Lecture 9: Data Formats, Data-mining Web and Social Media

LING 1340/2340: Data Science for Linguists

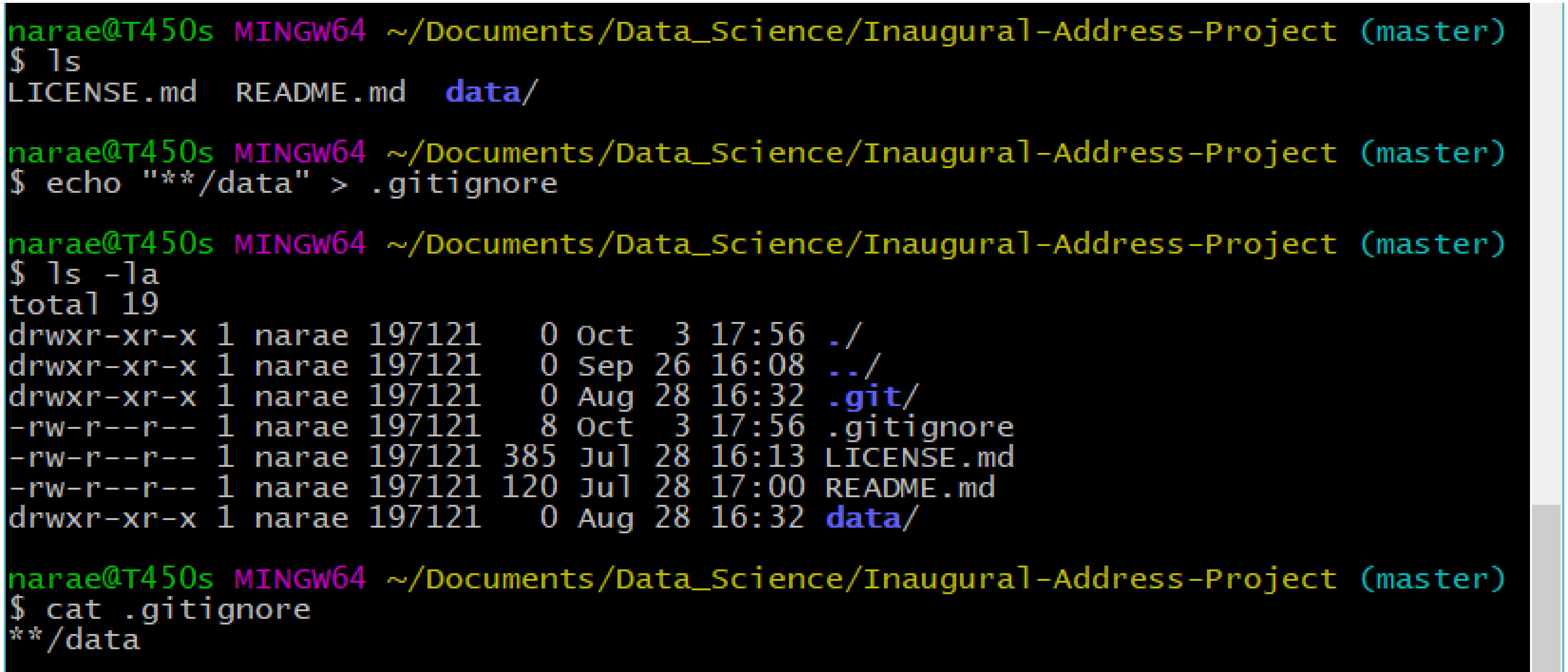
Na-Rae Han

Objectives

- Corpus linguistics
  - Gries & Newman (2013) "Creating and using corpora"
- Your term projects
  - Data management: copyright and licensing
- Data formats
- Data-mining web & social media
  - Twitter mining: To-do #8 review
  - Web mining

Your Project Repo

- As a principle, your term project -- including code and data -- should be **as public and open as possible**.
  - Set your repo to **public** at this time.
    - Justification needed for changing to private.
  - For now, store your data files in data/ directory, and have git ignore this directory through `.gitignore` file, like below:



```
narae@T450s MINGW64 ~/Documents/Data_Science/Inaugural-Address-Project (master)
$ ls
LICENSE.md  README.md  data/

narae@T450s MINGW64 ~/Documents/Data_Science/Inaugural-Address-Project (master)
$ echo "**/data" > .gitignore

narae@T450s MINGW64 ~/Documents/Data_Science/Inaugural-Address-Project (master)
$ ls -la
total 19
drwxr-xr-x 1 narae 197121   0 Oct  3 17:56 ./
drwxr-xr-x 1 narae 197121   0 Sep 26 16:08 ../
drwxr-xr-x 1 narae 197121   0 Aug 28 16:32 .git/
-rw-r--r-- 1 narae 197121   8 Oct  3 17:56 .gitignore
-rw-r--r-- 1 narae 197121 385 Jul 28 16:13 LICENSE.md
-rw-r--r-- 1 narae 197121 120 Jul 28 17:00 README.md
drwxr-xr-x 1 narae 197121   0 Aug 28 16:32 data/

narae@T450s MINGW64 ~/Documents/Data_Science/Inaugural-Address-Project (master)
$ cat .gitignore
**/data
```

Data standards & exchange formats

	What	Notes, reference
CSV	Comma-separated values	Compatible with Excel
TSV	Tab-separated values	
HTML	Web pages	
XML	For markup and text encoding	A Gentle Introduction to XML by TEI
JSON	JavaScript Object Notation Twitter, Jupyter Notebook	Introducing JSON JSON example (vs. XML)

They are all TEXT files.

- Encoding: Latin-1, ASCII, UTF-8, UTF-16, CP1252, ...
- Line endings:
  - LF (`'\n'`: OS X & Linux) , CRLF (`'\r\n'`: Windows)
- But underneath it all, these files are all TEXT files with **special formatting syntax** and **special characters** designated for formatting purposes.
  - In command line, you can `cat` and `less` through the files.
  - You can open them up in a text editor (Atom, Notepad++) and edit.
  - Some editors/applications are aware of the format-specific syntax and will highlight/render accordingly.
    - Unlike, say, PDF files, style attributes are NOT part of the files themselves. (e.g., markdown file)

Do not re-invent the wheel.

- Don't try and parse them manually.
- There are Python libraries. Import and use them.
  - CSV & TSV: `pandas`
  - HTML & XML: Beautiful Soup (`bs4`)
  - JSON:
    - `json` library
    - `pandas.read_json`

Resource-specific (ad-hoc) formats

- Brown corpus

```
The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd
Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent/jj
primary/nn election/nn produced/vbd ``/`` no/at evidence/nn ''/''
that/cs any/dti irregularities/nns took/vbd place/nn ./.
```

- Korean Treebank corpus:

```
;;05:127: 저는 그 일을 할 수 있는 한 빨리 하겠습니다 .

(S  (NP-SBJ 저/NPN+는/PAU)
    (VP (NP-OBJ-LV 그/DAN
                   일/NNC+을/PCA)
        (VP (NP-ADV (S (NP-SBJ (S (NP-SBJ *pro*)
                                  (VP 하/VV+ㄹ/EAN))
                              (NP 수/NNX))
                        (ADJP 있/VJ+는/EAN))
                    (NP 한/NNX))
            (ADVP 빨리/ADV)
            (VP (LV 하/VV+겠/EPF+습니다/EFN))))
    ./SFN)
```

It is up to end users to write code to parse data files.

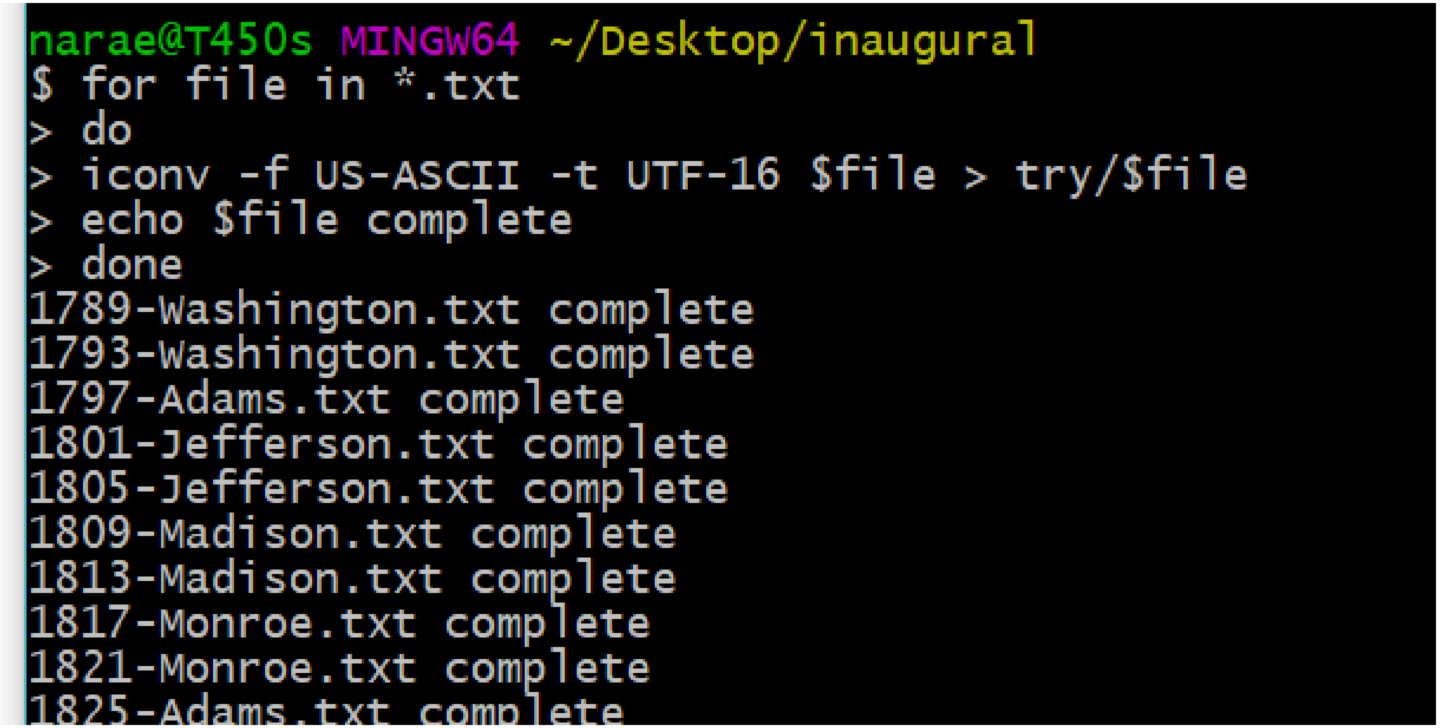
Refer to documentation!

Format conversion

- When dealing with corpora, you may need to convert 100+ files at once.
  - On-line services are too cumbersome.
  - Try batch-processing through command line.
- Automatic tools available on command line.
  - Encoding conversion: `iconv` (Linux, OS X, on Git Bash)
  - Line ending conversion: `unix2dos`, `dos2unix`
  - Pandoc http://www.pandoc.org/
    - Universal document coverter
    - HTML, XML, PDF, LaTeX, Markdown, Epub, MS Doc, ...

Batch processing through shell scripting

- Your command line is actually running a programming environment: bash shell.
- You can *program* in command line, even for loops!



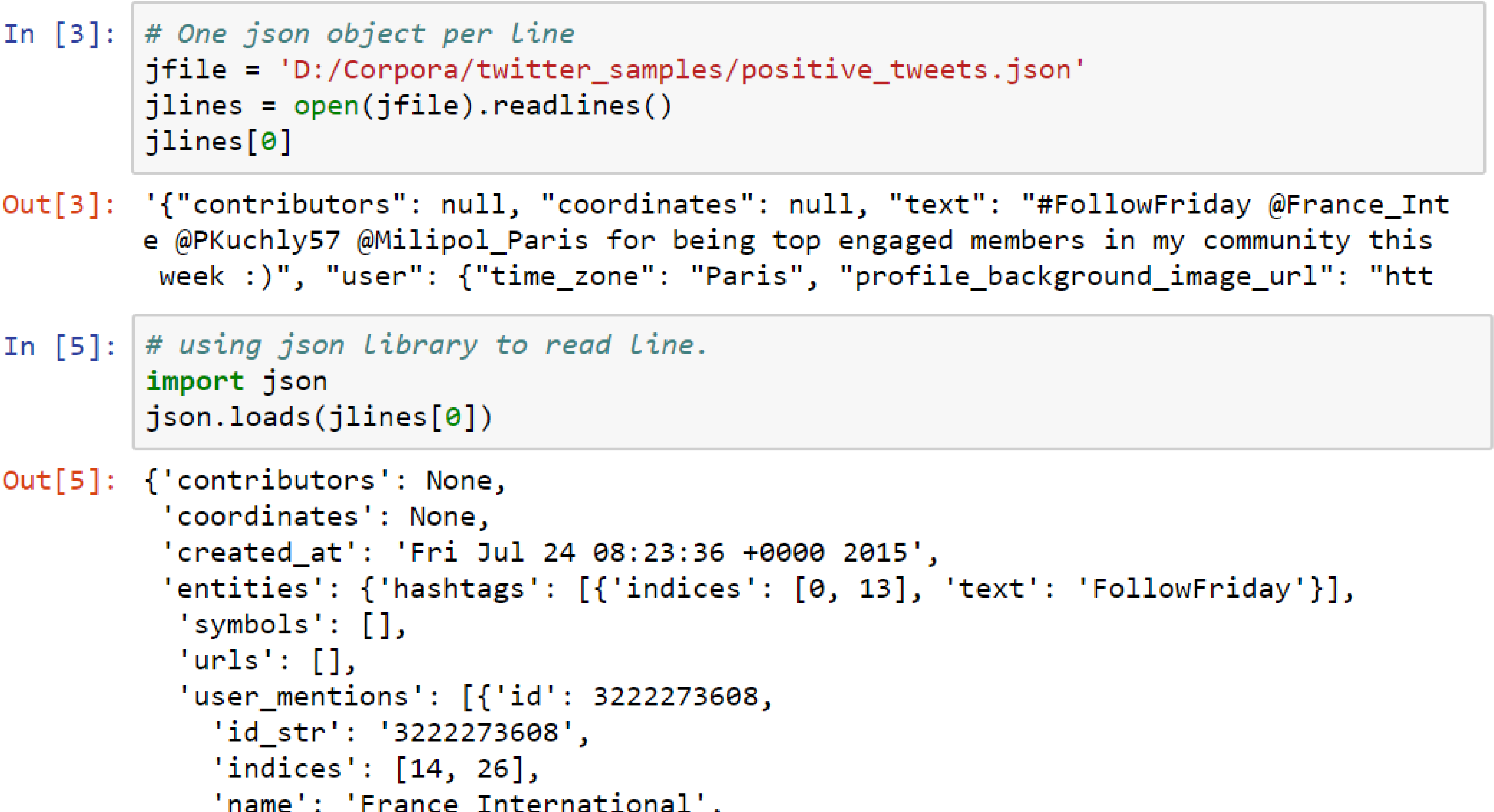
```
narae@T450s MINGW64 ~/Desktop/inaugural
$ for file in *.txt
> do
> iconv -f US-ASCII -t UTF-16 $file > try/$file
> echo $file complete
> done
1789-Washington.txt complete
1793-Washington.txt complete
1797-Adams.txt complete
1801-Jefferson.txt complete
1805-Jefferson.txt complete
1809-Madison.txt complete
1813-Madison.txt complete
1817-Monroe.txt complete
1821-Monroe.txt complete
1825-Adams.txt complete
```

Twitter mining

- Twitter sample corpus
  - Static corpus: download from the NLTK data page
- How does one data-mine Twitter?
  - Answer: through **API** (Application Program Interface)
  - To-do #8
  - Getting acquainted with JSON format
  - Data Analysis using Twitter API and Python, The Code Way tutorial
  - And a couple more on the Learning Resource page
- Libraries used: `tweepy`, `json`
- How did you like Twitter Mining?

Processing a static Twitter corpus

- "Twitter Samples" corpus can be downloaded from http://www.nltk.org/nltk_data/



```
In [3]:  # One json object per line
         jfile = 'D:/Corpora/twitter_samples/positive_tweets.json'
         jlines = open(jfile).readlines()
         jlines[0]

Out[3]: '{"contributors": null, "coordinates": null, "text": "#FollowFriday @France_Int
        e @PKuchly57 @Milipol_Paris for being top engaged members in my community this
         week :)", "user": {"time_zone": "Paris", "profile_background_image_url": "htt

In [5]:  # using json library to read line.
         import json
         json.loads(jlines[0])

Out[5]: {'contributors': None,
         'coordinates': None,
         'created_at': 'Fri Jul 24 08:23:36 +0000 2015',
         'entities': {'hashtags': [{'indices': [0, 13], 'text': 'FollowFriday'}],
          'symbols': [],
          'urls': [],
          'user_mentions': [{'id': 3222273608,
            'id_str': '3222273608',
            'indices': [14, 26],
            'name': 'France International',
```

Web mining

- Involves "web crawling" "web spyder", ...
- `scrapy` is the most popular library.
  - https://scrapy.org/
  - You will have to install it first. How to use "pip"?
- Scrapy tutorial:
  - Official Scrapy:
    - https://doc.scrapy.org/en/latest/intro/tutorial.html
  - Digital Ocean:
    - https://www.digitalocean.com/community/tutorials/how-to-crawl-a-web-page-with-scrapy-and-python-3

Wrapping up

- No class next Tuesday.
- Project 1st report due on Thursday.
  - Focus on data curation.
  - Details will be posted on the Project Guidelines page.
- Wednesday office hours: 2:30 -- 4
  - I'm out of town until Wednesday morning. Email me.
- Get started with Machine Learning content:
  - http://www.pitt.edu/~naraehan/ling1340/resources.html#mining