Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

416591334

CRC 1412
REGISTER

Jupyter Notebooks for Linguists

Introduction to NLP FLAIR with Jupyter

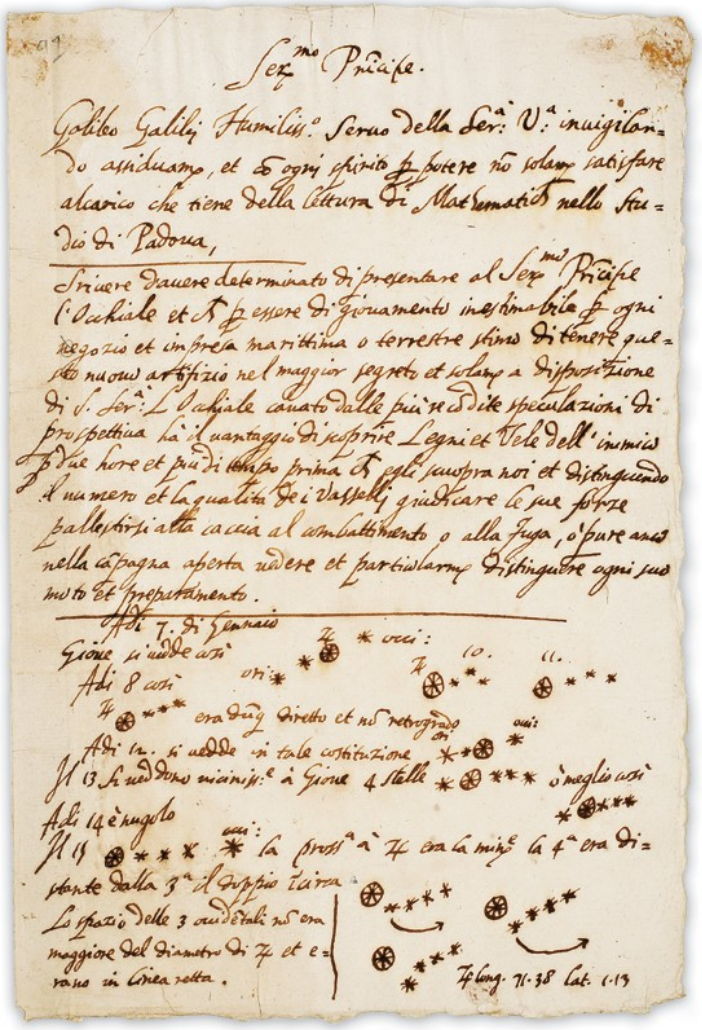
André Renis

07.06.2022 – 10.06.2022

Jupyter notebooks

CRC 1412
REGISTER

- Invented in 2014 by Fernando Pérez and Brian Granger starting with Julia, Python and R. - Now supporting 137 different kernels (runtime environments, programming languages)
- Inspired by Galileo's famous notebooks recording the discovery of the moons of Jupiter
- Describing text passages and scientific calculations/sketches are mixed in one document allowing a high transparency/documentation of the scientific approach
- This idea of a mixed scientific document is transferred into cloud computing with the possibility of interactive parallel work on the same resources within a browser
- Jupyter notebooks are running on a server: Starting programming works out of the box. There is no installation needed on your device.
- Servers are sometimes providing very fast GPU/TPU calculations


Sez.mo Prencipe.

Galileo Galilej Humiliss.o Servo della Ser.a V.a inuigilan=
do assiduam.te, et di ogni spirito di potere no solam.te satisfare
alcarico che tiene della Lettura di Mathematica nello stu=
dio di Padoua,

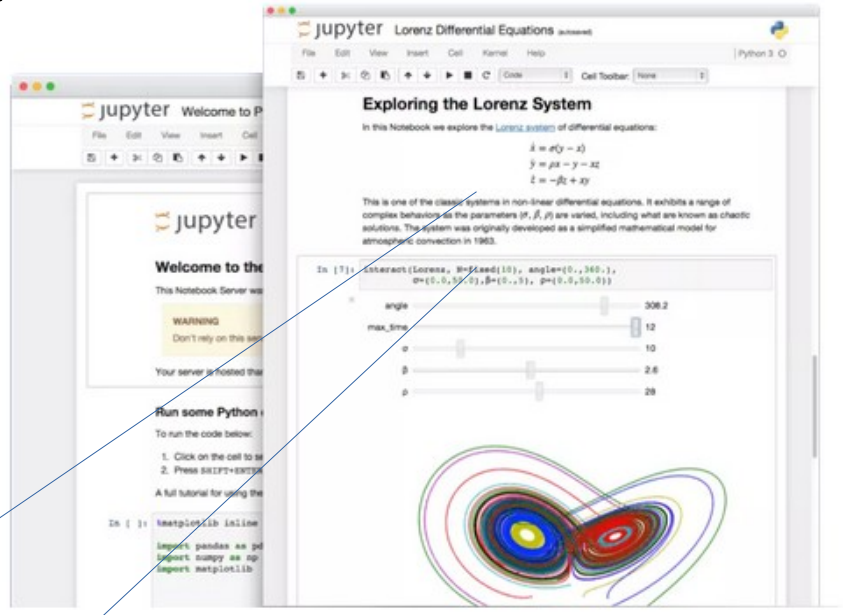
Scriuere dauere determinato di presentare al Ser.mo Prencipe
l'Occhiale et di essere di giouamento inestimabile in ogni
negozio et impresa marittima o terrestre stimo di tenere que=
sto nuouo artificio nel maggior segreto et solam.te a dispositione
di V. Ser.a: L'Occhiale cauato dalle più recondite speculationi di
prospettiua ha l'auantaggio di scoprire Legni et Vele dell'inimico
più hore et più di tempo prima di esser egli scoperto noi et distinguendo
il numero et la qualità de i Vasselli giudicare le sue forze
esser terribili alla caccia al combattimento o alla fuga, o pure anco
nella campagna aperta uedere et particolarm.te distinguere ogni suo
moto et preparamento.

Adi 7. di Genaro
Giove si uedde così
Adi 8 detto
era drio diretto et no retrogrado
Adi 12. si uedde in tale costitutione
Il 13 si uedono mirabilm.te 4 Stelle disposte così
Adi 14 è nuuolo
Il 15 la prossa a ♄ era la mig.a la 4.a era di=
stante dalla 3.a il doppio di tutta
Lo spatio delle 3 cui dietro no era
maggiore del diametro di ♄ et e=
rano in linea retta.
♄ long. 11.38 Lat. 1.13

Jupyter notebooks – introduction

CRC 1412 REGISTER

- Can be installed on your own server (or local device). But most commonly used in a maintained cloud environment, like:
  - Google Colaboratory (we will use this service)
  - Kaggle.com
  - Upcoming Jupyter-Hub at hu-berlin.de
- Jupyter-Notebook-Documents (*.ipynb) are stored
- in JSON-format on the server
  - → Every file has to be uploaded
- Two different types of sections:
  - A.) Formatted text sections: Markdown or LaTeX
  - B.) Interactive code sections: Python (and many others)
- Code is executed either entirely or section by section (input sections and output sections; values of variables are stored!)



Jupyter notebooks – introduction

CRC 1412 REGISTER

- Jupyter notebooks are offering a new way of social coding (entirely open-source)
- Scientists can work together with same computing environment, testing code together, data input and results
- 3rd party modules have to be installed with pip (in a code section) like in a local installation. Example:
  - ***!pip install flair***
  - → Everything needed to run a notebook is by definition part of the document (no hidden secrets behind)
    - Whole working environment can be stored (as documentation) in a docker-image (virtual machine):
  - → Results can be reproduced anytime/anywhere with same libs/configuration/code etc.
- → ***Let's start with a first notebook document and install flair on google colab!***

Jupyter notebooks – hands on!

CRC 1412 REGISTER

- You can either work on your own or start a new notebook with someone else
  - → You can share documents in real time with Google Colab
- We will work in the same document the whole course: Let's start with a text section in markdown and produce some formatted text, like a headline and description of our project
- We want to explore Jupyter notebooks with Google Colab by:
  - Uploading a text-file (very small corpus) to Google Colab
  - Reading the content of the file with Python and store it to variable

What is FLAIR?

- A very simple framework for state-of-the-art NLP. Developed by Humboldt University of Berlin and Zalando.
- ***Flair supports:***
- Named entity recognition (NER)
- Part-of-speech tagging (PoS)
- Sense disambiguation and classification
- A growing number of languages
- Text/word embeddings
- Sequence labeling, text classification, similarity learning and text regression
- And much more!

Working with FLAIR

- **What we want today:**

- Read our text file with the FLAIR library
- Perform a POS-tagging on our text
- Perform a NER-tagging on our text
- Optionally perform a semantic frame detection (or search for offensive language)
- Save the results to an excel-sheet (you will need to construct nested cells if you have spans!)
- → You have to work with python documentation, flair documentation and openpyxl documentation

Levels of tagging in flair

- Tagging a whole text
- Tagging a whole sentence
- Tagging spans (ex. NER tagging) “George Washington”
- Tagging tokens (ex. POS tagging)

Solution of the exercise

```
from flair.models import MultiTagger
from flair.data import Sentence
from openpyxl import Workbook


file = open("merkel_interview.txt", "r")
content = file.read()

wb = Workbook()
ws = wb.create_sheet("merkel_interview", 0)

# Header in row 1 columns A,B,C,D,E
ws.append(['TEXT','POS','SCORE','NER','SCORE'])

tagger = MultiTagger.load(['de-ner', 'de-pos'])
sentence = Sentence(content)
tagger.predict(sentence)

for token in sentence:
    label = token.get_label("de-pos")
    txt = token.text
    pos = label.value
    score = label.score
    # POS tagging values beginning at row #2 columns A,B,C
    ws.append([txt,pos,score]);
```

Solution of the exercise 2

CRC 1412 REGISTER

flair

```
# values for ner-tagging are going in column D and E
# Still looking for a better solution than the one below. I'am sorry. This is not beautiful ;-)

for span in sentence.get_spans('de-ner'):

    txt = str(span) # String looks like this: Span[17:19]: "Wiebke Hollersen" → PER (0.9998)
    txt = txt[txt.index('[')+1:txt.index(']')] # we cut everything out within the square brackets
    txt = txt.split(":") # and we get the numbers of the index
    start = int(txt[0])+2 # sheet starts at row #2 because of header. Index of flair starts at 0
    end = int(txt[1])+1 # end position only +1

    ws['D' + str(start)] = span.tag # put values in fields for NER-data
    ws['E' + str(start)] = span.score

    if((end - start) > 0): # NER tagging is a span > token (only one row)
        # we have to merge the cells
        ws.merge_cells('D' + str(start) + ':D' + str(end))
        ws.merge_cells('E' + str(start) + ':E' + str(end))

wb.save('merkel_interview.xlsx')
```

HUMBOLDT-UNIVERSITÄT ZU BERLIN

zalando research

Solution of the exercise 3

Resulting Excel-Sheet with merged cells looks like this:

TEXT	POS	SCORE	NER	SCORE
Angela	NE	0,999997		
Merkel	NE	0,999994	PER	0,999817
:	$.	0,999997		
Sie	PPER	0,999997		
sprach	VVFIN	0,999999		
über	APPR	0,999999		
Putin	NE	0,891732	PER	0,999612
,	$,	1		
den	ART	0,99999		
Krieg	NN	0,999994		
,	$,	1		
aber	KON	0,999842		
nicht	PTKNEG	0,999999		
über	APPR	0,999969		
den	ART	1		
3.	ADJA	1		
Oktober	NN	0,999994		
Wiebke	NE	0,999938		
Hollersen	NE	0,9999	PER	0,999785

Usefull resources

Jupyter

https://jupyter.org/

https://www.datacamp.com/tutorial/tutorial-jupyter-notebook

Flair

https://github.com/flairNLP/flair

We made lessons 1 + 2 of this tutorial
https://github.com/flairNLP/flair/blob/master/resources/docs/TUTORIAL_1_BASICS.md

https://www.informatik.hu-berlin.de/en/forschung-en/gebiete/ml-en/Flair

NLP

https://thomaskrause.github.io/nlp-mit-python/
https://web.stanford.edu/~jurafsky/slp3/