KED2022 Assignment 2: RegEx NLP

Alex Flückiger | University of Lucerne

07 April 2022

Requirements

- Deadline: 15 April 2022, 23:59
- File format: executable shell script
- Naming schema: `SURNAME_KED2022_2.sh`
  Replace `SURNAME` with your surname.
- Use the shell template provided here.
- All tasks require shell commands unless stated otherwise.
- Submit your solutions on time via the respective exercise module on OLAT. The module is only open until midnight.
- Find solutions individually. When you are stuck, post your issue in the OLAT forum and ask friends. In terms of programming, Google may be your best friend.

Motivation

You learn how to extract specific parts of a text and clean a document using regular expressions (RegEx).

Often when you download news articles from a publisher, they do not provide the data in a well-structured format.[1] In some cases, there may be just a single file comprising multiple articles in which the metadata is not properly separated from the actual content. This task is based on a real-world example that makes the use of regular expressions indispensable.

How to start?

Use a text editor to write your script (e.g., `Visual Studio`). You may want to try out the commands directly in your shell and, after successfully running them, copy them over into your script. The command `history` shows the history of all used commands.

Follow this shell template when you write your script.

While coming up with a solution, you may find the following commands useful along the way:

```
# have a look at the document
more newspaper_articles.txt

# useful to check what egrep matches before you write into a file
# colouring output is probably activated by default
egrep --colour "pattern" newspaper_articles.txt
```

[1] Sometimes publishers do this on purpose as they also offer paid services for the curation and analysis of content.

1 Parse Information with RegEx

In this task, you systematically extract the metadata of all the articles contained in the file `newspaper_articles.txt`. You can either download the file or simply update your local git repository using `git pull` and change into the folder `assignments` where you wil find the same file.

In our case, the metadata is written at the beginning of each newspaper article and looks like this:

```
Von BRUNO VANONI, BERN.
591 words
26 February 2004
Tages Anzeiger
TANZ
German
(c) 2004 Tages Anzeiger Homepage Address: https://www.tagesanzeiger.ch/
```

You are going to write three separate commands to extract the following information using regular expressions and to write the resulting output into a new file:

1. number of words (e.g., 591 words; as given in the file, don't count with `wc`)
2. publishing date (e.g., 26 February 2004)
3. names of authors (e.g., Von BRUNO VANONI). Try to write a pattern that doesn't match the location following the person's name (e.g., Bern).

Write the patterns as *specific*, *generalizable*, and *simple* as possible. Moreover, try to cover all cases without getting false positives (i.e., things that shouldn't be matched). Covering all cases may be too hard[2], just push the limit as far as you can. It is more important that you get some practice performing RegEx rather than to come up with a perfect solution.

To be clear, you need a single command for each of the three subtasks, similar to the one below. If this command fails on your system for any reason, get in touch as there may be a technical problem that needs troubleshooting.

```
### EXAMPLE COMMAND
# extract with a pattern and write the matches into a file
egrep -ho "words" newspaper_articles.txt > test.txt
```

Add the commands to your script and write a short comment for each of them.

2 Removing Parts of a Document

In the first task, you have extracted relevant metadata. After parsing these parts, they can be removed from the document to keep only the actual content of the newspaper articles. Thus, you can reuse the same patterns[3] that you have used before.

Write a single line consisting of multiple `sed` commands to remove non-content parts in one go and save the final, cleaned output as a new file. Your command should have a similar structure as below:

```
### EXAMPLE COMMAND
# replace the pattern with an empty sequence to remove it
cat name.txt | sed -E "s/pattern1//g" | sed -E "s/pattern2//g" > clean.txt
```

Feel free to improve your patterns or to write new ones to remove even more parts of the document on the mission to clean the data.

Add the command to your script and write a short comment for it.

[2] An example that may be hard to match: *Von MIT KRISTA SAGER* SPRACH WERNER BOSSHARDT*

[3] Reuse the pattern only, not the entire command with egrep.

3 Test your script

This task is a simple sanity check for your script. Your script has to pass this test, yet you don't need to include it in your submission.

Your deliverable has to be a runnable script comprising all the commands to accomplish the tasks above. To test your script, run the commands below. Once you call the script, it executes all commands, one after another. Everything should be reconstructed accordingly in the test folder. If not, correct the script so that it runs without any issue.

```
mkdir test_script
cd test_script
bash PATH/SCRIPT_NAME.sh    # e.g. bash ../flueckiger_KED2022_2.sh
cd ..
rm -r test_script
```

4 Feedback

Please answer the following questions at the end of your script. Start your answers with the # symbol to make them comments that are ignored when running the script.

1. Do you have any questions concerning the exercise or the commands?
2. How long did it take to solve this exercise? Give a fair estimation.