Luka Dragar, Matej Vatovec, Aleksander Kovač

EXTRACTION

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

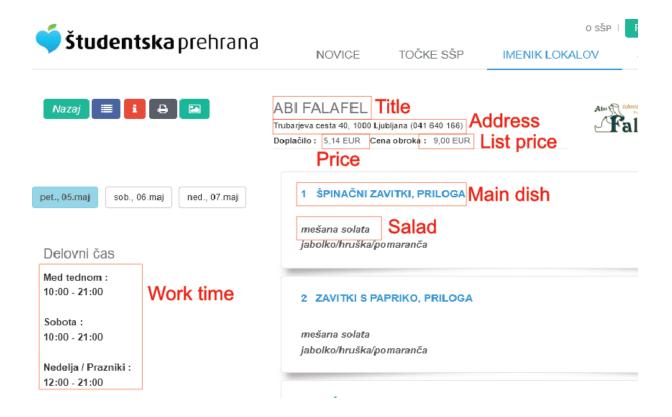
MENTOR: prof. dr. Marko Bajec, doc. dr. Slavko Žitnik

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.



3. Implementacija

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

3.1 Tabela regularnih izrazov in XPath za posamezne strani

REGEX

RTV Slo"

Overstock

Študentska prehrana

```
(r"(?:\s*.*?){1}\s*.*?<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

Študentska prehrana

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, missmatch):
   output: wrapper, if wrapper was changed else None
   let terminal tag = wrapper[missmatch-1] be the closing tag
   B1= []
   for line, index in wrapper[missmatch:]:
       if tag(line) == terminal_tag:
            B1= wrapper[missmatch: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 nadaljnih strani.