Web Information Extraction and Retrieval
Programming Assignment 2

Jure Bevc, Luka Tavčer, Grega Dvoršak
University of Ljubljana, Faculty for Computer and Information Science
bevc.jure@gmail.com, lt7339@student.uni-lj.si, gd4667@student.uni-lj.si

May 1, 2020

1 Introduction

This report summarizes our work on the second assignment for Web Information Extraction and Retrieval. Using the web pages provided in the assignment instructions and our own selected web pages, we implemented three data extraction methods that find relevant data items in the given pages.

2 Selected web pages

Besides using the data records from websites specified in the instructions, we also extracted data from an advertisement and shopping website *Bolha.com* [1]. The layout of this site is similar to the Overstock website, where there are several data records and each of them have data records we are interested in. Figure 1 shows these data records and their names.

Figure 1: Data items of a single record on bolha.com

3 Extraction using regular expressions

The Python **re** module allows us to extract groups of characters with regular expression syntax [2]. The following list of regular expressions were used to extract data records from the RTV SLO website:

- Author:
  `<div class=\"author-name\">([^<]*)</div>`
- PublishedTime:
  `\s*(\d[^<]*)<br>`
- Title:
  `<title>([^<]*)</title>`
- SubTitle:
  `<div class=\"subtitle\">([^<]*)</div>`
- Lead:

`<p class=\"lead\">(.*)</p>`

- Content:
  `<p[^>]*>([^<]*)</p>.*`

The following list of regular expressions were used to extract data records from the Overstock website:

- Title:
  `PROD_ID[^>]*><b>([^<]*)</b></a><br>`
- ListPrice:
  `<b>List Price:</b></td><td align=\"left\"nowrap=\"nowrap\"><s>([^<]*)</s></td></tr>`
- Price:
  `<b>Price:</b></td><td align=\"left\" nowrap=\"nowrap\"><span class=\"bigred\"><b>([^<]*)</b></span></td></tr>`
- Saving/SavingPercent:
  `<b>You Save:</b></td><td align=\"left\" nowrap=\"nowrap\"><span class=\"littleorange\">([^<]*)</span></td></tr>`
- Content:
  `</td><td valign=\"top\"><spanclass=\"normal\">([^<]*)<br>`

The following list of regular expressions were used to extract data records from the Bolha.com website:

- Title:
  `<h3 class=\"entity-title\"><a[^>]*>([^<]*)</a></h3>(?=.*Zadnji oglasi)`
- Description:
  `<div class=\"entity-description-main\">\n*\s*([^<]*)<br>`
- Price:
  `<strong class=\"price price--hrk\">[^\w]*([^<]*)`
- PublishedDate:
  `<time class=\"date[^>]*>([^<]*)</time>`
- ImageUrl:
  `<img class=\"img entity-thumbnail-img.*\ssrc=\"([^\"]*)\"`

4 Extraction using XPath

To extract data with XPath we used libary **lxml** [3]. The following XPath expressions were used to extract data records from the RTV SLO website:

- Author:
  `//*[@id="main-container"]//*[@class="article-meta"]//div[@class="author-name"]/text()`
- PublishedTime:
  `//*[@id="main-container"]//*[@class="publish-meta"]/text()`
- Title:
  `//*[@id="main-container"]//header/h1/text()`
- SubTitle:
  `//*[@id="main-container"]//header/div[@class="subtitle"]/text()`
- Lead:
  `//*[@id="main-container"]//header/p[@class="lead"]/text()`
- Content:
  `//*[@id="main-container"]//div[@class="article-body"]//*[not(self::script)]/text() [normalize-space()]`

Overstock website was a little trickier because it only consisted of tables with almost no properties or classes or distinct identifiers. We needed to iterate trough rows and for every row get its title and its content. When we come to two rows without the title we stop and return data. The following list of XPath expressions were used to extract data records from the Overstock website, where index **{i}** represents **i-th** data row:

- Title:
  `//table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[@bgcolor][{i}]/td[2]/a//text()`
- ListPrice:
  `//table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[@bgcolor][{i}]/td[2]/table//table//tr[1]/td[2]//text()[normalize-space()]`
- Price:
  `//table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[@bgcolor][{i}]/td[2]/table//table//tr[2]/td[2]//text()[normalize-space()]`
- Saving/SavingPercent:
  `//table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[@bgcolor][{i}]/td[2]/table//table//tr[3]/td[2]/span//text()`
- Content:
  `//table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[@bgcolor][{i}]/td[2]/table/tbody/tr/td[2]//text()[normalize-space()]`

The following list of regular expressions were used to extract data records from the Bolha.com website:

- Title:
  `//div[@class="content-main"]//ul/li/article//h3[@class="entity-title"]//text()`
- Description:
  `//div[@class="content-main"]//ul/li/article//div[@class="entity-description-main"]//text()[normalize-space()]`
- Price:
  `//div[@class="content-main"]//ul/li/article//div[@class="entity-prices"]//strong[@class="price price--hrk"]/text()[normalize-space()]`
- PublishedDate:
  `//div[@class="content-main"]//ul/li/article//div[@class="entity-pub-date"]/time/text()[normalize-space()]`
- ImageUrl:
  `//div[@class="content-main"]//ul/li/article/div[@class="entity-thumbnail"]/a/img/@src`

5 Automatic Web extraction

The main idea for the implementation of automatic web extraction algorithm was taken from a Webmaster implementation [4] with some adaptations. The main differences are apparent, where the original algorithm analyzed multiple pages and applied clustering, this implementation compares two web pages at a time. The pseudocode is shown below.

```
diff_score(string1,string2):
    MCS = max_common_sequence(string1, string2
    return (len(string1) + len(string2) - 2* len(MCS)) / (len(string1) + len(string2))

Webmaster(pages)
    keep_tags = list['p', 'div', 'h1', 'title', 'table']
    layouts = list[]
    tag_content = list[]
    diff_threshold = 0.25

```

```
    for all pages:
        # used BeautifulSoup
        layouts <- insert(get_tags_corresponding_to(keep_tags, page)
        tag_content <- insert(get_content_from(keep_tags, page)

    # compare similarity between pages
    document_diff = levensteinDistance(layouts[0], layouts[1])

    # generate a common layout pattern
    for i = all indexes of layout:
        if layouts[0][i] != layouts[1][i]:
            remove(layouts[0][i], layouts[1][i])
            remove(tag_content[0][i], tag_content[1][i])

    # remove banners and navigation links
    for i = all indexes of layout
        if layouts[0][i] == layouts[1][i]:
            d = diff_score(tag_content[0][i], tag_content[1][i])
            if d < diff_treshold:
                remove(layouts[0][i], layouts[1][i])
                remove(tag_content[0][i], tag_content[1][i])

    print_results(document_diff, layouts, tag_content)
```

First the HTML pages are decomposed into a sequence of layout blocks, where we specify which tags will be taken into account in line 6 of the pseudocode and then extract tags and their contents in line 13 and 14. We use the BeautifulSoup library for extraction of tags and contents [5]. The next thing we do is to compute the distance measure between the two pages. Levenstein distance is used for this. The original algorithm uses a similar method to determine, which set of pages is similar enough to be in the same cluster. Here, we only have two pages, so clustering is not needed. The code from line 20 to 23 extracts common layout blocks and generates a common layout pattern. Next, we remove the portions of the layout pattern, where the contents of tags are similar enough and are considered static and not important to the pattern. This is done in lines 26 to 31 and corresponds to the section of removing banners and navigation links in the original algorithm. To determine the difference between contents of tags in order to remove those, which are too similar, we use **diff_score** which is based on the same measure from the original algorithm. The threshold value is set to 0.25. It is defined in lines 1 to three in the original algorithm. We skip the step of obtaining titles and main text, because it is originally done to determine importance of clusters, and since there are only two pages, clusters are not important. Finally, we print the results as shown in the three examples below.

- Wraper for the RTV SLO website website:

  # pages included:
  # rtvslo.si/Audi A6 50 TDI quattro$_n emirvpremijskemrazredu - RTVSLO.si.html$
  $\#rtvslo.si/VolvoXC40D4AWDmomentum_suverenomednajboljsevrazredu - RTVSLO.si.html$
  $initial\_document\_difference$: 0.9690522243713733
  $(diff\_score, blocktag, blockfeature)$
  $(0.7037037037037037, title, AudiA650TDIquattro : nemirvpremijskemrazredu - RTVSLO.si)$
  $(0.657563025210084, div, < divstyle = "position : absolute; top : -10000px; width : 0px; height : 0px;" > ...)$
  $(0.6217616580310881, div, < div >< iframeallow = "encrypted - media" allowfullscreen = "true" allowtra...)$
  $(0.6266666666666667, div, < iframeallow = "encrypted - media" allowfullscreen = "true" allowtranspar...)$
  $(0.6743119266055045, div, < ahref = "https : //4d.rtvslo.si/zivo/tvs1" target = "_blank" title = "TVSL...)$
  $(0.6743119266055045, div, < ahref = "https : //4d.rtvslo.si/zivo/tvs2" target = "_blank" title = "TVSL...)$
  $(0.6743119266055045, div, < ahref = "https : //4d.rtvslo.si/zivo/tvs3" target = "_blank" title = "TVSL...)$
  $(0.6625514403292181, div, < ahref = "https : //4d.rtvslo.si/zivo/tvkp" target = "_blank" title = "TVKo...)$
  $(0.68, div, < ahref = "https : //4d.rtvslo.si/zivo/tvmb" target = "_blank" title = "TVMa...)$
  $(0.6790697674418604, div, < ahref = "https : //4d.rtvslo.si/zivo/tvmmc" target = "_blank" title = "TVM...)$
  $(0.658008658008658, div, < ahref = "https : //4d.rtvslo.si/zivo/ra1" target = "_blank" title = "Radio...)$

(0.6570247933884298, div, < ahref = "https : //4d.rtvslo.si/zivo/val202" target = "_blank" title = "Ra...)
(0.6607929515418502, div, < ahref = "https : //4d.rtvslo.si/zivo/ars" target = "_blank" title = "Radio...)
(0.6567796610169492, div, < ahref = "https : //4d.rtvslo.si/zivo/rakp" target = "_blank" title = "Radi...)
(0.6504424778761062, div, < ahref = "https : //4d.rtvslo.si/zivo/rasi" target = "_blank" title = "Radi...)
(0.6652719665271967, div, < ahref = "https : //4d.rtvslo.si/zivo/ramb" target = "_blank" title = "Radi...)
(0.65234375, div, < ahref = "https : //4d.rtvslo.si/zivo/capo" target = "_blank" title = "Radi...)
(0.6635071090047393, div, < ahref = "https : //4d.rtvslo.si/zivo/rammr" target = "_blank" title = "MMR...)
(1.0, div, Testnovegeneracije)
(0.9428571428571428, div, Ljubljana − MMCRTVSLO)
(1.0, div, MihaMerljak)
(0.875, div, 28.december2018ob08 : 51)
(1.0, div, Oglas)
(1.0, div, (c)2000 − 2019GemiusSAversion2.0 : /rtvslo.si/300x250, 2019 − KRKA − OUDR)
(1.0, div, < imgheight = "1" src = "./AudiA650TDIquattro_nemirvpremijskemrazredu − R...)
(0.9927536231884058, p, evampogledna Audijevspisekdodatneopremeneodvzamevoljedoivljenja, ...)
(0.9980988593155894, p, Enakoveljazaseksiluizinteligentnomatrinoosvetlitvijo, pazaportn...)
(0.9968503937007874, p, < iframeallowfullscreen = "" border = "0" frameborder = "0" height = "350" src...)
(0.9932885906040269, p, < strong > Kljunitehninipodatki :< /strong >)
(0.9881656804733728, p, −natestuAudiA650TDIquattrotiptronic)
(0.6875, p, < strong > Mere :< /strong >)
(0.9285714285714286, p, < strong > Pogon :< /strong >)
(1.0, div, Oglas)
(1.0, div, Oglas)
(1.0, div, Adformpublishertag)
(0.972027972027972, div, < imgheight = "1" src = "./AudiA650TDIquattro_nemirvpremijskemrazredu − R...)
(0.9855072463768116, div, Oglas)
(1.0, div, Prikazujkomentarje)

WraperfortheOverstockwebsite :

#pagesincluded :
#overstock.com/jewelry01.html
#overstock.com/jewelry02.html
initial_document_difference : 0.6140350877192983
(diff_score, blocktag, blockfeature)
(0.8298555377207063, table, < tbody >< tr >< tdalign = "left" >< table >< tbody >< tr >< td >< spanclass = "litt...)
(0.3586666666666667, table, < tbody >< tr >< td >< spanclass = "littlewhite" > Search :< /span > < selectnam...)
(0.9993669105233539, table, < tbody >< tr >< tdbgcolor = "000000" valign = "top" width = "1" >< imgborder = "...)
(0.9383321653819201, table, < tbody >< tr >< td >< table >< tbody >< tr >< td > < /td >< td >< spanclass = "little...)
(0.9557007988380537, table, < tbody >< tr >< td > < /td >< td >< spanclass = "littleorange" >< b > Diamonds < /b...)
(0.99383231311445, table, < tbody >< tr >< tdalign = "center" >< br/ >< tableborder = "1" cellpadding = "0"...)
(0.8249708284714119, table, < tbody >< tr >< td >< tablebgcolor = "d2dbfb" border = "0" cellpadding = "0" ce...)
(0.986464073618319, table, < tbody >< tr >< tdvalign = "top" >< tableborder = "0" cellpadding = "2" cellspa...)
(0.7115384615384616, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.918200408997955, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right" nowr...)
(0.6972111553784861, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.8599078341013825, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right" nowr...)
(0.9935897435897436, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)

(0.8744038155802861, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(0.9935897435897436, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.8933333333333333, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(1.0, table, < tbody >< tr >< tdalign = "right" >< ahref = "http : //www.overstock.com/cgi − ...)
(0.7106109324758842, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.9144215530903328, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(0.691683569979716, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.8731218697829716, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(0.9936102236421726, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.8962457337883959, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(1.0, table, < tbody >< tr >< tdalign = "right" >< ahref = "http : //www.overstock.com/cgi − ...)
(0.7110754414125201, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.9087323943661972, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)
(0.9157566302652106, table, < tbody >< tr >< td >< ahref = "http : //www.overstock.com/cgi−bin/d2.cgi?PA...)
(0.9977900552486187, table, < tbody >< tr >< tdvalign = "top" >< table >< tbody >< tr >< tdalign = "right"nowr...)

- Wraper for the Bolha.com website :

pages included:
bolha.com/Nogomet.html
bolha.com/Macke.html
initial_document_difference: 0.9790356394129979
(diff$_s$core, blocktag, blockfeature)
(0.8333333333333334, title, Nogomet)
(0.40594059405940597, div, < iframename = "$_c$mpLocator"src = "./Nogomet$_f$iles/saved$_r$esource.htm...)
(1.0, div, < divid = "google_ads_iframe_/4099697/dfp$_w$allpaper_njuskalo$_{0_c}$ont...)
(1.0, div, < divid = "google_ads_iframe_/4099697/dfp$_b$illboard_njuskalo$_{0_c}$ont...)
(0.9846153846153847, div, ProdampopolnomanovefantovskenogometnesupergeAdidasPREDATORTANGO18...)
(1.0, div, < divid = "google_ads_iframe_/4099697/dfp_list_top_njuskalo_0$_c$onta...)
(0.9655172413793104, div, DresRooney(ManchesterUnited))
(0.9777777777777777, div, kotnove, noenenekajkratvtelovadnicinotranjadolina21cm)
(0.9770114942528736, div, istonovamajicaFiorentine.tevilkaM.Kontakt − 031559248)
(0.9846153846153847, div, Vsestranskekratkehlae AdidasClima365, primernezarazlineportneakt...)
(0.9844961240310077, div, ProdamnoveAdidasNemezizTango17 + 360Agilityvelikosti42, naroeniprek...)
(1.0, div, < divid = "google_ads_iframe_/4099697/dfp_middle1_njuskalo$_{0_c}$ontai...)
(0.9846153846153847, div, ModriMessiAdidas, vrhunski, obutisamodvakratnatreningu, ohranjenisko...)
(0.978021978021978, div, OtrokaFelpaUdinesezgornjidel.140 − 152CMKONTAKT : 041713270)
(0.9753086419753086, div, OtrokikompletMilanza11/12let.KONTAKT − 041713270)
(0.975, div, TrenerkaEmpoliOtrokado152CM.KONTAKT − 041713270)
(0.9770114942528736, div, OtrokavetrovkaErreatevilka34in38.KONTAKT − 041713270)
(0.9771863117870723, div, < divid = "google_ads_iframe_/4099697/dfp_list_middle_njuskalo_0$_c$o...)
(0.972972972972973, div, VetrovkaDiadoraotrokiXL.KONTAKT − 041713270)
(0.981651376146789, div, NovavetrovkaAsics.tevilke, kisonavoljoM, L, 164CM, 152CM.KONTAKT − 041...)
(0.975609756097561, div, OtrokavetrovkaErreatevilka36KONTAKT : 041713270)

(0.9759036144578314, div, $OtrokaFelpazatreningtevilka34.KONTAKT - 041713270$)
(0.971830985915493, div, $Otrokehlakedo152cm.KONTAKT - 041713270$)
(0.9736842105263158, div, $GornjidelMilanza11/12let.KONTAKT041713270$)
(0.975, div, $OtrokavetrovkaMilanza7/8let.KONTAKT - 041713270$)
(0.9746835443037974, div, $NovavetrovkaErreatevilkaS, L.KONTAKT - 041713270$)
(0.9805825242718447, div, $NovaAsicstrenerkazatreningaliprostiastevilkaM.KONTAKT - 041713270$)
(0.9846153846153847, div, $Adidasbeli11PRO, usnjeni, uporabljani2meseca, vrhunski, mehki, skrbnov...$)
(0.9733333333333334, div, $Prodamnikolinoenenogometnekopaket.43.5.$)
(0.9743589743589743, div, $ProdamnogometnekopakeNiket41.Nikolinoene.$)
(0.9846153846153847, div, $ProdamodlinoohranjeneotrokekopakezameanopodlagoPumaFuture18.4...$)
(0.9846153846153847, div, $Prodam2robustnagolazmreo, dimenzij120x80cm, masivneizdelave.Cenaza...$)
(0.9771863117870723, div, $JomaTopFlexmokedvoranskecopatezafutsal/malinogomet, t.42Osebnipr...$)
(1.0, div, $< divid = "google\_ads\_iframe\_/4099697/dfp\_list\_bottom\_njuskalo\_0\_{c}o...$)
(1.0, div, $Izberi$)
(1.0, div, $< divaria - expanded = "false" class = "selectr - selected" disabled = "undefi...$)
(1.0, div, $< ulclass = "selectr - labelselectr - tags" >< /ul >$)
(1.0, div, $--$)
(1.0, div, $< divclass = "selectr - input - container" >< inputautocapitalize = "off" aut...$)
(1.0, div, $< inputautocapitalize = "off" autocomplete = "off" autocorrect = "off" clas...$)
(0.9772727272727273, div, $< ularia - expanded = "false" aria - hidden = "true" class = "selectr - options"...$)
(1.0, div, do)
(1.0, div, $< divid = "google\_ads\_iframe\_/4099697/dfp_{s}kyscraper\_njuskalo\_0\_{c}on...$)
(1.0, div, $< divid = "google\_ads\_iframe\_/4099697/dfp_{r}ectangle\_njuskalo\_0\_{c}ont...$)
(0.8007662835249042, div, $OutletkeramineploiceCOLLAGELENOMUSGOsonazalogi.ifra : 04394Mode...$)
(0.9849624060150376, div, $Starnekajmesecev, samopreiskuen, enalepkanadisplayu.Zaradinakupa1...$)
(0.9863013698630136, div, $Prodamsuperohranjen6560b, vkljucnospriklopnopostajo.Specifikacije : ...$)
(0.9859154929577465, div, $Znamka : victoria's secret Mesto : Ljubljana Materiali : xxx Oznake : tigervz...$)
(0.9864864864864865, div, $KUPIMTRAKTOR, KATEREKOLIZNAMKE, LAHKOTUDIPOKODOVAN, VOKVARI, BRI$
(0.986013986013986, div, $Mesto : Ljubljana Materiali : bomba Oznake : higienamesenoperiloitnik...$)
(1.0, div, $< divid = "google\_ads\_iframe\_/4099697/dfp\_halfpage\_njuskalo\_0\_{c}onta...$)

In the above examples, the final wrapper is presented as tuples of diff_scores, block tags and block features, separated by commas.

References

[1] Styria digital marketplaces d.o.o. Bolha. `https://www.bolha.com/`. Accessed: 2020-04-30.

[2] Python. Regular expression operations. `https://docs.python.org/3/library/re.html`. Accessed: 2020-04-30.

[3] Stephan Richter. lxml - xml and html with python. `https://lxml.de/`. Accessed: 2020-04-30.

[4] The webmaster algorithm. `http://www.unixuser.org/~euske/python/webstemmer/howitworks.html`. Accessed: 2020-05-01.

[5] The beautiful soup library. `https://www.crummy.com/software/BeautifulSoup`. Accessed: 2020-05-01.