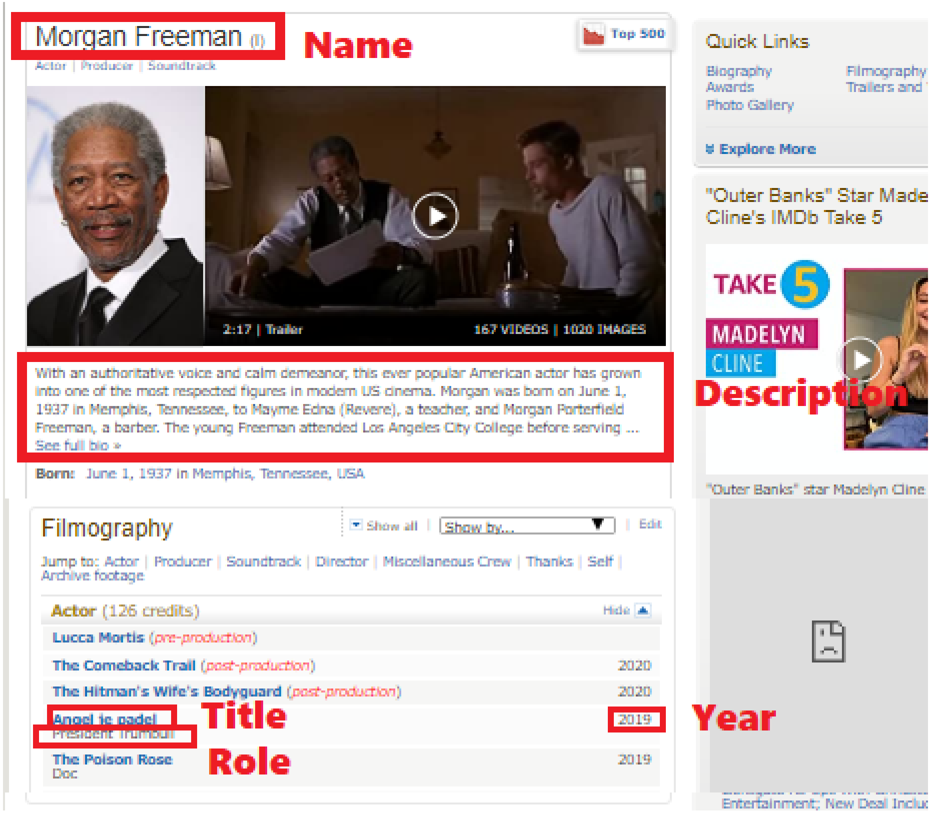
Coding assignment 2

Web information extraction and retrieval

Jošt Gombač, Marko Krajinović, Rok Petrovčič Vižintin

1 Page selection

Figure 1: imdb.com page data selection



With an authoritative voice and calm demeanor, this ever popular American actor has grown into one of the most respected figures in modern US cinema. Morgan was born on June 1, 1937 in Memphis, Tennessee, to Mayme Edna (Revere), a teacher, and Morgan Porterfield Freeman, a barber. The young Freeman attended Los Angeles City College before serving ...
See full bio »

Born: June 1, 1937 in Memphis, Tennessee, USA

Actor (126 credits)

Lucca Mortis (pre-production)

The Comeback Trail (post-production) 2020

The Hitman's Wife's Bodyguard (post-production) 2020

Angel is padel 2019
President Trumbull

The Poison Rose 2019
Doc

We selected the name and description of the actors. The page also includes a list of films they have worked on (as an actor, producer or other role). For each film we extract its title, year and actors role in the film.

2 XPath

2.1 rtvslo.si

```
author = //div[@class='author-name']/text()
published_time = //div[@class='publish-meta']/text()
title = //h1/text()
subtitle = //div[@class='subtitle']/text()
lead = //p[@class='lead']/text()
content = //div[@class='article-body']//text()[not(ancestor::figure)]
```

We additionally manipulate date to proper format.

2.2 overstock.com

```
rows = //table[@border='0' and @cellpadding='0' and @cellspacing='0'
      and @width='100%']//tr[@bgcolor]
for each row do:
  el = ./td[-1]
  title = el.xpath(./a/b/text())
  prices = el.xpath(./table/tbody/tr/td/table/tbody/tr/td//text())
  list_price = prices[1]
  price = prices[3]
  saving = prices[-1].split()[0]
  saving_percent = prices[-1].split()[-1]
  content = el.xpath(./table/tbody/tr/td/span/text())
```

2.3 imdb.com

```
name = //td[@class='name-overview-widget__section']
      /h1/span[@class='itemprop']/text()
description = //div[@class='name-trivia-bio-text']/div[@class='inline']
      //text()[not(ancestor::span)]"
films = //div[contains(@class, 'filmo-row')]

for each film do:
  title = film.xpath(./b//text())
  year = film.xpath(./span[@class='year_column']/text())
  role = film.xpath("./text()")
```

Each film role selection captured different number of elements as this field was not structured well. We implemented some post-processing checks to insure the right data was captured.

3 Regular expressions

3.1 rtvslo.si

```
author = <div class="author-name">(.*?)</div>
published_time = <div class="publish-meta">(.*?)<br>
title = <h1>(.*?)</h1>
subtitle = <div class="subtitle">(.*?)</div>
lead = <p class="lead">(.*?)</p>
content = <article class="article">(.*?)</article>
remove_figures = <figure .*?>.*?</figure>
content = content.replace(remove_figures, "")
```

3.2 overstock.com

```
rows = <tr bgcolor=.*?>
  <td valign="top" align="center">
  <table><tbody><tr><td><a href=.*?><img src=.*? border="0"
  width="80" height="80"></a></td></tr>
  <tr><td><a href=.*?>More Info...</a></td></tr>
  </tbody></table></td><td valign="top">
  <a href=.*?><b>(.*?)</b></a><br>
  <table><tbody><tr><td valign="top"><table>
  <tbody><tr><td align="right" nowrap="nowrap"><b>List Price:</b></td>
  <td align="left" nowrap="nowrap"><s>(.*?)</s></td></tr>
  <tr><td align="right" nowrap="nowrap"><b>Price:</b></td>
  <td align="left" nowrap="nowrap"><span
   class="bigred"><b>(.*?)</b></span></td></tr>
  <tr><td align="right" nowrap="nowrap"><b>You Save:</b></td>
  <td align="left" nowrap="nowrap"><span
   class="littleorange">(.*?)</span></td></tr>
  </tbody></table>
  </td><td valign="top"><span class="normal">(.*?)<br>
  <a href=.*?><span class="tiny"><b>Click here to
   purchase.</b></span></a></span><br>

for each row do:
  construct JSON object
```

3.3 imdb.com

```
name = <td class="name-overview-widget__section">.*?
    <h1 class="header"> <span class="itemprop">(.*?)</span>
description = <div class="name-trivia-bio-text">.*?<div
        class="inline">(.*?)<span
films = <div class="filmo-row .*?>
  <span class="year_column">
  &nbsp;(.*?)
  </span>
  <b><a href=.*?
  >(.*?)</a></b>(.*?)</?div
```

Each captured role was then post-processed to ensure proper data structure.

4 Wrapper

4.1 Description

The wrapper simultaneously iterates over the DOM of two web pages. The DOM of each page is cleaned before wrapper generation. We remove the following tags: *script*, *style*, *meta*, *link*, *noscript*, *svg*, *path*, *iframe* and *map*. We remove these, because they usually contain data irrelevant to the crawler. The user could remove tags they want to parse from the unwanted tags list if they deemed it necessary. We also remove empty strings and newlines. Each DOM element is then compared. If they match, they are added to the wrapper. If elements match in everything except their text content, that content is marked as interesting. Otherwise, the next match is found, and all unmatched elements in between get marked as optional. After a wrapper is generated, it is checked for repeating elements. Those are marked as such. the output is an almost HTML-like text file with a few differences. Optional elements are enclosed in **()?**, repeating elements are enclosed in **()+** and potentially interesting text is replaced with **#Text**.

4.2 Pseudocode

```
GenerateWrapper(nodeA, nodeB, wrapper)
  indexA = 0
  indexB = 0
  while nodeA.children.lenth > indexA and nodeB.children.length > indexB:
    elementA = nodeA.children[indexA]
    elementB = nodeB.children[indexB]
    if elementA and elementB have matching tags and a matching id proeprty:
      if elementA is text:
        if elementA.textValue == elementB.textValue
          wrapper.add(new textElement(elementA.textValue))
        else
          wrapper.add(new textElement("#text"))
        indexA++, indexB++
      else
        tagElement = new tagElement(elementA.tag, elementA.attributes)
        wrapper.add(tagElement)
        if elementA.children.length > 0
          GenerateWrapper(elementA, elementB, tagElement)
        indexA++, indexB++
    else:
      matchingIndexA = FindIndexOfNextMatch(nodeA, nodeB, indexA, indexB)
      matchingIndexB = FindIndexOfNextMatch(nodeB, nodeA, indexB, indexA)
      if matchingIndexA and matchingIndexB are None:
        add elementA and elementB as optional nodes to wrapper
      if (matchingIndexA - indexA) < (matchingIndexB - indexB):
        add elements in nodeA from indexA to matchingIndexA as optional nodes into the wrapper
        indexA = matchingIndexA
      else:
        add elements in nodeB from indexB to matchingIndexB as optional nodes into the wrapper
        indexB = matchingIndexB
  add remaining elments from the node that still has unprocessed children as optional nodes into the wrapper
  FindRepeatingNodes(wrapper)
  return wrapper

FindRepeatingNodes(wrapper)
  if wrapper has less than 2 children:
    return
  current = wrapper.children[0]
  for child in wrapper.children[1:]:
    if child and current are equal:
      mark current as repeating
      mark child as repeating
      mark child as removed, so it is not printed when we stringify the wrapper
    FindRepeatingNodes(child)
    current = child
```

4.3 Page wrapper results

Due to the relatively naive implementation of our wrapper generation algorithm and wrapper pretty-printing, the wrappers tend to be rather long as they are similar to the actual HTML of the web page. Included are only parts of each wrapper which showcase the optional, list or text elements. The full outputs are present in the repository folder *results/C*.

4.3.1 rtvslo.si

```
(<div class="iAdserver adloaded" data-iadserver-zone="299" id="f4d0659e109b98fa56aa9d5c2e67bd1f" style="text-align: center; position: absolute;"></div>)+
  <div id="main-container" class="container article-container" data-id="475392">
    <div class="section-heading blue">
      <h3 class="section-title animated-circles-onhover">
        <a href="https://www.rtvslo.si/zabava/avtomobilnost/testi/">#Text</a>
      </h3>
    </div>
    <div id="article-edit-btn" class="edit-btn-container" style="display:none">
      <a href="https://admin.rtvslo.si/?c_mod=newsadmin&op=news&func=edit&id=475392" class="edit-btn" rel="nofollow" target="_blank">Uredi</a>
    </div>
    <div class="news-container blue article-old article-type-1">
      <div class="row">
        <header class="article-header">
          <div class="section-heading blue">
            <h3 class="section-title animated-circles-onhover">
              <a href="https://www.rtvslo.si/zabava/avtomobilnost/testi/">#Text</a>
            </h3>
            <a class="comments-icon ml-auto" data-comments-number="13" href="https://www.rtvslo.si/zabava/avtomobilnost/testi/audi-a6-50-tdi-quattro-nemir-v-premijskem-razredu/475392#article-comments-anchor"></a>
          </div>
          <h1>#Text</h1>
          <div class="subtitle">#Text</div>
```

4.3.2 overstock.com

```
(<tr>
  <td colspan="2" height="4">
    <img src="jewelry01_files/ms.gif" width="1" height="4" border="0"></img>
  </td>
</tr>)?
(<tr bgcolor="#dddddd">
  <td valign="top" align="center">
    <table>
      <tbody>
        <tr>
          <td>
            <a href="http://www.overstock.com/cgi-bin/d2.cgi?PAGE=PROFRAME&PROD_ID=51595">
              <img src="jewelry01_files/T917417.jpg" border="0" width="80" height="80"></img>
            </a>
          </td>
        </tr>
        <tr>
          <td>
            <a href="http://www.overstock.com/cgi-bin/d2.cgi?PAGE=PROFRAME&PROD_ID=51595">More Info...</a>
          </td>
        </tr>
      </tbody>
    </table>
  </td>
  <td valign="top">
    <a href="http://www.overstock.com/cgi-bin/d2.cgi?PAGE=PROFRAME&PROD_ID=51595">
      <b>14-kt. Diamond Cross Pendant (.06 TW)</b>
    </a>
    <br>
    <table>
      <tbody>
        <tr>
          <td valign="top">
            <table>
              <tbody>
                <tr>
                  <td align="right" nowrap="nowrap">
                    <b>List Price:</b>
                  </td>
```

4.3.3 imdb.com

```
(<div class="filmo-row even" id="archive_footage-tt0480949">
  <span class="year_column"> 2005 </span>
  <b>
    <a href="/title/tt0480949/?ref_=nm_flmg_arf_20">Venecia 2005: Cr nica de Carlos Boyero</a>
  </b>
  (TV Short)
  <br>
  Josef
</div>)?
(<div class="filmo-row odd" id="archive_footage-tt0410407">
  <span class="year_column"> 2003 </span>
  <b>
    <a href="/title/tt0410407/?ref_=nm_flmg_arf_21">Orwell Rolls in His Grave</a>
  </b>
  (Documentary)
  <br>
  Self
</div>)?
(<div class="filmo-row even" id="archive_footage-tt0279641">
  <span class="year_column"> 2002 </span>
  <b>
    <a href="/title/tt0279641/?ref_=nm_flmg_arf_22">Who Is Alan Smithee?</a>
  </b>
  (TV Movie documentary)
  <br>
  Self (uncredited)
</div>)?
(<div class="filmo-row odd" id="archive_footage-tt0296727">
  <span class="year_column"> 2000 </span>
  <b>
    <a href="/title/tt0296727/?ref_=nm_flmg_arf_23">Lord Stanley's Cup: Hockey's Ultimate Prize</a>
  </b>
  (Video documentary)
  <br>
  Self
</div>)?
```