FRI Web information extraction and retrieval 2022
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

Structured data extraction

Matjaž Zupanič, Jure Tič, Miha Malenšek

Abstract

In this report we will describe our Python implementation of three different approaches to structured data extraction from the Web - using regular expressions, using XPath, and *RoadRunner*-like Wrapper implementation or other automatic content extraction.

Keywords

web-crawling, Python, XPath, regex, automatic content extraction

Advisors: Slavko Žitnik

Chosen Web pages

We chose two Žurnal24.si pages for processing. We recovered article topics, the number of views, its title, author, time and date of publishing, the article's lead, and content. All data items are marked in Figure 1.

Regex expressions

This section lists all Regex expressions i used to obtain required data. The content obtained with these expressions was then further processed to remove unwanted elements such as scripts and tabulator signs

0.1 rtvslo.si

- **Author:**

  ```
  '<div class="author-name">([\s\S]*?)</div>'
  ```

- **Published Time:**

  ```
  '<div class=\"publish-meta\">([\s\S]*?)<br>'
  ```

- **Title:**

  ```
  '<h1>([\s\S]*?)<\/h1>'
  ```

- **Subtitle:**

  ```
  '<div class=\"subtitle\">([\s\S]*?)<\/div>'
  ```

- **Lead:**

  ```
  '<p class=\"lead\">([\s\S]*?)<\/p>'
  ```

- **Content:**

  ```
  '<div class=\"article-body\">([\s\S]*?)
  <div class=\"article-column\">'
  ```

0.2 overstock.com

- **List Price:**

  ```
  '(?<=(<b>List Price:<\/b><\/td><td nowrap=\"nowrap\"
  align=\"left\"><s>))([\s\S]*?)(?=(<\/s><\/td><\/tr>))'
  ```

- **Price:**

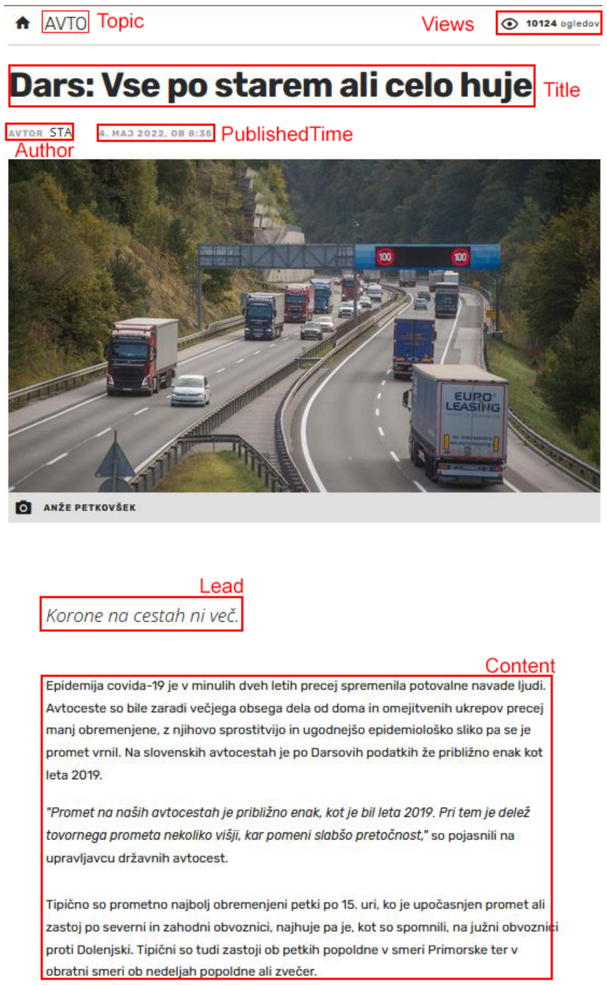
  ```
  '(?<=(<b>Price:<\/b>
  <\/td><td nowrap=\"nowrap\" align=\"left\">
  <span class=\"bigred\"><b>))([\s\S]*?)
  (?=<\/b><\/span><\/td><\/tr>)'
  ```

- **Saving:**

  ```
  '(?<=(<b>You Save:<\/b>
  <\/td><td nowrap=\"nowrap\"
  align=\"left\">
  <span class=\"littleorange\">))([\s\S]*?)(?=\()'
  ```

- **Saving percent:**

  ```
  '((<b>You Save:<\/b><\/td>
  <td nowrap=\"nowrap\" align=\"left\">
  <span class=\"littleorange\">
  ([\s\S]*?)\())
  ([\s\S]*?)(?=(\)<\/span><\/td><\/tr>))'
  ```



Dars: Vse po starem ali celo huje

Korone na cestah ni več.

Epidemija covida-19 je v minulih dveh letih precej spremenila potovalne navade ljudi. Avtoceste so bile zaradi večjega obsega dela od doma in omejitvenih ukrepov precej manj obremenjene, z njihovo sprostitvijo in ugodnejšo epidemiološko sliko pa se je promet vrnil. Na slovenskih avtocestah je po Darsovih podatkih že približno enak kot leta 2019.

"Promet na naših avtocestah je približno enak, kot je bil leta 2019. Pri tem je delež tovornega prometa nekoliko višji, kar pomeni slabšo pretočnost," so pojasnili na upravljavcu državnih avtocest.

Tipično so prometno najbolj obremenjeni petki po 15. uri, ko je upočasnjen promet ali zastoj po severni in zahodni obvoznici, najhuje pa je, kot so spomnili, na južni obvoznici proti Dolenjski. Tipični so tudi zastoji ob petkih popoldne v smeri Primorske ter v obratni smeri ob nedeljah popoldne ali zvečer.

Figure 1. Identification of data items and data records.

- **Title:**
```
'(</tbody></table></td><td valign=\"top\">
\n<a href=\"(.*?)\">
<b>)([\s\S]*?)(?=</b></a><br> )'
```
- **Content:**
```
'(?<=(<td valign=\"top\"><span class=\"normal\">))
([\s\S]*?)(?=<br><a href=)'
```

0.3 zurnal24.si

- **Author:**
```
'<div class="article__authors">([\s\S]*?)
<a href="([\s\S]*?)">([\s\S]*?)<\/a>'
```
- **Header:**
```
'<h1 class="article__title">([\s\S]*?)<\/h1>'
```
- **Date:**
```
'<time class="article__time">([\s\S]*?)<\/time>'
```
- **Ogledi:**
```
'<span class="article__views">(\s*)
<i class="icon-eye-outline"><\/i>(\s*)
<strong>([\s\S]*?)<\/strong>'
```
- **Lead:**
```
'<div class="article__leadtext">
([\s\S]*?)<\/div>'
```
- **Content:**
```
'<div class="article__content ([\s\S]*?)>([\s\S]*?)
<div class="article__aditionl_content">'
```

XPath expressions

This section lists all XPath expressions used with data extraction. The results of XPath queries were further processed to remove additional text tags *(such as newline tags, tab tags, etc.)*.

0.4 rtvslo.si

- **Author:** *'//div[@class="author-name"]/text()'*
- **Published Time:** *'//div[@class="publish-meta"]/text()'*
- **Title:** *'//header[@class="article-header"]/h1/text()'*
- **Title:** *'//div[@class="subtitle"]/text()'*
- **Subtitle:** *'//p[@class="lead"]/text()'*
- **Lead:** *'//p[@class="lead"]/text()'*
- **Content:** *'//div[@class="article-body"]/descendant::text()'*

0.5 overstock.com

Overstock was a messier html page, for which we prepared a "base" XPath, to avoid long expressions. The base path, stored in the *"xpath_to_elements"* variable, was used as a base expression to which we appended the rest of the XPath expressions to. Due to the requested data formatting, we had to create an additional "base" XPath expression, named *"xpath_to_element"*, which selects specific list elements.

- **Base XPath expression:** *'//body/table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr[*]/td[2]'*
- **Additional XPath expression:** *'//body/table[2]/tbody/tr[1]/td[5]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr'*
- **Titles:** *xpath_to_elements + '/a/b/text()'*
- **List Prices:** *xpath_to_elements + '/table/tbody/tr/td[1]/table/tbody/tr[1]/td[2]/s/text()'*
- **Prices:** *xpath_to_elements + '/table/tbody/tr/td[1]/table/tbody/tr[2]/td[2]/span/b/text()'*

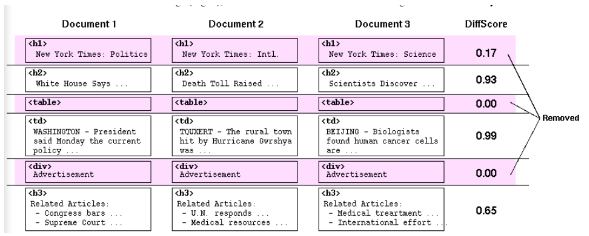
- **Saving:** *’substring-before(’ + xpath_to_element + ’[@*][’ + str(i+1) + ’]/td[2]/table/tbody/tr/td[1]/table /tbody/tr[3]/td[2]/span/text(), ” ”)’*
- **Saving Percent:** *’substring-after(’ + xpath_to_element + ’[@*][’ + str(i+1) + ’]/td[2]/table/tbody/tr/td[1]/table /tbody/tr[3]/td[2]/span/text(), ” ”)’*

0.6 zurnal24.si

- **Topics:** *’//div[@class=”article__sections”]/a/text()’*
- **Views:** *’//span[@class=”article__views”]/strong/text()’*
- **Title:** *’//h1[@class=”article__title”]/text()’*
- **Author:** *’//div[@class=”article__authors”]/a/text()’*
- **Published Time:** *’//time[@class=”article__time_small”]/text()’*
- **Lead:** *’//div[@class=”article__leadtext”]/text()’*
- **Content:** *’//div[@class=”article__content no_page_break cf”]/p/text() l //div[@class=”article__content no_page_break cf”]/h2/text()’*

Automatic Web extraction Webstemmer

For automatic webpage exctractrion we chose webStemmer [1] like approach. We briefly followed algorithm proposed on webpage, but because of some uncertainties we did some parts in our own way. First we parsed the page into DOM tree with BeautifulSoup by tags. Then we decomposed tree into list of blocks and we only kept those which had some text inside since other are not relevant for information retrieval. The next phase was aligning phase. In that phase we aligned tags of both web pages. In order to align tags and compensate for redundant tags we searched for same sequence of text in both web pages. Text had to be long enough so we are


Document 1
Document 2
Document 3
DiffScore
<h1> New York Times: Politics
<h1> New York Times: Intl.
<h1> New York Times: Science
0.17
<h2> White House Says ...
<h2> Death Toll Raised ...
<h2> Scientists Discover ...
0.93
<table>
<table>
<table>
0.00
<td> WASHINGTON - President said Monday the current policy ...
<td> TQUXERT - The rural town hit by Hurricane Gwrshya was ...
<td> BEIJING - Biologists found human cancer cells are ...
0.99
<div> Advertisement
<div> Advertisement
<div> Advertisement
0.00
<h3> Related Articles: - Congress bars ... - Supreme Court ...
<h3> Related Articles: - U.N. responds ... - Medical resources ...
<h3> Related Articles: - Medical treartment ... - International effort ...
0.65
Removed
Figure 3. Aligning html blocks in webStemmer like approach

certain in correctness of alignment - we used dictionary data structure. Then we filled missing tags. Now we got structure shown on Image 2. After that we removed tags that didn’t match to create common layout blocks Then we calculated distance between Calculating distance-we used Levenstain distance and normalize it over string length. Then we removed elements that were to similar, we took threshold of 10%, as shown in picture 3. Remaining block are our wrapper.

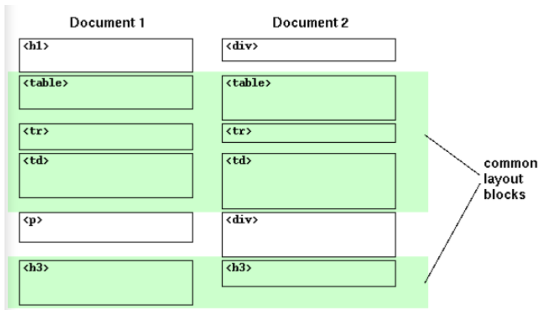


Figure 2. Aligning html blocks in webStemmer like approach

References

[1] Webstemmer - how it works. https://www.unixuser.org/~euske/python/webstemmer/howitworks.html. Accessed: 2022-5-5.