

DSCI 558: Building Knowledge Graphs

Homework 2: Information Extraction

Released: Jan 31st, 2021

Due: Feb 7th, 2021 @ 23:59

Ground Rules

This homework must be done individually. You can ask others for help with the tools, however, the submitted homework has to be your own work.

Summary

In this homework, you will extract data from unstructured text. You will be using the data you extracted in the previous homework (from IMDb) and spaCy (<https://spacy.io/>), an open-source software library for advanced natural language processing (NLP).

Task 1: Crawl data (1 point)

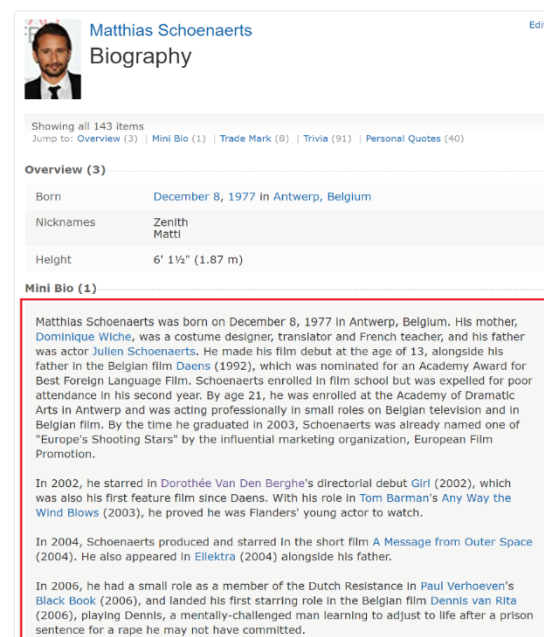
Crawl the biographies of the 500 actors/actresses from IMDb that have **the same birthday (day and month)** with you using the seed url:

https://www.imdb.com/search/name/?birth_monthday=mm-dd

If there are not enough actors/actresses or the actors/actresses you collected don't have enough information, you can use adjacent dates. **List the dates you used in the report.**

Store your results in a comma-separated values (.csv) file with no header. Each row should represent a single entity and should contain two values: URL and biography. Remove all hyperlinks from the collected bios and make sure that you end up with a total of 500 entries in which each entity has a biography text (while scraping/parsing you should ignore entries with no biography text on their page).

Figure 1 shows an example page of a cast entity with its biography marked in a red box.



Matthias Schoenaerts Biography

Showing all 143 items
Jump to: Overview (3) | Mini Bio (1) | Trade Mark (8) | Trivia (91) | Personal Quotes (40)

Overview (3)

Born	December 8, 1977 in Antwerp, Belgium
Nicknames	Zenith Matti
Height	6' 1½" (1.87 m)

Mini Bio (1)

Matthias Schoenaerts was born on December 8, 1977 in Antwerp, Belgium. His mother, Dominique Wiche, was a costume designer, translator and French teacher, and his father was actor Julien Schoenaerts. He made his film debut at the age of 13, alongside his father in the Belgian film *Daens* (1992), which was nominated for an Academy Award for Best Foreign Language Film. Schoenaerts enrolled in film school but was expelled for poor attendance in his second year. By age 21, he was enrolled at the Academy of Dramatic Arts in Antwerp and was acting professionally in small roles on Belgian television and in Belgian film. By the time he graduated in 2003, Schoenaerts was already named one of "Europe's Shooting Stars" by the influential marketing organization, European Film Promotion.

In 2002, he starred in Dorothée Van Den Berghe's directorial debut *Girl* (2002), which was also his first feature film since *Daens*. With his role in Tom Barman's *Any Way the Wind Blows* (2003), he proved he was Flanders' young actor to watch.

In 2004, Schoenaerts produced and starred in the short film *A Message from Outer Space* (2004). He also appeared in *Elektra* (2004) alongside his father.

In 2006, he had a small role as a member of the Dutch Resistance in Paul Verhoeven's *Black Book* (2006), and landed his first starring role in the Belgian film *Dennis van Rita* (2006), playing Dennis, a mentally-challenged man learning to adjust to life after a prison sentence for a rape he may not have committed.

Figure 1. An example actor/actress page with bio marked

Task 2: spaCy (9 points)

In this task you will extract structured data, for each entity, from the unstructured biography text. You will perform **Rule-Based Extraction**. We are interested in the following attributes:

birthplace	Place where the entity was born
education	Educational institution(s) attended by the entity
parents	Name(s) of the parents of the entity
awards	Name(s) of all awards listed in the bio (belonged to the entity or not)
performances	Name(s) of the performance(s) in which the entity participates in
colleagues	Name(s) of people that the entity has worked with

Figure 2 shows these attributes over an example biography text. Attributes are marked by different colors as follows:

- Green: birthplace
- Blue: education
- Red: parents
- Yellow: awards
- Brown: performances
- Black: colleagues

Matthias Schoenaerts was born on December 8, 1977 in Antwerp, Belgium. His mother, Dominique Wiche, was a costume designer, translator and French teacher, and his father was actor Julien Schoenaerts. He made his film debut at the age of 13, alongside his father in the Belgian film Daens (1992), which was nominated for an Academy Award for Best Foreign Language Film. Schoenaerts enrolled in film school but was expelled for poor attendance in his second year. By age 21, he was enrolled at the Academy of Dramatic Arts in Antwerp and was acting professionally in small roles on Belgian television and in Belgian film. By the time he graduated in 2003, Schoenaerts was already named one of "Europe's Shooting Stars" by the influential marketing organization, European Film Promotion.

In 2002, he starred in Dorothee Van Den Berghe's directorial debut Girl (2002), which was also his first feature film since Daens. With his role in Tom Barmann's Any Way the Wind Blows (2003), he proved he was Flanders' young actor to watch.

In 2004, Schoenaerts produced and starred in the short film A Message from Outer Space (2004). He also appeared in Ellektra (2004) alongside his father.

In 2006, he had a small role as a member of the Dutch Resistance in Paul Verhoever's Black Book (2006), and landed his first starring role in the Belgian film Dennis van Rita (2006), playing Dennis, a mentally-challenged man learning to adjust to life after a prison sentence for a rape he may not have committed.

Figure 2: An example biography text with the required attributes marked

Before moving to this section's subtasks, familiarize yourself with spaCy. We provide a python notebook (Task2.ipynb), which contains instructions, code and descriptions on how to perform

information extraction using spaCy and how to implement Rule-Based Matching patterns and functionalities.

Task 2.1 (1 point)

Select one webpage of a single entity that contains all requested attributes (or two if you cannot find a single webpage that contains all attributes) from what you have scraped (out of the 500). Screenshot the page and highlight the required attributes over the screenshot (similar to what is shown in Figure 2).

Task 2.2 (6 points)

Implement two extractors for each attribute:

- **Lexical:** One that uses only textual phrases or regular expressions.
- **Syntactic:** One that uses lexical phrases, the POS tags, dependency parse-tree and named entity type (you can create any type of rules that are available in spaCy's rule-based matcher).

The code should be implemented in python, the script should accept three arguments:

- A input `.csv` file from previous task
- An output `.jl` file, which includes all the extracted values of each attribute and the original url (for each entry). Each value should be a list (except for the url and birthplace). An example file is provided (`sample__task_2_2.jl`).
- A number to indicate the extractor to be used (0 for lexical and 1 for syntactic)

List your extractors in the report.

Notes:

- Your script file (`.py`) should include 12 extractors (6 attributes x 2 types of extractors)
- Each of your two output files (`.jl`) should include 500 entries and **you will need to submit these output files.**
- In the provided notebook, we present a sample code that can be used as a starting point for extracting the spouse of the entity.
- Clarification regarding the difference between **lexical** and **syntactic** extractors:
 - **Lexical** extractors can use only surface-level information or exact text matching
 - An example of such phrase is the text "was born in". So if a sentence has the surface text as "X was born in Y" then this pattern will get activated.
 - In spaCy this corresponds to using the following attributes: `ORTH`, `TEXT`, `LOWER`, `LENGTH`, `IS_ALPHA`, `IS_ASCII`, `IS_DIGIT`, `IS_LOWER`, `IS_UPPER`, `IS_TITLE`, `IS_PUNCT`, `IS_SPACE`, `IS_STOP`, `LIKE_NUM`, `LIKE_URL`, `LIKE_EMAIL`, `SHAPE`
 - **Syntactic** extractors can use the POS and dependency tree.
 - An example of such phrase would involve navigating the dependency tree of the sentence, or matching parts of speech tags (and not just exact string matching).
 - In spaCy this corresponds to using of all of the above attributes and additionally: `POS`, `TAG`, `DEP`, `LEMMA`, `ENT_TYPE`

Task 2.3 (2 points)

In this task you will validate your extractors. Start by building a ground-truth by picking 20 pages out of the 500 pages (any that you want), then label them manually. Store your labeled data in 6 csv files (one for each attribute) with 4 columns:

- url: url of the entity page
- extraction: the labeled value from the bio (if a bio contains multiple values of the same attribute (e.g., 2 parent names), each row should contain only one value as in Figure 3)
- lexical: 0 if your lexical extractor failed to acquire the value you labeled or 1 if it succeeded
- syntactic: similar to lexical

Finally, report the **recall** and **precision** (of each one of your extractors on the labeled data in the report file). More details about precision and recall can be found here:

[https://en.wikipedia.org/wiki/Precision_and_recall#Definition_\(information_retrieval_context\)](https://en.wikipedia.org/wiki/Precision_and_recall#Definition_(information_retrieval_context))

For example, for the parents attribute the file would look as shown in Figure 3. As seen in the figure, the first entity has two values (ground truth, which we manually inserted/labeled). The first extractor failed to extract the second movie but succeeded in the first one. The second extractor succeeded in both. The **recall** of the lexical extractor of this attribute would be 50% and the **recall** for the syntactic one would be 100%.

```
url,extraction,lexical,syntactic
https://www.imdb.com/name/nm0774386,"Julien Schoenaerts",1,1
https://www.imdb.com/name/nm0774386,"Dominique Wilche",0,1
```

Figure 3: An example of an evaluation file

Submission Instructions

You must submit (via Blackboard) the following files/folders in a single **.zip** archive named **Firstname_Lastname_hw02.zip**:

- Firstname_Lastname_hw02_report.pdf: pdf file with your answers to Tasks 1 and 2
- Firstname_Lastname_hw02_bios.csv: as described in Task 1
- Firstname_Lastname_hw02_cast0.jl: output of lexical extractor in Task 2.2
- Firstname_Lastname_hw02_cast1.jl: output of syntactic extractor in Task 2.2
- Firstname_Lastname_hw02_task_2_2.py: as described in Task 2.2
- Csv files containing the evaluation results as described in Task 2.3:
 - Firstname_Lastname_hw02_task_2_3__birthplace.csv
 - Firstname_Lastname_hw02_task_2_3__parents.csv
 - Firstname_Lastname_hw02_task_2_3__education.csv
 - Firstname_Lastname_hw02_task_2_3__performances.csv
 - Firstname_Lastname_hw02_task_2_3__awards.csv
 - Firstname_Lastname_hw02_task_2_3__colleagues.csv
- **source**: This folder includes all the code you wrote to accomplish Tasks 1 and 2 (i.e. your crawler/parser, your extractors code, etc...)