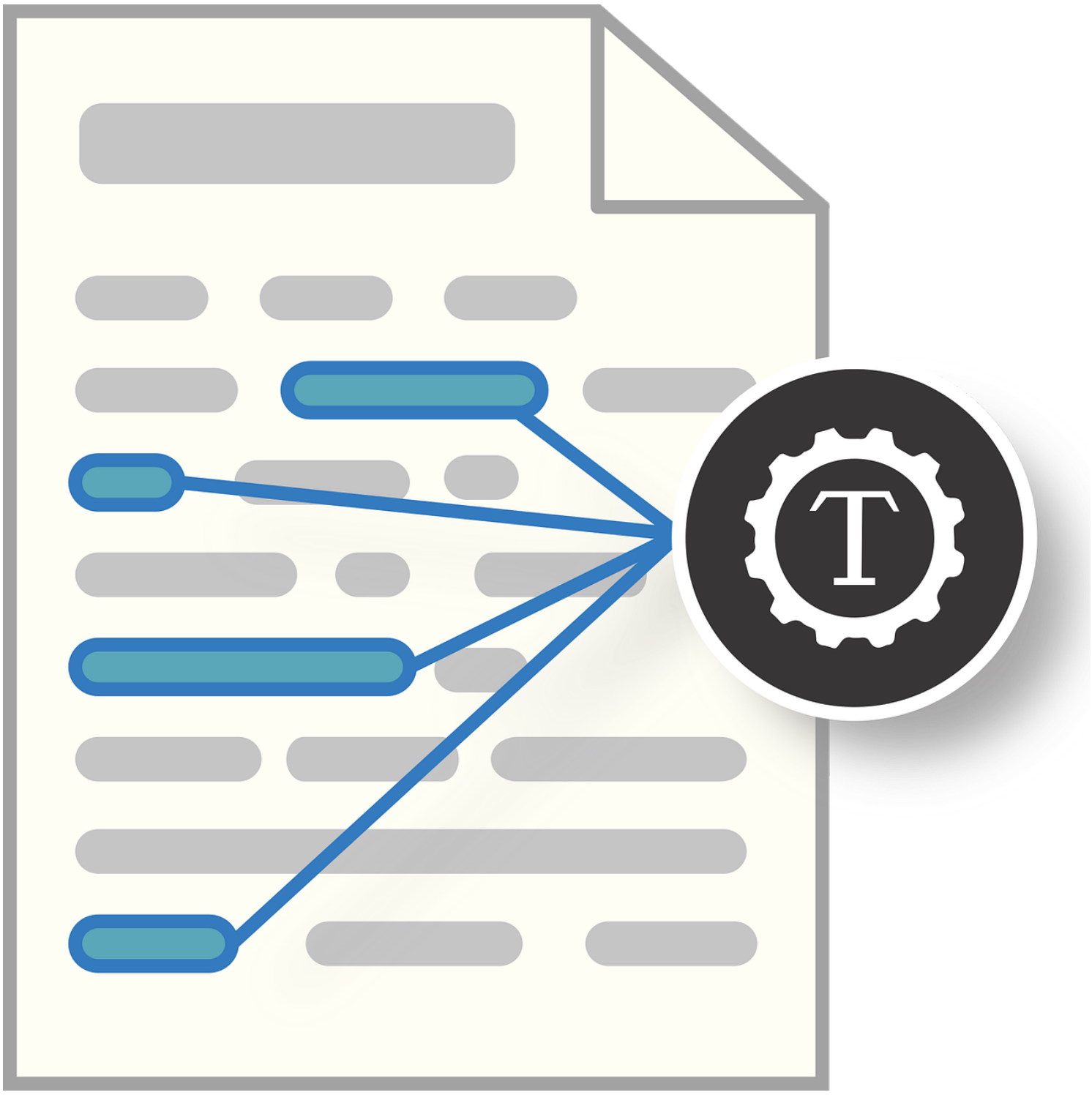

Photo credit: Pixabay

Named Entity Recognition with NLTK and SpaCy

NER is used in many fields in Natural Language Processing (NLP)

Susan Li Follow

```
power/NN
in/IN
the/DT
mobile/JJ
phone/NN
market/NN
and/CC
ordered/VBD
the/DT
company/NN
to/TO
alter/VB
its/PRP$
practices/NNS)
```

Figure 5

Google is recognized as a person. It's quite disappointing, don't you think so?

SpaCy

SpaCy's named entity recognition has been trained on the OntoNotes 5 corpus and it supports the following entity types:

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including "%".
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	"first", "second", etc.
CARDINAL	Numerals that do not fall under another type.

Figure 6 (Source: SpaCy)

Entity

```
import spacy
from spacy import displacy
from collections import Counter
import en_core_web_sm
nlp = en_core_web_sm.load()
```

We are using the same sentence, “European authorities fined Google a record $5.1 billion on Wednesday for abusing its power in the mobile phone market and ordered the company to alter its practices.”

One of the nice things about Spacy is that we only need to apply nlp once, the entire background pipeline will return the objects.

```
doc = nlp('European authorities fined Google a record $5.1 billion
on Wednesday for abusing its power in the mobile phone market and
ordered the company to alter its practices')
pprint([(X.text, X.label_) for X in doc.ents])
```

```
[('European', 'NORP'),
 ('Google', 'ORG'),
 ('$5.1 billion', 'MONEY'),
 ('Wednesday', 'DATE')]
```

Figure 7

European is NORD (nationalities or religious or political groups), Google is an organization, $5.1 billion is monetary value and Wednesday is a date object. They are all correct.

Token

During the above example, we were working on entity level, in the following example, we are demonstrating token-level entity annotation using the BILUO tagging scheme to describe the entity boundaries.

TAG	DESCRIPTION
`B EGIN`	The first token of a multi-token entity.
`I N`	An inner token of a multi-token entity.
`L AST`	The final token of a multi-token entity.
`U NIT`	A single-token entity.
`O UT`	A non-entity token.

Figure 8 (Source: SpaCy)

```
pprint([(X, X.ent_iob_, X.ent_type_) for X in doc])
```

```
[(European, 'B', 'NORP'),
 (authorities, 'O', ''),
 (fined, 'O', ''),
 (Google, 'B', 'ORG'),
 (a, 'O', ''),
 (record, 'O', ''),
 ($, 'B', 'MONEY'),
 (5.1, 'I', 'MONEY'),
 (billion, 'I', 'MONEY'),
 (on, 'O', ''),
 (Wednesday, 'B', 'DATE'),
 (for, 'O', ''),
 (abusing, 'O', ''),
 (its, 'O', ''),
 (power, 'O', ''),
 (in, 'O', ''),
 (the, 'O', ''),
 (mobile, 'O', ''),
 (phone, 'O', ''),
 (market, 'O', ''),
 (and, 'O', ''),
 (ordered, 'O', ''),
 (the, 'O', ''),
 (company, 'O', ''),
 (to, 'O', ''),
 (alter, 'O', ''),
 (its, 'O', ''),
 (practices, 'O', '')]
```

Figure 9

`"B"` means the token begins an entity, `"I"` means it is inside an entity, `"O"` means it is outside an entity, and `""` means no entity tag is set.

Extracting named entity from an article

Now let's get serious with SpaCy and extracting named entities from a New York Times article, — "F.B.I. Agent Peter Strzok, Who Criticized Trump in Texts, Is Fired."

```
from bs4 import BeautifulSoup
import requests
import re

def url_to_string(url):
    res = requests.get(url)
    html = res.text
    soup = BeautifulSoup(html, 'html5lib')
    for script in soup(["script", "style", 'aside']):
        script.extract()
    return " ".join(re.split(r'[\n\t]+', soup.get_text()))

ny_bb =
url_to_string('https://www.nytimes.com/2018/08/13/us/politics/peter-
strzok-fired-fbi.html?
hp&action=click&pgtype=Homepage&clickSource=story-
heading&module=first-column-region&region=top-news&WT.nav=top-news')
article = nlp(ny_bb)
len(article.ents)
```

188

There are 188 entities in the article and they are represented as 10 unique labels:

```
labels = [x.label_ for x in article.ents]
Counter(labels)
```

```
Counter({'CARDINAL': 5,
         'DATE': 29,
         'EVENT': 1,
         'GPE': 35,
         'LOC': 1,
         'NORP': 5,
         'ORDINAL': 1,
         'ORG': 26,
         'PERSON': 84,
         'WORK_OF_ART': 1})
```

Figure 10

The following are three most frequent tokens.

```
items = [x.text for x in article.ents]
Counter(items).most_common(3)
```

```
[('Strzok', 32), ('F.B.I.', 17), ('Trump', 10)]
```

Figure 11

Let's randomly select one sentence to learn more.

```
sentences = [x for x in article.sents]
print(sentences[20])
```

```
Firing Mr. Strzok, however, removes a favorite target of Mr. Trump from the ranks of the F.B.I. and gives Mr. Bowdich and the F.B.I. director, Christopher A. Wray, a chance to move beyond the president's ire.
```

Figure 12

Let's run `displacy.render` to generate the raw markup.

```
displacy.render(nlp(str(sentences[20])), jupyter=True, style='ent')
```

Firing Mr. Strzok PERSON , however, removes a favorite target of Mr. Trump PERSON from the ranks of the F.B.I. GPE and gives Mr. Bowdich PERSON and the F.B.I. GPE director, Christopher A. Wray PERSON , a chance to move beyond the president's ire.

Figure 13

One miss-classification here is F.B.I. It is hard, isn't it?

Using spaCy's built-in displaCy visualizer, here's what the above sentence and its dependencies look like:

```
displacy.render(nlp(str(sentences[20])), style='dep', jupyter =
True, options = {'distance': 120})
```

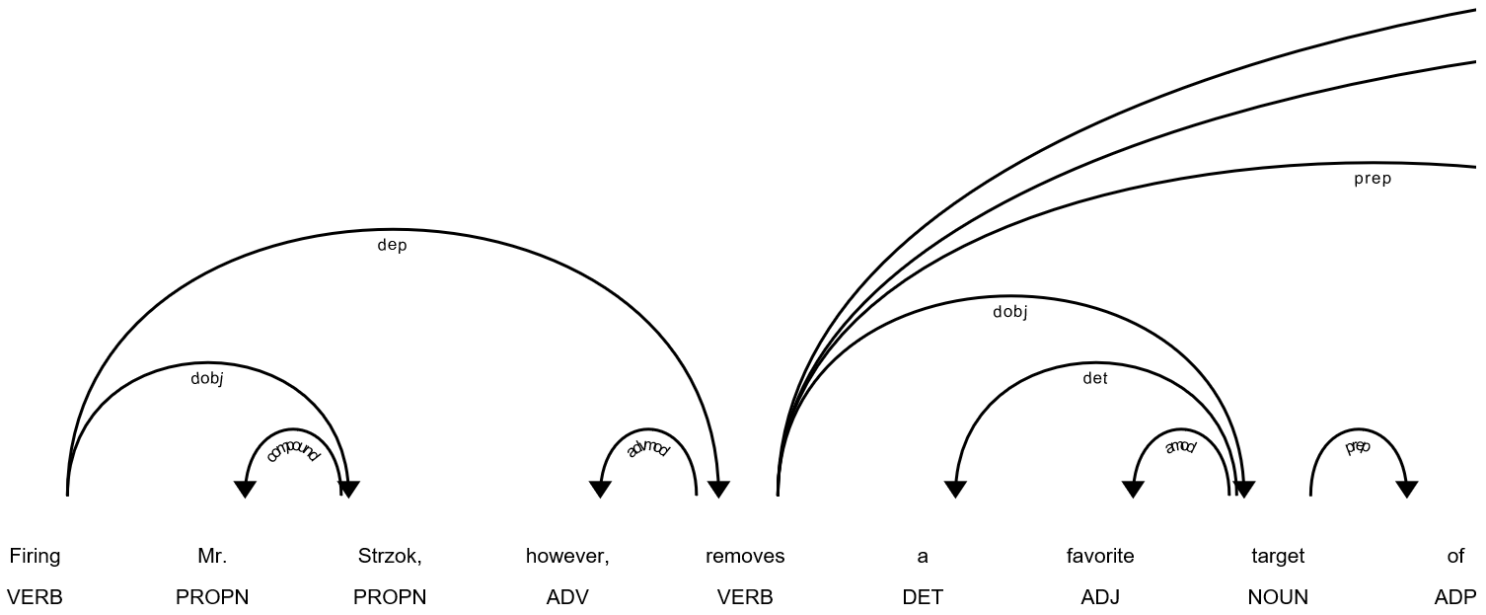

Figure 14

Next, we verbatim, extract part-of-speech and lemmatize this sentence.

```
[(x.orth_,x.pos_, x.lemma_) for x in [y
                                      for y
                                      in nlp(str(sentences[20]))
                                      if not y.is_stop and y.pos_ !=
'PUNCT']]
```

```
[('Firing', 'VERB', 'fire'),
 ('Mr.', 'PROPN', 'mr.'),
 ('Strzok', 'PROPN', 'strzok'),
 ('removes', 'VERB', 'remove'),
 ('favorite', 'ADJ', 'favorite'),
 ('target', 'NOUN', 'target'),
 ('Mr.', 'PROPN', 'mr.'),
```

```
('Trump', 'PROPN', 'trump'),
('ranks', 'NOUN', 'rank'),
('F.B.I.', 'PROPN', 'f.b.i.'),
('gives', 'VERB', 'give'),
('Mr.', 'PROPN', 'mr.'),
('Bowdich', 'PROPN', 'bowdich'),
('F.B.I.', 'PROPN', 'f.b.i.'),
('director', 'NOUN', 'director'),
('Christopher', 'PROPN', 'christopher'),
('A.', 'PROPN', 'a.'),
('Wray', 'PROPN', 'wray'),
('chance', 'NOUN', 'chance'),
('president', 'NOUN', 'president'),
('’s', 'PART', '’s'),
('ire', 'NOUN', 'ire')]
```

Figure 15

```
dict([(str(x), x.label_) for x in nlp(str(sentences[20])).ents])
```

```
{'Bowdich': 'PERSON',
 'Christopher A. Wray': 'PERSON',
 'F.B.I.': 'GPE',
 'Strzok': 'PERSON',
 'Trump': 'PERSON'}
```

Figure 16

Named entity extraction are correct except “F.B.I”.

```
print([(x, x.ent_iob_, x.ent_type_) for x in sentences[20]])
```

```
[(Firing, 'O', ''), (Mr., 'O', ''), (Strzok, 'B', 'PERSON'), (,, 'O', ''), (however, 'O', ''), (,, 'O', ''), (removes, 'O',
''), (a, 'O', ''), (favorite, 'O', ''), (target, 'O', ''), (of, 'O', ''), (Mr., 'O', ''), (Trump, 'B', 'PERSON'), (from, 'O',
''), (the, 'O', ''), (ranks, 'O', ''), (of, 'O', ''), (the, 'O', ''), (F.B.I., 'B', 'GPE'), (and, 'O', ''), (gives, 'O', ''),
(Mr., 'O', ''), (Bowdich, 'B', 'PERSON'), (and, 'O', ''), (the, 'O', ''), (F.B.I., 'B', 'GPE'), (director, 'O', ''), (,, 'O',
''), (Christopher, 'B', 'PERSON'), (A., 'I', 'PERSON'), (Wray, 'I', 'PERSON'), (,, 'O', ''), (a, 'O', ''), (chance, 'O', ''),
(to, 'O', ''), (move, 'O', ''), (beyond, 'O', ''), (the, 'O', ''), (president, 'O', ''), (’s, 'O', ''), (ire, 'O', ''), (.,
'O', '')]
```

Figure 17

Finally, we visualize the entity of the entire article.

F.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is Fired GPE - The New York Times ORG SectionsSEARCHSkip to contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG byF.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York

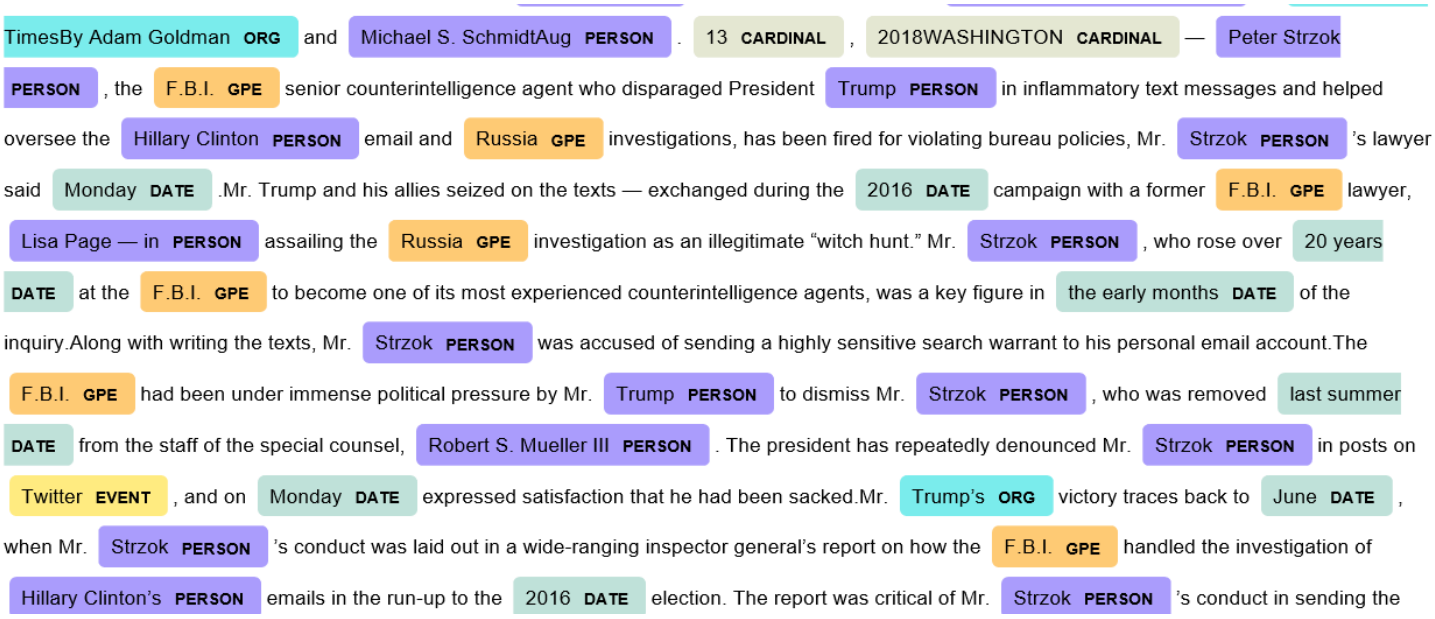

Figure 18

Try it yourself. It was fun! Source code can be found on Github. Happy Friday!

Machine Learning NLP Named Entity Recognition Python Towards Data Science