[[1]](#footnote-1)

Poročilo druge seminarske naloge  
Mark Bogataj in Jakob Maležič  
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani  
Ljubljana, Slovenija  
Mentor: asist. prof. dr. Slavko Žitnik

# Opis izbranih spletnih strani

Za dodatni spletni strani sva si izbrala dve strani iz novičarske spletne strani [Žurnal24](https://www.zurnal24.si/). Pri izbiranju strani sva se odločila za dve, ki opisujeta avtomobile, podobno kot strani iz domene rtvslo, saj naju je zanimala razlika med stranema, ki sicer izgledata zelo podobno, ampak po sami strukturi pa se precej razlikujeta. Primer strani je prikazan na sliki 1.

Iz spletnih strani domene zurnal24 sva izluščila naslednje podatke:

* author - ime avtorja, ki je napisal novico
* publishedTime - datum in čas objave novice
* title - naslov novice
* viewCount - število ogledov novice
* lead – uvod
* content - samo tekstovna vsebina

# Ekstrakcija z uporabo regularnih izrazov

## Overstock

Regularni izraz za ekstrakcijo podatkov iz spletnih strani Overstock je sledeč:

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

Večina delov regularnega izraza je zavitih še v oklepaj z vprašajem na koncu, kar pomeni, da je lahko ta del opcijski. S tem regularni izraz izlušči vsebino tudi, če katera od cen ni navedena. To sva tudi testirala tako, da sva iz html kode odstranila značko, ki vsebuje ceno.

## Rtvslo

Regularni izraz za ekstrakcijo podatkov iz spletnih strani Rtvslo je sledeč:

<h1>(.\*)</h1>\n<div class="subtitle">(.\*)</div>[\s\S]\*<p class="lead">(.\*)</p>[\s\S]\*<div class="author">[\s\S]\*<div class="author-name">(.\*)</div>[\s\S]\*<div class="publish-meta">\n[\W]\*(.\*)<br/>[\s\S]\*</figure>[\n]\*<p(?:[\s\r]+[^>]\*)?>(.\*)</p>

Na koncu sva dobljeno vsebino počistila na ta način, da sva odstranila vse html značke, ki so ostale v vsebini. To sva naredila z uporabo regularnega izraza <[^>]\*>.

Graphical user interface, application

Description automatically generated

Slika 1: Primer strani z označenimi podatki, ki smo jih izluščili

## Žurnal24

Regularni izraz za ekstrakcijo podatkov iz spletnih strani Žurnal24 je sledeč:

<span class="article\_\_views">[\n]\*<i.\*>.\*</i>[\n]\*<strong>(.\*)</strong>[\s\S]\*<h1 class="article\_\_title">(.\*)</h1>[\n]\*<div class="article\_\_authors">[\s\S]\*<a href=.\*>(.\*)</a>[\n]\*</div>[\n]\*<time class="article\_\_time">(.\*)</time>[\s\S]\*<div class="article\_\_leadtext">[\n]\*(.\*)[\n]\*</div>[\s\S]\*<div class="article\_\_content no\_page\_break cf" .\*>\n([\s\S]\*)</div>[\n]\*<div class="text-center fold\_article\_\_bellow\_content">

Na koncu sva dobljeno vsebino ponovno počistila in odstranila vse html značke, ki so ostale v vsebini. To sva naredila z uporabo regularnega izraza: <[^>]\*>.

# Ekstrakcija z uporabo xpath

## Overstock

Najprej z xpath izrazom izberemo vse kartice, ki imajo ozadje določene barve in imajo 2 <td> otroka z atributom @valign='top': cards = treeContent.xpath("//tbody/ tr[(contains(@bgcolor, '#ffffff') or contains(@bgcolor, '#dddddd')) and count(td[@valign='top'])=2]"). Nato iz vsakega element iz seznama kartic izluščimo informacije:

* card.xpath("td[@valign='top']/a/b/text()")[0]
* card.xpath("td[@valign='top']/table/tbody/tr/td[1]/table/tbody/tr[1]/td[2]/s/text()")[0]
* card.xpath("td[@valign='top']/table/tbody/tr/td[1]/table/tbod/tr[2]/td[2]/span/b/text()")[0]
* card.xpath("substring-before(td[@valign='top']/table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span/text(), ' (')")
* card.xpath("substring-before(substring-after(td[@valign=\'top\']/table/tbody/tr/td[1]/table/tbody/tr[3]/td[2]/span/text(), '('), ')')")
* card.xpath("td[@valign='top']/table/tbody/tr/td[2]/span/text()")[0]

Na koncu vsebino pridobljeno z zadnjim izrazom (opis) uredimo tako, da znak za novo vrstico zamenjamo s predledkom.

## Rtvslo

Xpath izrazi za ekstrakcijo podatkov iz strani rtvslo so:

* treeContent.xpath('//\*[@id="main-container"]/div[3]/div/header/h1/text()')[0]
* treeContent.xpath('//\*[@id="main-container"]/div[3]/div/header/div[2]/text()')[0]
* treeContent.xpath('//\*[@id="main-container"]/div[3]/div/div[1]/div[1]/div/text()')[0]
* treeContent.xpath('normalize-space(//\*[@id="main-container"]/div[3]/div/div[1]/div[2]/text())')
* treeContent.xpath('//\*[@id="main-container"]/div[3]/div/header/p/text()')[0]
* treeContent.xpath('//\*[@id="main-container"]/div[3]/div/div[2]/article/p/text() | //\*[@id="main-container"]/div[3]/div/div[2]/article//strong/text()')

## Žurnal24

Pri ekstrakciji podatkov iz strani Žurnal24 najprej izluščimo glavo članka iz katerega bom razbrali število ogledov, naslov, avtorja in čas objave:

* treeContent.xpath('//article/header')[0]
* articleHeader.xpath('div[1]/span/strong/text()')[0]
* articleHeader.xpath('h1/text()')[0]
* articleHeader.xpath('div[2]/a/text()')[0]
* articleHeader.xpath('time/text()')[0]

Uvod in vsebino pa izluščimo z naslednjima izrazoma:

* treeContent.xpath('//article/div/div[2]/div/div[3]/text()')[0]
* treeContent.xpath('//article/div/div[2]/div/div[5]//text()[normalize-space(.)]')

# Implementacija avtomatizirane ekstrakcije

TODO JAKOB

1. . [↑](#footnote-ref-1)