKA 2

Ekstrakcija strukturiranih podatkov

Končno projektno poročilo

Avtorja:
BERNIK Matic
TOVORNIK Robert

Profesor:
dr. Marko BAJEC
Asistent:
doc. dr. Slavko ŽITNIK

May 8, 2019

1 Uvod

1.1 Opis problema

Cilj seminarke naloge je bil implementacija treh pristopov za ekstrakcijo podatkov v strukturirani obliki s spletnih strani. Pristopi so obsegali:

- uporabo regularnih izrazov,
- uporabo XPath izrazov,
- implementacijo RoadRunner algoritma.

Poleg stirih spletnih strani dveh pred-dolocenih vrst (overstock.com in rtvslo.si) je naloga zahtevala obdelavo dveh dodatnih spletnih strani poljubnega tipa. V ta namen sva izbrala strani oglasov s portala `bolha.com`.

2 Izbrani strani bolha.com

Dodatno sva za namene ekstrakcije podatkov izbrala par 'detail' spletnih oglasov s portala bolha.com.

Strukturi spletnih strani in podatki, katere sva se namerila iz njiju izlusciti, sta razvidni iz spodnjih slik.

Razvidno je, da se strani med drugim razlikujeta tudi v stevilu podatkov o uporabniku (uporabnikova telefonska in mobilna stevilka) ter dolzini clenov v navigacijskem razdelku (kategorija).

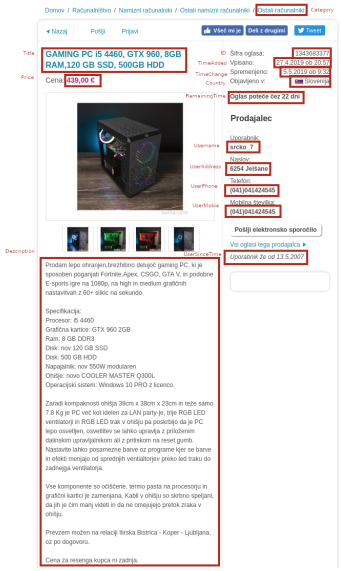

Figure 1: Slika prikazuje prvo izmed izbranih spletnih strani s portala bolha.com. Na njej so z rdeco barvo oznaceni elementi, katere nase implementacije izluscijo iz dokumenta.

Figure 2: Slika prikazuje drugo izmed izbranih spletnih strani s portala bolha.com. Na njej so z rdeco barvo oznaceni elementi, katere nase implementacije izluscijo iz dokumenta.

3 Implementacije

Programska koda iz katere so razvidni tudi vsi uporabljeni regularni in XPath izrazi je dostopna na naslovu https://github.com/roberttovornik/webpage-data-extraction/blob/master/code/data_extraction.py.

3.1 Regularni izrazi

Za ekstrakcijo podatkov z uporabo regularnih izrazov sva uporabljala Pythonovski modul 're'.

3.1.1 bolha.com

Ekstrakcija vecine podatkov je bila relativno enostavna, saj se pred polji, ki nas zanimajo, nahajajo labele, ki jih unikatno oznacujejo. Uporabljen regularni izraz za ekstrakcijo datuma spremembe oglasa je tako naprimer

```
'<p><label>Spremenjeno:<\/label>(.*)<\/p>'.
```

Za vsak tip podatka sva uporabila svoj regularni izraz.

3.1.2 rtvslo.si

Zaradi blizine naslova in podnaslova clankov znotraj HTML dokumentov, sva oba izluscila kar z uporabo enega samega regularnega izraza

```
'<h1>(.*)</h1>\textbackslash s+
<div class=\textbackslash"subtitle\textbackslash">(.*)<\textbackslash/div>'
```

, s cimer sva tudi pridobila na robustnosti pogoja.

Ekstrakcije vsebine clanka sva se lotila tako, da sva najprej poiskala vse odstavke oz. paragrafe <p> z uporabo regularnega izraza '<p[^>]*?>([\s\S]*?)<\/p>'. Od teh sva izlocila tiste odstavke, ki vsebujejo '<iframe', besedilo iz vseh preostalih pa zdruzila. V tako dobljenem besedilu sva nato zamenjala znacke '
' s prelomi vrstic '\n' in odstranila vse preostale znacke, ki so ustrezale regularnim izrazom '<\/?[a-zA-Z]*>'

3.1.3 overstock.com

Zaradi ponavljanja uporabe enakih HTML elementov kot sta <td> in <tr> skozi dokument je bila ekstrakcija tu nekoliko tezavnejsa in je zahtevala uporabo daljsih regularnih izrazov. Tako sva uporabila izraz

```
'<tr bgcolor=\"#[fd]*\">\s*<td valign="top" align="center">[\s\S]*?
<td valign="top">([\s\S]*?)<\/td>\textbackslash s*<\/tr>\s*<tr>\s*
<td colspan="2" height="4">'
```

da sva iz HTML dokumenta najprej izluscila seznam posamezih oglasov oz. njihovih relevantnih odsekov.

V nadaljevanju sva obdelovala vsakega od oglasov posebej. Podatke vezane na ceno izdelka ('ListPrice', 'Price', 'Saving' in 'SavingPercent') sva izluscila z uporabo enega daljsega regularnega izraza:

```
'<tbody><tr><td align="right" nowrap="nowrap"><b>List Price:</b></td>
<td align="left" nowrap="nowrap"><s>(.*)</s></td></tr>\s+<tr><td align="right"
nowrap="nowrap"><b>Price:</b></td><td align="left" nowrap="nowrap">
<span class="bigred"><b>(.*)</b></span></td></tr>\s+<tr>
<td align="right" nowrap="nowrap"><b>You Save:</b></td><td align="left"
nowrap="nowrap"><span class="littleorange">([\^{}\(]*)\s+\(([\^{}\)]*)\)</span>
</td></tr>\s+</tbody>'
```

3.2 XPath

Za ekstrakcijo podatkov z uporabo XPath izrazov sva uporabljala Pythonovski modul 'lxml'. Modul preko metode text_content() omogoca pridobitev vsega besedila iz celotnega HTML pod-drevesa, na racun cesar sva poenostavila nekatere XPath izraze.

V nekaterih primerih sva za osnovo XPath oznake vzela izraz, kakrsnega za HTML element zgenerira Google Chrome Developer Tool in ga nato preuredila. Spremembe so bile potrebne zaradi vecje robustnosti oz. ker so se samodejno generirani izrazi raje opirali na pozicije elementov, kot pa na njihove atribute.

3.2.1 bolha.com

Primer XPath izraza za pridobitev podatka o kategoriji oglasa s pripadajoco Python kodo:

```
tree = html.fromstring(document)
tree.xpath('//*[@id="breadcrumbs"]/nav/a[last()-1]')[0].text_content()
```

Osnovne podatke o oglasu kot so 'ID', 'TimeAdded', 'TimeChange' in 'Country' sva izluscla kot besedilo zaporednih paragrafov odseka 'documentInfo':

```
document_info_paragraphs =
tree.xpath('//div[@class = "infoBox"]/div[@class="documentInfo"]/p')
    article['ID'] = document_info_paragraphs[0].xpath('./text()')[0]
    article['TimeAdded'] = document_info_paragraphs[1].xpath('./text()')[0]
    article['TimeChange'] = document_info_paragraphs[2].xpath('./text()')[0]
    if len(document_info_paragraphs)>4:
        article['Country'] = document_info_paragraphs[3].xpath('./span/text()')[0]
```

3.2.2 rtvslo.si

Podatke sva izluscila z uporabo locenih XPath izrazov za vsakega izmed podatkov. Ker je vsebina clanka 'Content' razdeljena na razlicne paragrafe, sva text le-teh poiskala z uporabo regularnega izraza

```
'//*[@id="main-container"]/div[3]/div/div[2]/article/p/text()'
```

in jih s prelomi vrstic zdruzila.

3.2.3 overstock.com

Najprej sva z uporabo regularnega izraza:

```
'/html/body/table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td
/table/tbody/tr[@bgcolor]'
```

poiskala HTML pod-drevesa, ki pripadajo posameznim oglasom. V nadaljevanju sva za vsak oglas loceno izluscila zahtevane podatke. Primer stavka za izluscitev vsebine oglasa 'Content':

```
listings[i]['Content'] = listing_trees[i].xpath('./td[2]/table/tbody/tr/td[2]')[0]
.text_content()
```

3.3 RoadRunner

4 Rezultati

Izhodi vsake od implementiranih metod so dostopni v obliki .json datotek na povezavi https://github.com/roberttovornik/webpage-data-extraction/tree/master/data/extracted_data.

5 Povezave

Github repozitorij projekta se nahaja na naslovu: https://github.com/roberttovornik/webpage-data-extraction