Language Technology - EDAN20
Lab Session 0 - Fall 2020

Introduction to Python - Numpy - Scikit-learn
Spell Checker

Hicham Mohamad, hi8826mo-s
hsmo@kth.se

September 20, 2020

1 Objectives

The objectives of this lab are to:

- Be sure that you can log on the computer system
- Have the right programming environment
- Have a hands-on introduction to **Python**
- Know of a few Unix tools to derive **word statistics**
- Know of the main functions of **Scikit-learn**

The student's presence in the computer room through Zoom is not compulsory for this initial session. Its goal is to be sure that all the students have the elementary Python programming skills they need for the course. This means that if you know Python and scikit-learn (well enough), you can skip the lab. However, each student will have to run a short program at home on **spelling correction** and comment it. This last exercise is compulsory as well as handing in the report. See the last section of the page.

2 Organization and location

The initial lab sessions will take place on Septembe1 7 and 2

There can be last minute changes. Please check the official times here: `https://cloud.timeedit.net/lu/web/lth1/ri1Q5006.html`

In this lab, we will review the **Python syntax** and some tools. Attendance is not compulsory, but you have to run the **spell checker** by Peter Norvig at home to make sure you understand Python.

3 Outline

1. We will use Python 3 and the **Anaconda** distribution in the labs: `https://www.anaconda.com/distribution`. Anaconda has most packages, including numpy, we need for the course.
2. Anaconda is available on the lab machines. You add it to your path by running: `$ initcs` If you use a personal machine, you will have to download and install it.

3. **regex** is one of the few modules that is not part of Anaconda. You can find it on pypi: `https://pypi.python.org/pypi/regex/`. `regex` can handle regular expressions and Unicode. It should already be installed on the LTH computer network. On your personal machine, install them with `pip`:

   ```
   python -m pip install --upgrade regex
   ```

4. You will carry out the labs with **Jupyter notebooks**, where you will write code snippets (cells) that you can run interactively. You start jupyter with:

   ```
   $ jupyter lab
   ```

   or

   ```
   $ jupyter notebook
   ```

5. You may want also to use an interactive programming environment (IDE). We recommend **PyCharm**: `https://www.jetbrains.com/pycharm/`. The community edition is free. PyCharm should be available on the lab computers. If not, you will add the Python plugin to **IntelliJ** instead. Run:

   ```
   $ intellij-idea-community
   ```

   then Configure and add Python

6. On the LTH machines, the `regex` module is not available from PyCharm by default. You need first to configure your environment. To do so, in the File menu, select Settings..., then Project and Project Interpreter. In the Project Interpreter box, on the top of the right pane, add the new interpreter by pressing the cog icon, and Add... Then select Anaconda Python:

   ```
   /usr/local/anaconda3/bin/python
   ```

4 Course of the lab

In the lab session. your instructors will walk you through Python, Unix, and scikit-learn. You will:

1. Run all the code in the chapter: **A Tour of Python** is available here

   `https://github.com/pnugues/ilppp/tree/master/programs/appB/python`

   You may create a notebook for it or run it with PyCharm;

2. count the words of a text with Unix tools

3. run the quick introduction to scikit-learn:

   `https://scikit-learn.org/stable/tutorial/basic/tutorial.html`

   and understand `fit()` and `predict()`.

4. Optionally, you can also run the notebooks on **Pandas** and **Matplotlib**.

If you have questions on these topics, Python, Unix, numpy, and linear algebra, ask your instructors. They will help you.

5 Compulsory part

1. Run the spell checker here: `http://norvig.com/spell-correct.html`. Use Python 3 and make sure you understand all the code and Python syntax.

2. Hand in an **individual comment** of this program on one to two pages (not more). To write this **report**, please use the overleaf site (`www.overleaf.com`) and Latex. Once you are done, please send the overleaf link to it (not the PDF) to the EDAN20 course address. Please check your document with a spell checker before you send it.

6 Appendix

References

[1] Pierre Nugues, *Language processing with Perl and Prolog*, 2nd edition, 2014, Springer.

[2] P. Nugues, Language Technology - EDAN20, Lectures notes: `https://cs.lth.se/edan20/`

[3] VanderPlas, A Whirlwind Tour of Python. O'Reilly 2016. Online reading: `https://jakevdp.github.io/WhirlwindTourOfPython/`

[4] Tf-Idf measure: `https://en.wikipedia.org/wiki/Tf-idf`.

[5] Geron, Jupyter notebook on linear algebra from Machine Learning and Deep Learning in Python using Scikit-Learn, Keras and TensorFlow: `https://github.com/ageron/handson-ml2`

[6] Peter Norvig, *How to Write a Spelling Corrector*, `http://norvig.com/spell-correct.html`