

732A92/TDDE16 Text Mining

Introduction to Lab L1: Information Retrieval

Huanyu Li

Agenda

- Aims of this lab
- Lab task
- Instruction
- Notes
- Work on jupyter notebook

Aims

- To capture basic web crawling technique
- Get start to work with basic NLP techniques
- To implement vector model for storing and querying web information

Lab task --- To implement an Android app search engine

- Build a storage for data that describe a number of apps
 - Crawl the app store for Android app
 - Obtain the descriptive text for each app
 - Store the data locally
- Build a Information Retrieval model
 - Vector Model
 - Tf-idf
- Ranked query processor
 - Cosine similarity

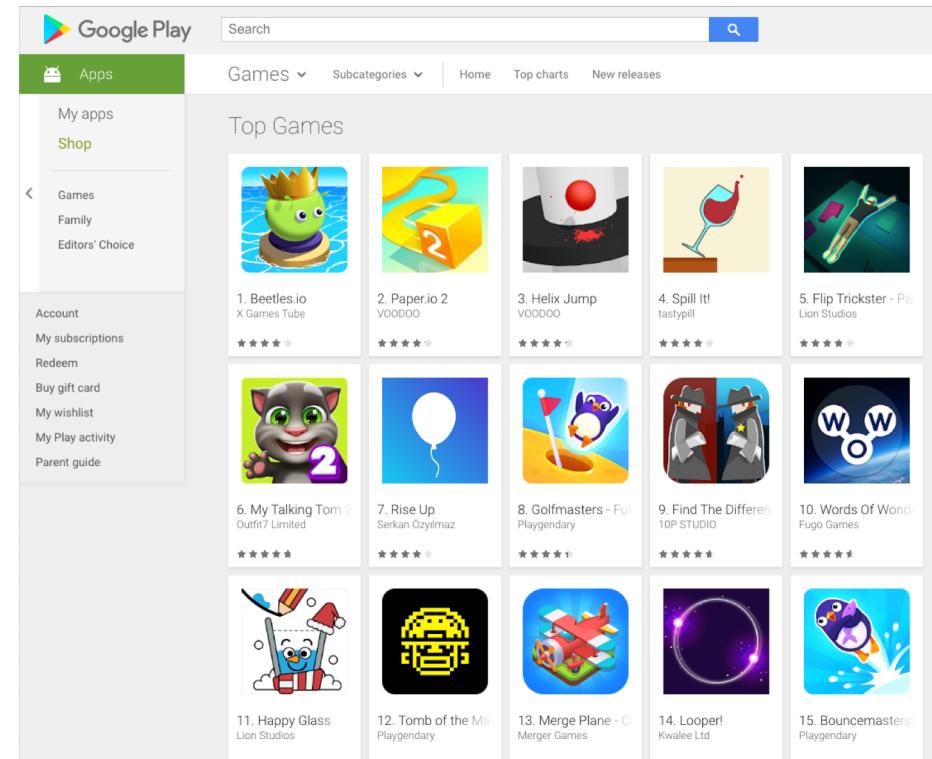
Instruction

- Step 1
 - Access web site such as Google Play and extract at least 1000 app descriptions (No constraint on what kind of app)
- Step 2
 - Preprocess app descriptions: tokenization, normalization, etc.
 - Build and store a vector model using tf-idf to define weights
- Step 3
 - Preprocess and model given query keywords or sentence
 - Write a ranked query processor

Instruction - an example

➤ Step 1

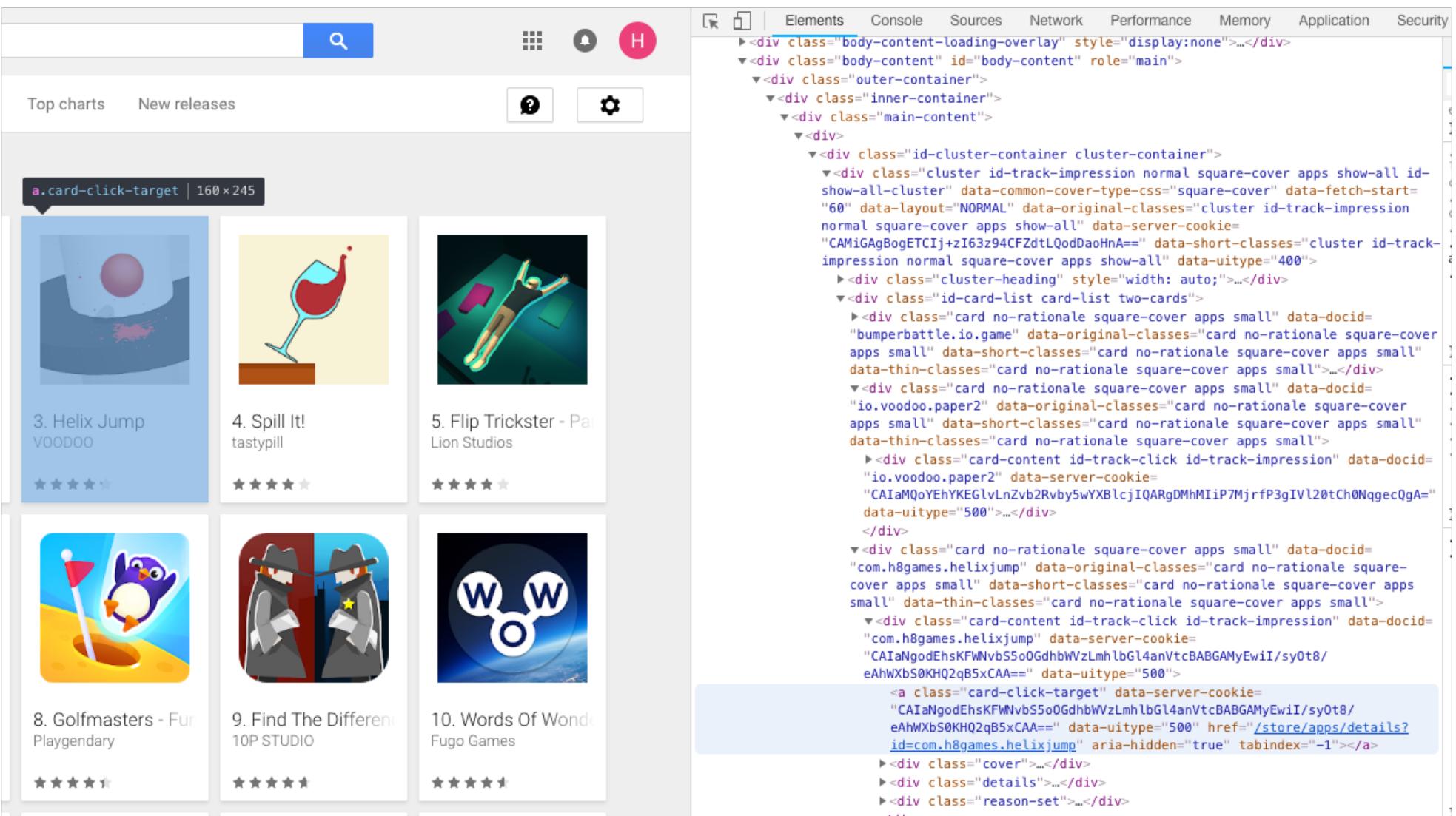
- Access web site such as Google Play and extract at least 1000 app descriptions (No constraint on what kind of app)
- https://play.google.com/store/apps/category/GAME/collection/topselling_free



Instruction - an example

➤ Step 1

- Inspect the webpage with Chrome or save it locally and open it with an editor to check the content



The screenshot shows the Google Play Store interface with several game cards displayed under the 'Top charts' section. The cards include:

- 3. Helix Jump VOODOO
- 4. Spill It! tastypill
- 5. Flip Trickster - Pa Lion Studios
- 8. Golfmasters - Fun Playgendary
- 9. Find The Difference TOP STUDIO
- 10. Words Of Wonder Fugo Games

Below the cards, the developer tools console is open, showing the DOM structure of the page. A specific element, the link for 'Helix Jump' (VOODOO), is highlighted with a pink box and has its class 'a.card-click-target' and dimensions '160x245' indicated. The console output shows the HTML structure for this element, including attributes like 'data-docid' and 'data-uitype'.

```
><div class="body-content-loading-overlay" style="display:none">...</div>
<div class="body-content" id="body-content" role="main">
  <div class="outer-container">
    <div class="inner-container">
      <div class="main-content">
        <div>
          <div class="id-cluster-container cluster-container">
            <div class="cluster id-track-impression normal square-cover apps show-all id-show-all-cluster" data-common-cover-type-css="square-cover" data-fetch-start="60" data-layout="NORMAL" data-original-classes="cluster id-track-impression normal square-cover apps show-all" data-server-cookie="CAMiGAgBogETCIj+zI63z94CFZdtLQodDaoHnA==" data-short-classes="cluster id-track-impression normal square-cover apps show-all" data-uitype="400">
              <div class="cluster-heading" style="width: auto;">...</div>
              <div class="id-card-list card-list two-cards">
                <div class="card no-rationale square-cover apps small" data-docid="bumperbattle.io.game" data-original-classes="card no-rationale square-cover apps small" data-short-classes="card no-rationale square-cover apps small" data-thin-classes="card no-rationale square-cover apps small">...</div>
                <div class="card no-rationale square-cover apps small" data-docid="io.voodoo.paper2" data-original-classes="card no-rationale square-cover apps small" data-short-classes="card no-rationale square-cover apps small" data-thin-classes="card no-rationale square-cover apps small">
                  <div class="card-content id-track-click id-track-impression" data-docid="io.voodoo.paper2" data-server-cookie="CAIAMQoYEHYKEGLvLnZvb2Rvby5WYXBlcjIQARgDMhMIIp7MjrfP3gIVl20tCh0NqgecQgA==" data-uitype="500">...</div>
                </div>
              <div class="card no-rationale square-cover apps small" data-docid="com.h8games.helixjump" data-original-classes="card no-rationale square-cover apps small" data-short-classes="card no-rationale square-cover apps small" data-thin-classes="card no-rationale square-cover apps small">
                <div class="card-content id-track-click id-track-impression" data-docid="com.h8games.helixjump" data-server-cookie="CAIAngodEhsKFNVbS5oGdhbWzLmhbgI4anVtcBABGAMyEwiI/sy0t8/eAhWXb50KHQ2qb5xCAA==" data-uitype="500">
                  <a class="card-click-target" data-server-cookie="CAIAngodEhsKFNVbS5oGdhbWzLmhbgI4anVtcBABGAMyEwiI/sy0t8/eAhWXb50KHQ2qb5xCAA==" data-uitype="500" href="/store/apps/details?id=com.h8games.helixjump" aria-hidden="true" tabindex="-1">...</a>
                <div class="cover">...</div>
                <div class="details">...</div>
                <div class="reason-set">...</div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Instruction - an example

➤ Step 1

- Each app has an unique identifier and a piece of descriptive text

The screenshot shows the Google Play Store interface for the app "Helix Jump". The URL in the address bar is highlighted with a red box and labeled "Identifier". The descriptive text at the bottom of the app's page is also highlighted with a red box and labeled "Descriptive text".

Identifier: com.h8games.helixjump

Descriptive text:

Exciting adventure of the bouncing ball through the helix tower labyrinth.

One-tap easy-to-learn controls, rich visual effects and addictive gameplay mechanics.

Instruction - an example

➤ Step 1

- Get webpage html text by using functions in *urllib* module or other module such as *beautifulsoup*
- Extract app identifier by using regular expression (*re* module)
 - (e.g. using regular expression "href=\"/store/apps/details.*?\"")
 - `re.findall(pattern, string)`
- Access the specific webpage of each app and extract descriptive text by using regular expression
 - Add &hl=en to acquire only descriptions (some apps in Swedish will be returned)
 - Extract description
 - (e.g. using regular expression "itemprop=\"description.*?\">.*?<div jsname=\".*?\">.*?</div>")
- Store the data locally

Instruction - an example

➤ Step 2

- Preprocess the data (*nltk* or *spaCy*)
 - Remove non-alpha-numeric characters
 - Tokenize
 - Lowercase
 - Remove stop words
 - Stemming
- Construct Vector Model
 - Compute tf-idf using *numpy* module or other modules such as *scikit-learn*

Instruction - an example

➤ Step 3

- Preprocess search keywords or sentence
 - Same approach as before
- Compute tf-idf for search keywords or sentence
 - Be clear about how you calculate the weights tf-idf or how the functions from a library calculates the weights
- Compute similarities and return top k results

Notes and Work on jupyter notebook

- Test your code fragments and functions as soon as you achieve
 - Make sure you have correctly obtained at least 1000 apps' urls first
 - Then access the app webpage
- Avoid call crawling function after you have saved app description in local files
- You might need to add parameter after the url
 - *https://play.google.com/store/apps/collection/topselling_free ?start=480&num=60*
- Run
 - *source /home/TDDE16/labs/environment/bin/activate*
 - *source /home/732A92/labs/environment/bin/activate*

Agenda

- Aims of this lab
- Lab task
- Instruction
- Notes
- Work on jupyter notebook

www.liu.se