Lecture 9: Data Formats, Text File Encoding & Conversion, Web Mining

LING 1340/2340: Data Science for Linguists

Na-Rae Han

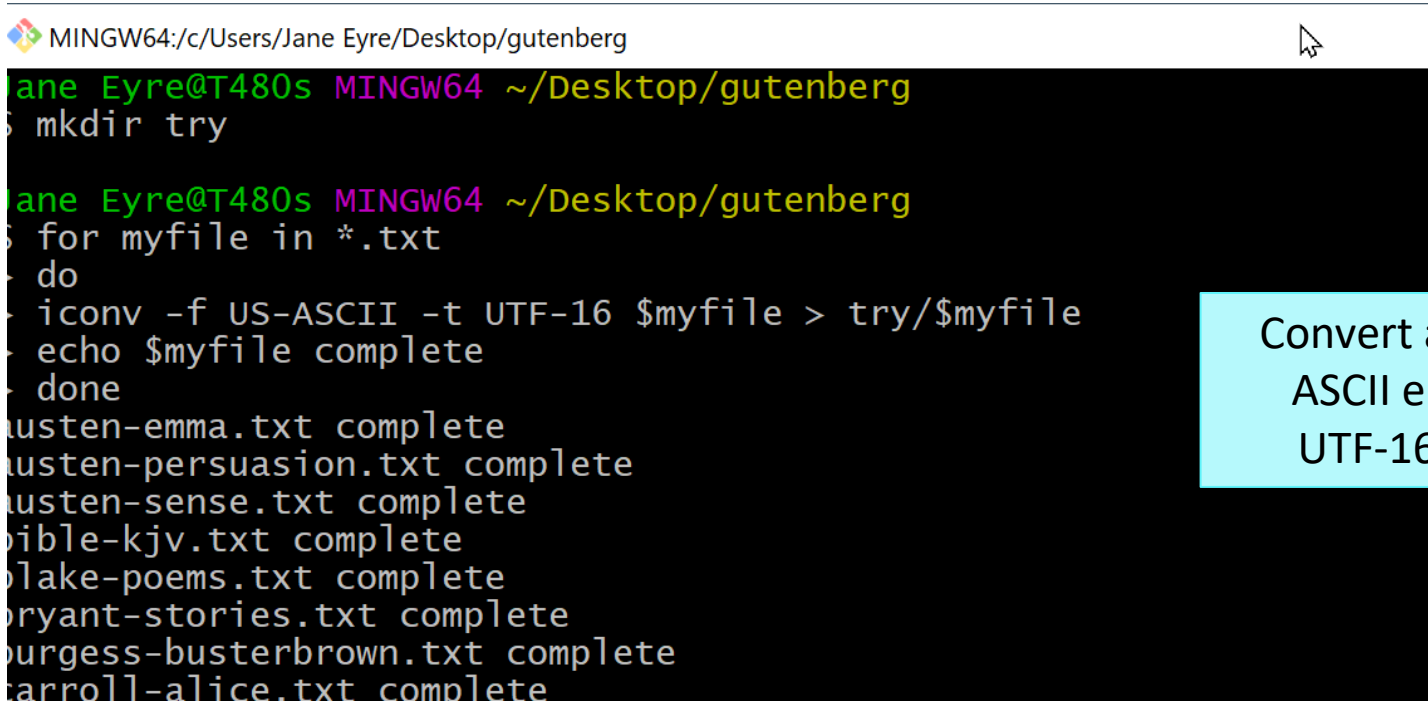
Objectives

- Corpus data: standard and popular formats
  - File formats: conversion
  - Review of common data formats
- Web and social media mining
  - Web pages: HTML basics
  - Twitter mining revisited

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!

Or: Z shell (zsh, on Macs)



Convert all files from ASCII encoding to UTF-16 encoding

Keeping an eye out for error messages!

```
Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg
$ for myfile in *txt
> do
> iconv -f US-ASCII -t UTF-16 $myfile > try/$myfile
> echo $myfile finished
> done
austen-emma.txt finished
austen-persuasion.txt finished
austen-sense.txt finished
bible-kjv.txt finished
blake-poems.txt finished
bryant-stories.txt finished
burgess-busterbrown.txt finished
carroll-alice.txt finished

iconv: chesterton-ball.txt:4631:7: cannot convert
chesterton-ball.txt finished
chesterton-brown.txt finished
chesterton-thursday.txt finished
edgeworth-parents.txt finished
melville-moby_dick.txt finished
milton-paradise.txt finished

iconv: shakespeare-caesar.txt:119:9: cannot convert
shakespeare-caesar.txt finished
shakespeare-hamlet.txt finished
shakespeare-macbeth.txt finished
whitman-leaves.txt finished

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg
$
```

Checking conversion output

```
Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg
$ cd try

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ ls -lh
total 22M
-rw-r--r-- 1 Jane Eyre 197121 1.7M Feb 14 12:29 austen-emma.txt
-rw-r--r-- 1 Jane Eyre 197121 911K Feb 14 12:29 austen-persuasion.txt
-rw-r--r-- 1 Jane Eyre 197121 1.3M Feb 14 12:29 austen-sense.txt
-rw-r--r-- 1 Jane Eyre 197121 8.3M Feb 14 12:29 bible-kjv.txt
-rw-r--r-- 1 Jane Eyre 197121  75K Feb 14 12:29 blake-poems.txt
-rw-r--r-- 1 Jane Eyre 197121 488K Feb 14 12:29 bryant-stories.txt
-rw-r--r-- 1 Jane Eyre 197121 166K Feb 14 12:29 burgess-busterbrown.txt
-rw-r--r-- 1 Jane Eyre 197121 283K Feb 14 12:29 carroll-alice.txt
-rw-r--r-- 1 Jane Eyre 197121 445K Feb 14 12:29 chesterton-ball.txt
-rw-r--r-- 1 Jane Eyre 197121 795K Feb 14 12:29 chesterton-brown.txt
-rw-r--r-- 1 Jane Eyre 197121 627K Feb 14 12:29 chesterton-thursday.txt
-rw-r--r-- 1 Jane Eyre 197121 1.8M Feb 14 12:29 edgeworth-parents.txt
-rw-r--r-- 1 Jane Eyre 197121 2.4M Feb 14 12:29 melville-moby_dick.txt
-rw-r--r-- 1 Jane Eyre 197121 915K Feb 14 12:29 milton-paradise.txt
-rw-r--r-- 1 Jane Eyre 197121 7.6K Feb 14 12:29 shakespeare-caesar.txt
-rw-r--r-- 1 Jane Eyre 197121 319K Feb 14 12:29 shakespeare-hamlet.txt
-rw-r--r-- 1 Jane Eyre 197121 196K Feb 14 12:29 shakespeare-macbeth.txt
-rw-r--r-- 1 Jane Eyre 197121 1.4M Feb 14 12:29 whitman-leaves.txt

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ ls -lh ../bible-kjv.txt
-rw-r--r-- 1 Jane Eyre 197121 4.2M Feb 12 11:34 ../bible-kjv.txt

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ ls -lh ../chesterton-ball.txt
-rw-r--r-- 1 Jane Eyre 197121 447K Feb 12 11:34 ../chesterton-ball.txt

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ ls -lh ../shakespeare-caesar.txt
-rw-r--r-- 1 Jane Eyre 197121 110K Feb 12 11:34 ../shakespeare-caesar.txt
```

Converted files.
All files are there, but...

Converted Bible file is now 8.3MB, which is double the size of the original. Good!

Not so for the two files that prompted an error.
Original files are in fact larger!

Always VALIDATE your conversion output!

```
Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ wc -l shakespeare-caesar.txt
118 shakespeare-caesar.txt

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ wc -l ../shakespeare-caesar.txt
3523 ../shakespeare-caesar.txt

Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ tail shakespeare-caesar.txt
   Cask. Peace ho, Caesar speakes

   Caes. Calphurnia

   Calp. Heere my Lord

   Caes. Stand you directly in Antonio's way,
When he doth run his course. Antonio

   Ant. C
Jane Eyre@X13-Yoga MINGW64 ~/Desktop/gutenberg/try
$ tail ../shakespeare-caesar.txt
Most like a Souldier ordered Honourably:
So call the Field to rest, and let's away,
To part the glories of this happy day.

Exeunt. omnes.


FINIS. THE TRAGEDIE OF IVLIVS CaeSAR.
```

Digging further. Original "Caesar" had 3523 lines, but the converted one has only 118...

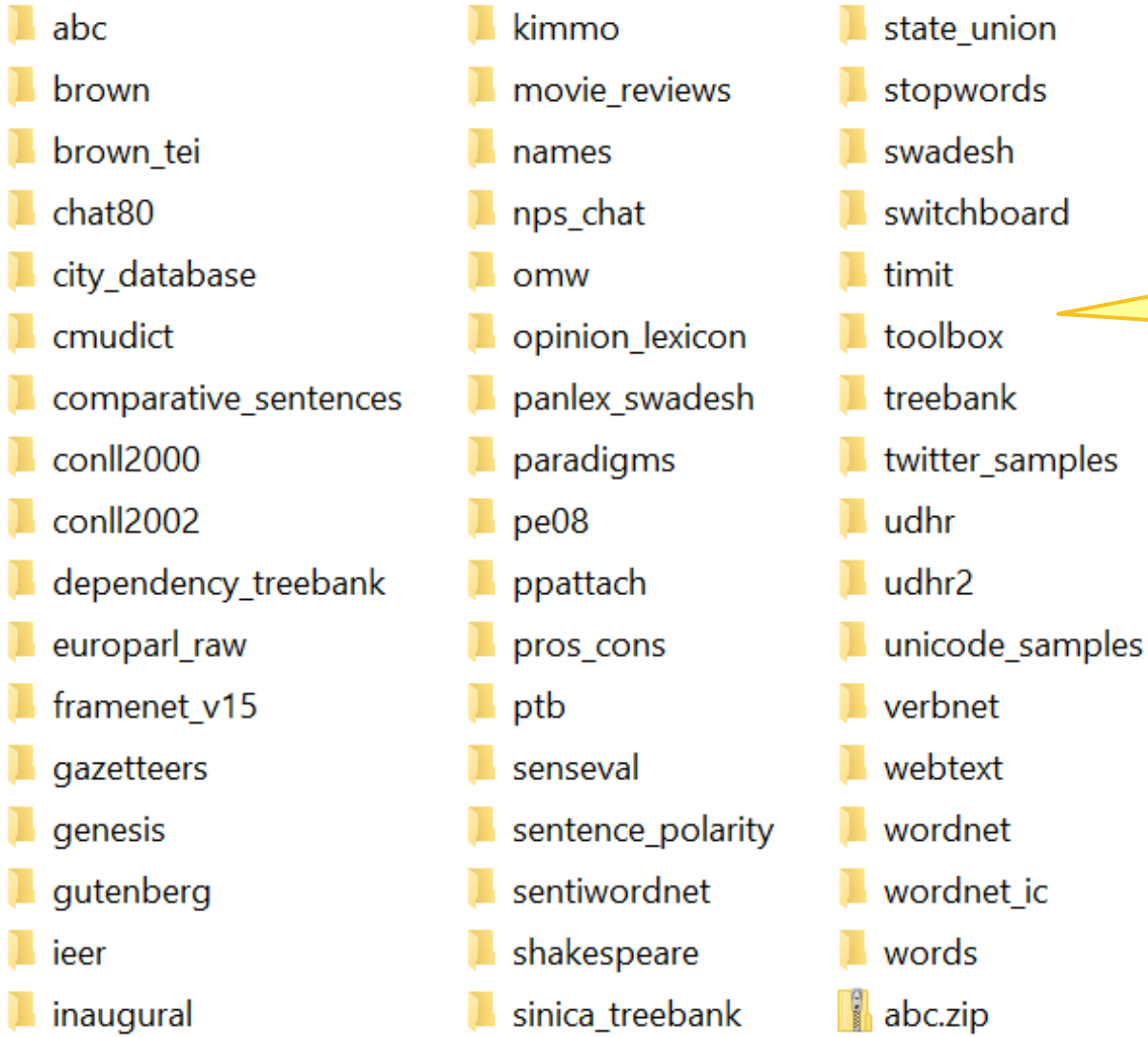
Turns out, file-write operation cut off when iconv encountered the encoding error!!!

Always check and validate output of your data and file transformation tasks!

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.
  - Finding out text file encoding, line ending: `file` command (also `file -i`)
  - Encoding conversion: `iconv` (Linux, OS X, on Git Bash)
  - Line ending conversion: `unix2dos`, `dos2unix`
  - Pandoc https://www.pandoc.org/
    - Universal document converter
    - HTML, XML, PDF, LaTeX, Markdown, Epub, MS Doc, ...
    - After installation, you can use it via command line

A brief tour of NLTK's many "corpus" data



Many of them are language data, not corpora per se

Diverse genres and data formats represented!

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)
```

NOT standard
(cf. XML, JSON).
Project-dependent.

It is up to end users to understand the data format, then write code to parse data files.

Refer to documentation!

Also: Python libraries may already exist

Dependency annotation: format

- https://raw.githubusercontent.com/UniversalDependencies/UD_English-EWT/dev/en_ewt-ud-dev.conllu

```
# sent_id = weblog-blogspot.com_nominations_20041117172713_ENG_20041117_172713-0002
# newpar id = weblog-blogspot.com_nominations_20041117172713_ENG_20041117_172713-p0002
# text = President Bush on Tuesday nominated two individuals to replace retiring jurists on federal courts in the Washington area.
1	President	President	PROPN	NNP	Number=Sing	5	nsubj	5:nsubj	_
2	Bush	Bush	PROPN	NNP	Number=Sing	1	flat	1:flat	_
3	on	on	ADP	IN	_	4	case	4:case	_
4	Tuesday	Tuesday	PROPN	NNP	Number=Sing	5	obl	5:obl:on	_
5	nominated	nominate	VERB	VBD	Mood=Ind|Tense=Past|VerbForm=Fin	0	root	0:root	_
6	two	two	NUM	CD	NumType=Card	7	nummod	7:nummod	_
7	individuals	individual	NOUN	NNS	Number=Plur	5	obj	5:obj	_
8	to	to	PART	TO	_	9	mark	9:mark	_
9	replace	replace	VERB	VB	VerbForm=Inf	5	advcl	5:advcl:to	_
10	retiring	retire	VERB	VBG	VerbForm=Ger	11	amod	11:amod	_
11	jurists	jurist	NOUN	NNS	Number=Plur	9	obj	9:obj	_
12	on	on	ADP	IN	_	14	case	14:case	_
13	federal	federal	ADJ	JJ	Degree=Pos	14	amod	14:amod	_
14	courts	court	NOUN	NNS	Number=Plur	11	nmod	11:nmod:on	_
15	in	in	ADP	IN	_	18	case	18:case	_
16	the	the	DET	DT	Definite=Def|PronType=Art	18	det	18:det	_
17	Washington	Washington	PROPN	NNP	Number=Sing	18	compound	18:compound	_
18	area	area	NOUN	NN	Number=Sing	14	nmod	14:nmod:in	SpaceAfter=No
19	.	.	PUNCT	.	_	5	punct	5:punct	_
```

Known as the **CoNLL-U format**
https://universaldependencies.org/format.html

Do not re-invent the wheel.

- If you can, avoid parsing them manually!
- There are Python libraries. Import and use them.
  - CSV & TSV: `pandas`
  - HTML & XML: Beautiful Soup (`bs4`)
  - JSON:
    - `json` library
    - `pandas.read_json`
- NLP-specific formats (Treebank, Universal Dependency, CoNLL):
  - Look at NLTK, see if it has reader
  - If not, chances are there is parser library written by someone somewhere (likely on GitHub)

Data-mining web & social media

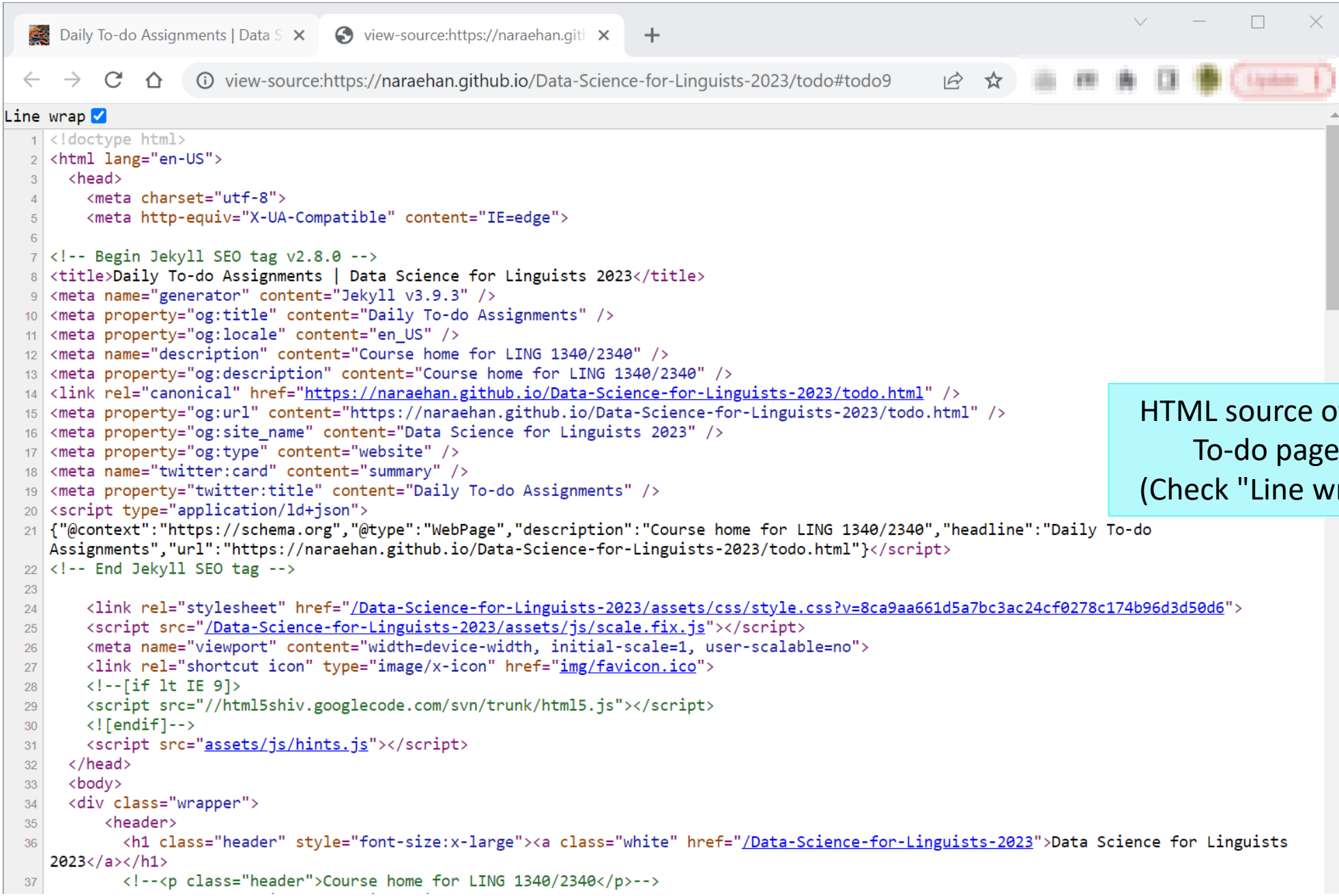
- Twitter sample corpus
  - Static corpus: download from the NLTK data page
- How does one data-mine Twitter?
  - Answer: through **API** (Application Program Interface)
  - Getting acquainted with JSON format
  - Tutorials on on the Learning Resource page
- Libraries used: `tweepy`, `json`

If you can pay for it…
(RIP free Twitter API)

Web mining

- Involves "web crawling" "web spyder", ...
- `scrapy` is the most popular library.
  - https://scrapy.org/

  ← You will have to install it first.
- You have collected a set of web pages. Now what?
  - A web page typically has tons of non-text, extraneous data such as headers, scripts, etc.
    - Example: https://naraehan.github.io/Data-Science-for-Linguists-2023/todo
  - You will need to parse each page to extract textual data.
  - Beautiful Soup (bs4) is capable of parsing XML and HTML files.
- OK, so you've processed the web pages as data. Now what?
  - Linguistic analysis?



HTML source of our To-do page. (Check "Line wrap")

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]
```

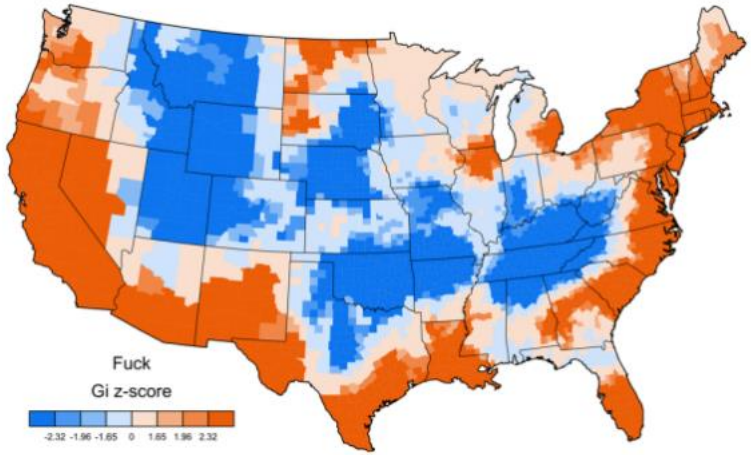
Out[3]:
```
'{"contributors": null, "coordinates": null, "text": "#FollowFriday @France_Int
e @PKuchly57 @Milipol_Paris for being top engaged members in my community this
 week :)", "user": {"time_zone": "Paris", "profile_background_image_url": "htt
```

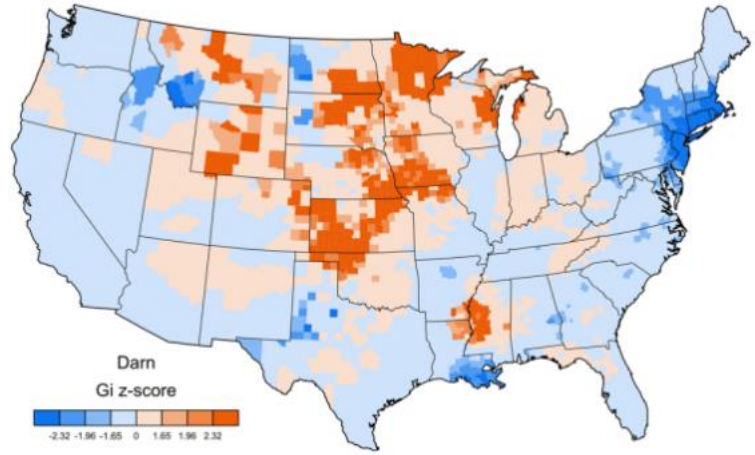
In [5]:
```
# using json library to read line.
import json
json.loads(jlines[0])
```

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',
```

Mining social media for swear words

- https://stronglang.wordpress.com/2015/07/28/mapping-the-united-swears-of-america/
  - Jack Grieve mined Twitter and mapped prominent swear words by geographic regions within the US





Wrapping up

- Next class
  - To-do #9: try out web scraping with BeautifulSoup
  - Corpus linguistics, annotation
- Your project
  - Progress Report #1 specs published
  - Work on it! Focus on DATA.