

Luka Dragar, Matej Vatovec, Aleksander Kovač

# EXTRACTION

2. domača naloga za predmet Iskanje in ekstrakcija podatkov s spleta

MENTOR: prof. dr. Marko Bajec, doc. dr. Slavko Žitnik

## 1. Uvod

V poročilu smo povzeli tri načine, kako se lahko izluščijo podatki iz treh različnih spletnih strani. Prva je bila rtvslo.si ter druga overstock.com.

## 2. Opis izbranih spletnih strani

Za 2 izbirni strani smo si izbrali ponudnike študentske prehrane, kjer smo izbrali katere podatke želimo izluščiti s pomočjo XPATH-a, regexov in algoritma »Road Runner«. Sledeča slika prikazuje izbrane parametre.

The screenshot displays the 'Študentska prehrana' website interface. At the top, there is a navigation bar with the logo, 'NOVICE', 'TOČKE SŠP', and 'IMENIK LOKALOV'. Below the navigation bar, there are several buttons: 'Nazaj', a menu icon, a red 'i' icon, a printer icon, and a camera icon. The main content area shows details for 'ABI FALAFEL'. The title 'ABI FALAFEL' is highlighted with a red box. The address 'Trubarjeva cesta 40, 1000 Ljubljana (041 640 166)' is also highlighted. The price 'Doplačilo : 5,14 EUR' and 'Cena obroka : 9,00 EUR' are highlighted. The main dish '1 ŠPINAČNI ZAVITKI, PRILOGA' is highlighted. The salad 'mešana solata' is highlighted. The work time 'Med tednom : 10:00 - 21:00' is highlighted. The weekend time 'Sobota : 10:00 - 21:00' is highlighted. The holiday time 'Nedelja / Prazniki : 12:00 - 21:00' is highlighted. The website also features a logo for 'Fal' and a small icon of a person.

Študentska prehrana

NOVICE TOČKE SŠP IMENIK LOKALOV

Nazaj

ABI FALAFEL Title

Trubarjeva cesta 40, 1000 Ljubljana (041 640 166) Address

Doplačilo : 5,14 EUR Cena obroka : 9,00 EUR List price

Price

pet., 05.maj sob., 06.maj ned., 07.maj

Delovni čas

Med tednom : 10:00 - 21:00 Work time

Sobota : 10:00 - 21:00

Nedelja / Prazniki : 12:00 - 21:00

1 ŠPINAČNI ZAVITKI, PRILOGA Main dish

mešana solata Salad

jabolko/hruška/pomaranča

2 ZAVITKI S PAPRIKO, PRILOGA

mešana solata

jabolko/hruška/pomaranča

### 3. Implementacija

Ukaz »pip install -r requirements.txt namesti potrebne pakete potem pa zaženemo run-extraction.py skripto ter odvisno od zahtevanega algoritma dodamo argument (A – regex, B – xpath, C – »Road Runner«).

#### 3.1 Tabela regularnih izrazov in XPath za posamezne strani

<b><u>REGEX</u></b>
<b>RTV Slo"</b>
<pre>regex_list = [     (r"&lt;h1&gt;(.*?)&lt;/h1&gt;", "Title"),     (r"&lt;div class=\"subtitle\"&gt;(.*?)&lt;/div&gt;", "SubTitle"),     (r"&lt;p class=\"lead\"&gt;(.*?)&lt;/p&gt;", "Lead"),     (r"&lt;div class=\"author-name\"&gt;(.*?)&lt;/div&gt;", "Author"),     (r"&lt;div class=\"publish-meta\"&gt;\n\t\t(.*?)&lt;br&gt;", "Published- Time"),     (r"&lt;div class=\"article-body\"&gt;(.*?)&lt;div class=\"gallery\"&gt;", "Content") ]</pre>
<b>Overstock</b>
<pre>regex_list = [     (r"&lt;td&gt;&lt;a href=\"http://www\.overstock\.com/cgi- bin/d2\.cgi\?PAGE=PROFRAME[\w\W]*?\"&gt;&lt;b&gt;(.*?)&lt;/b&gt;", "Title"),     (r"&lt;s&gt;(.*?)&lt;/s&gt;", "List price"),     (r"&lt;span class=\"bigred\"&gt;&lt;b&gt;(.*?)&lt;/b&gt;", "Price"),     (r"&lt;b&gt;You Save:[\w\W]*?class=\"littleorange\"&gt;(.*?)&lt;/span&gt;", "Sa- ving"),     (r"&lt;span class=\"normal\"&gt;(.*?)&lt;/span&gt;", "Content") ]</pre>
<b>Študentska prehrana</b>
<pre>regex_list = [     (r"&lt;h3 class=\"no-margin bold\"&gt;(.*?)&lt;/h3&gt;", "Title"),     (r"&lt;small&gt;(.*?)&lt;/small&gt;[\w\W]*", "Address"),     (r"&lt;span class=\" color-light-grey\"&gt;(.*?)&lt;/span&gt;", "Price"),     (r"&lt;span class=\" color-light-grey\"&gt;(.*?)&lt;/span&gt;\s*&lt;/small&gt;", "List price"),     (r"&lt;div class=\"col-md-12 text-bold\"&gt;(.*?)&lt;/div&gt;", "Work time"),     (r"&lt;strong class=\" color-blue\"&gt;(.*?)&lt;/strong&gt;", "Main dish"),</pre>

```
(r"<ul class=\"list-
unstyled\">(?:\s*<li>.*?</li>){1}\s*<li>.*?<i>(.*?)</i>", "Salad"),
]
```

## **XPATH**

### **RTV Slo**

```
{
  "Title": '//*[@id="main-container"]/div[3]/div/header/h1',
  "SubTitle": '//*[@id="main-container"]/div[3]/div/header/div[2]',
  "Lead": '//*[@id="main-container"]/div[3]/div/header/p',
  "Author": '//*[@id="main-container"]/div[3]/div/div[1]/div[1]/div',
  "PublishedTime": '//*[@id="main-contai-
ner"]/div[3]/div/div[1]/div[2]/text()[1]',
  "Content": '//*[@id="main-container"]/div[3]/div/div[2]/article',
}
```

### **Overstock**

```
common_path = '/html/body/'
common_path = f'{com-
mon_path}table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/t
able/tbody/tr[{i}]/td[2]'
{
  "Title": f'{common_path}/a/b',
  "List price": f'{com-
mon_path}/table/tbody/tr/td[1]/table/tbody/tr[1]/td[2]/s',
  "Price": f'/{com-
mon_path}/table/tbody/tr/td[1]/table/tbody/tr[2]/td[2]/span/b',
  "Saving": f'{com-
mon_path}/table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span',
  "Content":
    '/html/body/table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/t
d/table/tbody/tr[3]/td[2]/table/tbody/tr/td[2]/span/text()'
}
```

### **Študentska prehrana**

```
common_path = '/html/body/'
common_path = f'{common_path}div[3]/div[2]'

{
  "Locale name": f'{com-
mon_path}/div[1]/div[1]/div[2]/div[1]/h3[1]',
}
```

```

        "Address": f'{com-
mon_path}/div[1]/div[1]/div[2]/div[1]/small[1]',
        "Price": f'{com-
mon_path}/div[1]/div[1]/div[2]/div[1]/small[2]/span[2]',
        "List price": f'{com-
mon_path}/div[1]/div[1]/div[2]/div[1]/small[2]/span[4]',
        "Work time": f'{com-
mon_path}/div[2]/div[1]/div[1]/div[1]/div[3]/div[1]/div[2]/div[1]',
        "Main dish": f'//strong[@class="color-blue"][{i}]',
        "Salad": f'//ul[@class=list-unstyled][{i}]/li[2]/i[1]'
    }

```

## 3.2 Road runner

Ideja naše implementacije zaznavanja poljubnih elementov in iteratorjev z algoritmom Road runner je sledeča.

Kot začetni wrapper vzamemo prvo stran, ki jo lahko poljubno predprocesiramo. Nekatere dele strani lahko že takoj spremenimo v regex izraze.

Recimo `<script>` ali `<style>` značke. Tako imamo wrapper, ki je sestavljen iz značk, nizov in regex izrazov.

Nato začnemo primerjati wrapper z naslednjo stranjo. Po vrsti vzemamo elemente iz wrapperja in gledamo če jih najdemo na vrhu druge strani. Če dobimo ujemanje, posodobimo wrapper in odstranimo element iz druge strani. Če ne preverimo ali smo naleteli na iterator ali na poljubni element.

Podani sta psevdokodi za detekcijo iteratorja in detekcijo poljubnih elementov.

## Psevdokoda za detekcijo iteratorja:

```
function detect_iterators(wrapper, sample, mismatch):  
    output: wrapper, if wrapper was changed else None  
    let terminal_tag = wrapper[mismatch-1] be the closing tag  
    B1= []  
    for line, index in wrapper[mismatch:]:  
        if tag(line) == terminal_tag:  
            B1= wrapper[mismatch:index]  
            break  
        else number of closing tabs > number of openning tabs:  
            return None  
    for line in B1:  
        if line does not match wrapper upwards:  
            changes = detect_iterators()  
            if iterator is None:  
                changes = detect_optionals()  
            update_wrapper(changes)  
    repeat for sample  
    return wrapper
```

Iskanje poljubnega elementa:

```
for wrapper_element in wrapper_elements:
    if is_match(wrapper_element, second_page):
        new_wrapper += wrapper_element
    else:
        is_found, second_page_element_location =
            search_for_element(wrapper_element, second_page_element)
        if is_found:
            optional_element =
                second_page[:second_page_element_location]
            second_page = second_page[second_page_element_location:]
            new_wrapper += optional_element
        else:
            second_page = old_second_page
            is_found, wrapper_element_location =
                search_for_element(second_page_element, wrapper_elements)
            if is_found:
                optional_element =
                    wrapper_elements[:wrapper_element_location]
                wrapper_elements =
                    wrapper_elements[wrapper_element_location:]
                new_wrapper += optional_element
            else:
                new_wrapper += optional_elements_regex
```

Tako ponovno dobimo wrapper sestavljen iz značk, nizov in regex izrazov, ki ga lahko apliciramo na poljubno število nadaljnjih strani.