NLP

Bo Coleman

Note: Yes, the notebook from the video is not provided, I leave it to you to make your own :) it's your final assignment for the semester. Enjoy!

In [17]:
```
#!pip install spacy
```

In [4]:
```
import spacy
```

In [36]:
```
#!python -m spacy download en_core_web_sm
# !python -m spacy download en_core_web_lg
```

In [37]:
```
# nlp = spacy.load("en_core_web_sm")
nlp = spacy.load('en_core_web_lg')
```

In [38]:
```
text = 'What is up my dudes, my name is Bo.  I live in NYC.  I like turtles.  I am orig
```

In [39]:
```
processed_text = nlp(text)
processed_text
```

Out[39]:
```
What is up my dudes, my name is Bo.  I live in NYC.  I like turtles.  I am originally fr
om Arlington, Virginia
```

Sentences

In [40]:
```
n = 0
for sentence in processed_text.sents:
    print(n, sentence)
    n+=1
```

```
0 What is up my dudes, my name is Bo.
1  I live in NYC.
2  I like turtles.
3  I am originally from Arlington, Virginia
```

Words and Punctuation - Along with POS tagging

In [41]:
```
n = 0
for sentence in processed_text.sents:
    for token in sentence:
        print(n, token, token.pos_, token.lemma_)
        n+= 1
```

```
0 What PRON what
1 is AUX be
```

```
2 up ADP up
3 my PRON my
4 dudes NOUN dude
5 , PUNCT ,
6 my PRON my
7 name NOUN name
8 is AUX be
9 Bo PROPN Bo
10 . PUNCT .
11   SPACE
12 I PRON I
13 live VERB live
14 in ADP in
15 NYC PROPN NYC
16 . PUNCT .
17   SPACE
18 I PRON I
19 like VERB like
20 turtles NOUN turtle
21 . PUNCT .
22   SPACE
23 I PRON I
24 am AUX be
25 originally ADV originally
26 from ADP from
27 Arlington PROPN Arlington
28 , PUNCT ,
29 Virginia PROPN Virginia
```

Entities

In [42]:

```python
for entity in processed_text.ents:
    print(entity, entity.label_)
```

```
NYC LOC
Arlington GPE
Virginia GPE
```

Noun Chunks

In [44]:

```python
for noun_chunk in processed_text.noun_chunks:
    print(noun_chunk)
```

```
What
my dudes
my name
Bo
I
NYC
I
turtles
I
Arlington
Virginia
```

Syntactic Depensy Parsing

In [45]:
```
def pr_tree(word, level):
    if word.is_punct:
        return
    for child in word.lefts:
        pr_tree(child, level+1)
    print('\t'* level + word.text + ' - ' + word.dep_)
    for child in word.rights:
        pr_tree(child, level+1)
```

In [46]:
```
for sentence in processed_text.sents:
    pr_tree(sentence.root, 0)
    print('-------------------------------------------')
```

```
                What - nsubj
        is - advcl
                up - prep
                                my - poss
                        dudes - pobj
                my - poss
        name - nsubj
is - ROOT
        Bo - attr
-------------------------------------------
          - dep
        I - nsubj
live - ROOT
        in - prep
                NYC - pobj
-------------------------------------------
  - dep
                I - nsubj
        like - prep
                turtles - pobj
-------------------------------------------
          - dep
        I - nsubj
am - ROOT
        originally - advmod
        from - prep
                Arlington - pobj
                        Virginia - appos
-------------------------------------------
```

Word Vectorization

Only print the first 2 to save space in html output:

In [336…
```
n = 0
for sent in proc_fruits.sents:
    for token in sent:
        if n < 3:
            print(token, token.vector)
        n += 1
```

```
I [ 1.8733e-01  4.0595e-01 -5.1174e-01 -5.5482e-01  3.9716e-02  1.2887e-01
  4.5137e-01 -5.9149e-01  1.5591e-01  1.5137e+00 -8.7020e-01  5.0672e-02
  1.5211e-01 -1.9183e-01  1.1181e-01  1.2131e-01 -2.7212e-01  1.6203e+00
```

```
 -2.4884e-01  1.4060e-01  3.3099e-01 -1.8061e-02  1.5244e-01 -2.6943e-01
 -2.7833e-01 -5.2123e-02 -4.8149e-01 -5.1839e-01  8.6262e-02  3.0818e-02
 -2.1253e-01 -1.1378e-01 -2.2384e-01  1.8262e-01 -3.4541e-01  8.2611e-02
  1.0024e-01 -7.9550e-02 -8.1721e-01  6.5621e-03  8.0134e-02 -3.9976e-01
 -6.3131e-02  3.2260e-01 -3.1625e-02  4.3056e-01 -2.7270e-01 -7.6020e-02
  1.0293e-01 -8.8653e-02 -2.9087e-01 -4.7214e-02  4.6036e-02 -1.7788e-02
  6.4990e-02  8.8451e-02 -3.1574e-01 -5.8522e-01  2.2295e-01 -5.2785e-02
 -5.5981e-01 -3.9580e-01 -7.9849e-02 -1.0933e-02 -4.1722e-02 -5.5576e-01
  8.8707e-02  1.3710e-01 -2.9873e-03 -2.6256e-02  7.7330e-02  3.9199e-01
  3.4507e-01 -8.0130e-02  3.3451e-01  2.7063e-01 -2.4544e-02  7.2576e-02
 -1.8120e-01  2.3693e-01  3.9977e-01  4.5012e-01  2.7179e-02  2.7400e-01
  1.4791e-01 -5.8324e-03  9.5910e-01 -1.0129e+00  2.0699e-01  1.8237e-01
 -2.5234e-01 -2.6261e-01 -3.4799e-01 -2.4051e-02  4.4470e-01  5.9226e-02
  4.5561e-01  1.9700e-01 -4.8327e-01  8.9523e-02 -2.2373e-01 -1.5654e-01
  2.1578e-01  1.1673e-01  8.2006e-02 -8.0735e-01  2.3903e-01 -5.1304e-01
 -3.3888e-01 -3.1499e-01 -1.7272e-01 -6.7020e-01  2.7096e-01 -4.3241e-01
  4.3103e-02  2.1233e-02  1.3350e-02 -6.3938e-02 -2.4957e-01 -2.4938e-01
  3.4812e-01 -7.1321e-02  2.3375e-01 -9.5384e-02  5.2488e-01  6.8175e-01
 -1.0214e-01 -1.4914e-01 -7.5697e-02  1.7248e-01  2.5440e-01  1.5760e-01
 -5.9125e-01  2.4300e-01  6.3962e-01 -9.3280e-02 -2.7914e-01 -6.6262e-02
 -6.7170e-02 -4.0929e-01 -3.0300e+00  1.8250e-01  2.0113e-01  6.0628e-02
 -2.4769e-01  5.5324e-02 -4.9106e-01  3.1544e-01 -3.4231e-01 -6.3766e-01
 -3.6129e-01 -5.9029e-02  1.5510e-01  4.4577e-02  2.3572e-01 -1.7095e-01
 -2.2749e-01 -2.3184e-02  2.3868e-01  2.8170e-02  4.2965e-01 -1.2458e-01
 -3.6972e-02  2.0061e-01 -3.1405e-01 -8.5287e-02 -3.3496e-01 -9.7047e-02
 -1.4388e-01  1.1147e-01 -4.5232e-01 -2.4217e-01 -1.8245e-01 -6.7292e-01
  2.1933e-02 -5.4816e-02 -4.6508e-01  4.7767e-01 -2.4752e-01 -1.5790e-01
  1.1817e-01  5.6851e-02 -4.9151e-01  1.5496e-01  1.6425e-02  4.1650e-02
 -3.4990e-01 -1.5979e-01  3.9705e-01  2.2963e-01  2.4688e-01  1.9567e-02
 -2.8802e-01 -6.9983e-01  3.2744e-01  1.0833e-01  2.4945e-01 -7.8653e-01
 -6.1379e-02 -3.7359e-01 -1.1603e-01 -2.4950e-01  1.0161e-01  3.3994e-02
  1.5650e-01  2.1344e-01 -1.1094e-01 -5.7687e-03  1.7869e-01 -1.0127e-01
 -1.6891e-02  3.0001e-01 -3.4116e-01 -3.2390e-02  4.2514e-02  1.1850e-01
 -1.8337e-01 -6.2865e-01 -2.8021e-01  4.2351e-01  1.1277e-01  1.2121e-03
  1.5710e-01 -3.6321e-01 -4.9251e-01  1.1653e-01  2.4024e-01  1.7712e-01
  6.8700e-02 -4.4137e-01 -2.9877e-01 -1.2071e-02  2.8325e-01  1.0668e-01
 -1.8859e-01 -4.1345e-01 -3.4090e-01  4.7236e-02 -3.8309e-01  4.3572e-01
  2.4505e-01  2.7337e-01 -7.3038e-02  4.2514e-01 -3.2455e-02 -3.5211e-01
  4.5691e-01  1.9433e-01 -1.5230e-01  4.2675e-01  2.8795e-01 -5.5969e-01
 -1.3031e-01  8.9844e-02  4.2605e-01 -1.9632e-01 -7.1989e-02 -8.0189e-02
 -3.0425e-01 -4.6190e-01  2.8178e-01 -9.9872e-02  3.5097e-01  1.6123e-01
 -3.6548e-02 -3.6739e-01 -1.9819e-02  3.2130e-01  1.7479e-01  2.5175e-01
 -7.6439e-03 -9.3786e-02 -3.7852e-01  4.3725e-01  2.1288e-01  2.5096e-01
 -1.9613e-01 -2.8865e-01 -5.6726e-03  4.2795e-01  2.0625e-01 -3.7701e-02
 -1.2200e-01 -7.9253e-02 -1.0290e-01  1.0558e-02  4.9880e-01  2.5382e-01
  1.5526e-01  1.7951e-03  1.1633e-01  7.9300e-02 -3.9142e-01 -3.2483e-01
  6.3451e-01 -1.8910e-01  5.4050e-02  1.6495e-01  1.8757e-01  5.3874e-01]
think [-2.1788e-01  4.4128e-01 -4.3204e-01 -1.9803e-01 -2.7968e-03  2.8803e-01
  8.0648e-02 -1.4643e-01  2.2300e-02  2.5941e+00 -2.5959e-01 -8.4183e-02
  4.2613e-01  8.7662e-03 -3.3843e-01 -1.7814e-01 -4.2161e-01  3.3757e-01
 -3.9092e-01 -6.0522e-02  2.9517e-01  1.4590e-01  3.2846e-01 -5.8106e-02
 -1.9982e-01  1.1735e-01 -3.0825e-01 -2.2648e-01  4.7433e-01 -2.4415e-01
 -3.0177e-01  6.7390e-01  6.5974e-02  4.5855e-02 -3.6646e-02  9.9892e-02
  1.3664e-01  1.6360e-01 -1.7109e-01 -2.4558e-01 -1.2538e-01  2.1834e-01
 -1.5026e-01  1.3336e-01  3.3971e-01  9.4071e-02 -5.1316e-01 -1.5819e-01
  2.7468e-02 -3.4402e-02 -5.5910e-02  6.9633e-02 -1.2256e-02 -5.1804e-02
  2.7225e-01 -1.8355e-01 -2.4559e-01 -3.1370e-01  1.5620e-01  6.3014e-02
 -3.2111e-01 -5.1905e-01 -1.1712e-01  3.6818e-01  4.0482e-02 -4.3880e-01
 -8.3865e-02  9.5061e-02  3.1995e-01  5.2088e-02  2.3889e-01 -1.6807e-03
  3.1965e-01 -6.8731e-02  1.3477e-01  3.2888e-01  2.7228e-02 -1.9567e-01
```

```
 -1.0522e-01  4.3544e-01  1.8869e-02 -2.4586e-02  9.5041e-02 -1.0182e-01
  2.2237e-01 -3.9997e-01  7.7279e-02 -8.7118e-01  3.3649e-01 -2.3464e-01
 -1.7064e-01 -1.6163e-01 -2.1596e-01  4.3201e-01  2.2237e-01  5.4215e-02
  2.4430e-01  8.3594e-02  9.7403e-03  1.6315e-01 -1.7864e-01 -7.8538e-02
 -3.2577e-02 -1.3266e-01  4.1890e-01 -7.7905e-01 -2.1269e-01 -2.8179e-01
 -3.6263e-01  2.7969e-01  5.1118e-02 -4.3791e-01 -6.6222e-02 -2.9007e-01
  9.8879e-02  6.8701e-02 -1.7953e-01 -2.4516e-01  6.2370e-02 -1.8344e-01
  2.3848e-01 -3.8926e-01  1.4563e-02 -1.5011e-01  4.1463e-01  2.3519e-01
  1.5768e-01 -4.2338e-01 -5.9981e-02 -8.5539e-02 -1.9645e-01  1.6238e-01
  1.3915e-02  3.5895e-02  1.5087e-01 -1.2057e-01  2.5404e-03 -3.2900e-01
  1.1253e-01 -1.1478e-01 -2.2095e+00  2.1364e-01  1.4400e-02  3.3077e-01
 -7.3231e-02 -3.3165e-01 -3.4116e-01  1.9457e-01 -2.5111e-01 -2.9986e-01
 -2.3206e-02 -4.6035e-02 -8.0109e-02  8.9126e-02  5.2734e-02 -9.5490e-02
  2.0326e-02 -3.4696e-01 -1.3078e-01  4.9967e-02 -3.3681e-01  2.7430e-01
 -4.5855e-01 -5.0902e-02 -2.4579e-01 -9.9564e-02  2.7532e-01 -2.2780e-01
  1.1668e-01 -9.2933e-02 -1.3185e-01 -2.0557e-01  6.7947e-02 -4.6485e-01
  8.9792e-02 -1.6480e-02 -1.4847e-01  1.8379e-01  1.9782e-01 -8.6026e-02
  9.0624e-02 -8.4892e-02 -3.7597e-01 -1.9855e-01 -1.2090e-02  1.2820e-02
  7.7705e-02 -1.2059e-01  1.1353e-01  2.9137e-01  1.0847e-01  1.3505e-01
 -2.9519e-01 -3.0900e-01  1.1161e-01 -4.1132e-02 -7.6808e-02 -2.6873e-01
  9.1791e-02  3.3636e-01  1.9916e-01 -1.8500e-01 -1.0462e-01 -3.0590e-01
  1.5874e-01  9.2589e-02  2.1171e-02 -1.8780e-01  2.0077e-01  2.4509e-01
 -2.2700e-01 -2.1141e-01  1.9871e-02 -4.0445e-01  2.5579e-01  2.2388e-01
 -2.2641e-01 -7.4679e-02 -2.8030e-01 -1.1511e-01  1.2736e-01  1.9723e-01
  3.1854e-02  3.5542e-02  1.6587e-01  1.0235e-01  2.4897e-01  1.0350e-01
 -4.2974e-02 -8.1860e-02 -4.3416e-01  1.0390e-01 -1.6777e-02  1.6120e-01
  7.4300e-02 -9.9311e-02  1.8984e-01 -1.2747e-01 -6.0094e-02  4.9008e-01
  1.0554e-01 -6.0390e-02  2.0226e-02  5.3835e-01 -7.1150e-02 -1.4458e-01
  8.6395e-02 -5.1988e-02 -3.2138e-01  4.9567e-01  2.9081e-01 -2.8324e-01
 -1.6194e-01  1.0306e-02 -3.6499e-01 -4.5294e-02 -1.7151e-01 -5.6910e-02
  1.6989e-01  1.8708e-01  4.0052e-01  2.4493e-01  1.2178e-01  3.4254e-01
  1.5890e-01  8.0175e-02 -1.1652e-01  3.3864e-01  4.5713e-01  5.0961e-01
 -1.7444e-01 -1.1862e-02 -9.4227e-02 -4.2007e-01 -2.2938e-01  8.4131e-02
  1.8915e-01 -3.5540e-01 -1.7737e-01  3.4414e-01 -1.9804e-01 -2.3456e-01
  2.3658e-02 -4.2666e-02  7.3081e-02 -2.1149e-02  3.0316e-01 -2.9115e-01
  8.3060e-02  3.4784e-02 -1.5084e-01  2.7544e-02 -2.7939e-01  3.8548e-02
  2.2003e-01  1.8208e-01 -5.0746e-01 -1.6472e-01  3.2255e-01  3.0579e-01]
green [-7.2368e-02  2.3320e-01  1.3726e-01 -1.5663e-01  2.4844e-01  3.4987e-01
 -2.4170e-01 -9.1426e-02 -5.3015e-01  1.3413e+00 -8.6785e-01 -1.3183e-01
 -5.9679e-01 -3.4415e-01 -1.6121e-01 -9.2512e-04  5.3267e-01  2.1329e+00
  2.1933e-02 -5.1933e-01  3.6557e-01 -1.2978e-02 -2.7154e-01  4.8964e-03
 -1.1849e-01 -3.8338e-01 -4.8944e-01  4.9147e-01  1.3664e-01 -9.6163e-02
 -2.8429e-02  3.9630e-03  1.5542e-01 -2.9680e-01 -1.4895e-01 -5.5311e-02
  3.0003e-01  1.6376e-01 -1.6941e-01 -1.0166e-01  5.2141e-01  8.5416e-02
  1.6017e-02 -7.9741e-02  1.5934e-01  8.6290e-02 -2.1192e-01 -8.0312e-03
  2.0699e-01 -2.0541e-01 -1.3612e-01  2.4044e-02 -1.7975e-02 -2.7537e-01
  5.5046e-01 -7.4320e-01 -1.0718e-01  8.3590e-01  4.5894e-02 -8.3839e-03
 -3.7027e-01 -3.8694e-01  1.4741e-01 -6.2706e-02  5.5882e-01 -3.5788e-02
 -3.7742e-01 -2.5088e-01 -3.2712e-01 -2.1363e-02 -1.1778e-01  1.0936e-02
 -4.0838e-02  1.9662e-01 -2.0128e-01 -4.7566e-02  1.1487e-01 -9.0004e-02
  1.0354e-01 -5.2373e-01 -1.1288e-01  2.3075e-01  4.3984e-01  5.4876e-01
  1.9629e-01 -2.9718e-01  7.1773e-01  1.3269e+00  6.2276e-02 -8.8419e-02
  3.5253e-01  6.1762e-01  6.2818e-01  2.1847e-02  1.1744e-01  1.4717e-01
  2.4852e-02  3.1065e-01 -3.0706e-02 -4.8994e-01  1.9092e-01 -5.1000e-02
 -1.9395e-01 -4.9768e-01 -3.4417e-01 -8.2097e-01 -4.9253e-01  3.0066e-01
 -1.1905e-01  3.5405e-01 -5.9503e-01 -5.9864e-01 -9.2760e-02 -1.4563e-01
  6.8754e-01  1.8893e-01 -4.6852e-02  1.0246e-01 -8.7789e-02 -3.2801e-01
  3.1215e-01 -1.7373e-01 -3.4827e-01 -1.9547e-01  1.1008e-01  2.2747e-01
  4.4502e-01  8.1171e-02 -3.8463e-03 -1.9223e-01 -1.6651e-01  3.9317e-02
  2.3909e-01 -3.0472e-01 -2.9583e-01 -6.2451e-01  1.0243e-01 -2.3324e-01
```

```
 5.0008e-01  8.9740e-02 -2.1251e+00  2.4246e-01  2.7600e-01  1.1749e-01
 7.1881e-02  1.7860e-01 -4.4795e-03  1.5575e-01 -2.7073e-01 -8.8036e-02
-1.1564e-02 -1.4186e-02  4.9359e-01  1.6096e-01 -4.4652e-01 -2.0159e-01
-3.1921e-01  4.0095e-03 -3.9027e-01  2.6482e-01 -8.7063e-02  3.9982e-01
-3.0174e-01  3.6335e-01  6.5750e-02 -4.8644e-01 -1.8118e-01 -7.6974e-01
 1.7686e-01  3.7618e-01  1.1485e-01  9.7655e-03 -3.1654e-01  7.6573e-02
-2.9506e-01 -2.2645e-01  6.8611e-01  6.6346e-02  2.2698e-01 -2.0357e-01
-1.1136e-01 -3.9789e-02 -3.1132e-01 -3.9395e-01 -2.6340e-01  4.1417e-02
-2.2766e-01 -1.5583e-01 -3.9518e-01 -1.7292e-01  3.4403e-01  4.0990e-01
-9.3649e-02 -1.2536e-01  2.1836e-01  2.7454e-01  2.3929e-01  5.4202e-01
-1.8898e-01  6.1104e-02 -9.9625e-02  6.9587e-02 -1.7275e-01  3.9217e-01
 9.1343e-02  2.5958e-01  5.0131e-01  1.0328e-01  2.8023e-01 -4.2147e-01
-2.3985e-01  5.0814e-01  4.0660e-01 -3.2745e-03  1.3557e-02  2.6442e-01
 1.8914e-02 -1.9332e-02  2.0762e-01 -3.9842e-01 -5.6105e-01 -2.6695e-01
-7.6739e-03  2.8867e-01  3.1247e-01 -4.4065e-03  3.4002e-01 -5.1330e-02
-4.3934e-01  6.1596e-02  1.4591e-01  3.7920e-01  4.3088e-01  3.6122e-01
-2.0847e-01  5.6458e-01 -5.6009e-02 -4.6236e-01  8.1828e-01  8.1877e-01
-1.5978e-01 -3.0881e-01 -5.5235e-01  4.7371e-02 -3.8537e-02  3.7726e-01
 6.0784e-02 -4.3161e-01 -3.3027e-01 -1.8559e-01  1.1674e-01 -1.3420e-01
-2.0262e-01  8.2621e-02  3.2163e-01  2.5451e-01  1.3104e-01  5.2760e-01
-4.7345e-03  1.9238e-01 -6.3701e-02  2.6855e-01  1.2537e-01  6.0333e-01
 3.4068e-01 -3.6425e-01 -3.5315e-01 -4.3298e-01 -4.2086e-01  1.5704e-01
-2.5552e-01  1.6895e-01  7.9552e-02 -3.1513e-01  8.5769e-02 -7.9049e-02
 4.9882e-04  4.1551e-01  1.3062e-01  2.1869e-01  1.7056e-01 -2.3690e-01
-3.9074e-01  5.9123e-02 -8.0229e-02  1.1957e-01  3.7294e-01  3.8980e-01
 4.2767e-01 -1.1234e-01 -4.0517e-01  2.4357e-01  4.3730e-01 -4.6152e-01
-3.5271e-01  3.3625e-01  6.9899e-02 -1.1155e-01  5.3293e-01  7.1268e-01]
```

In [55]:

```python
proc_fruits = nlp('''I think green apples are delicious. While pears have a strange tex
apples, pears, bowls = proc_fruits.sents
```

In [58]:

```python
dude = processed_text.vocab['dudes']
print(apples.similarity(dude))
print(pears.similarity(dude))
print(bowls.similarity(dude))
```

```
0.4225541055202484
0.3517932891845703
0.433989018201828
```

Assignment

Find your favorite news source and grab the article text.

1. Show the most common words in the article.
2. Show the most common words under a part of speech. (i.e. NOUN: {'Bob':12, 'Alice':4,})
3. Find a subject/object relationship through the dependency parser in any sentence.
4. Show the most common Entities and their types.
5. Find Entites and their dependency (hint: entity.root.head)
6. Find the most similar noun (chunks) in the article

I am using the NYTimes as my news source. The article can be found here: https://www.nytimes.com/2022/04/23/health/mental-health-crisis-teens.html

In [83]:
```
import docx2txt
```

In [84]:
```
NYT_text = docx2txt.process('NYTimes_article.docx')
```

In [330...
```
NYT = nlp(NYT_text)
```

1. Show the most common words in the article.

In [ ]:
```
from collections import Counter
```

In [98]:
```
tokens = [token.text for token in NYT if not token.is_stop and not token.is_punct]
token_freqs = Counter(tokens)
token_freqs.most_common(10)
```

Out[98]:
```
[('M', 83),
 ('\n\n', 72),
 ('\xa0', 40),
 ('said', 30),
 ('Linda', 27),
 ('school', 15),
 ('parents', 13),
 ('percent', 13),
 ('health', 11),
 ('Elaniv', 11)]
```

Further cleaning is required to remove the newline characters and other unwanted text.

1. Show the most common words under a part of speech. (i.e. NOUN: {'Bob':12, 'Alice':4,})

In [104...
```
import pandas as pd
```

In [120...
```
token_pos = [token.pos_ for token in NYT if not token.is_stop and not token.is_punct]
NYT_df = pd.DataFrame({'token':tokens, 'type':token_pos})
NYT_df = NYT_df.groupby(['type', 'token']).size().reset_index(name='counts')
most_common = NYT_df.sort_values(['type', 'counts'], ascending=False).groupby('type').h
most_common
```

Out[120...

	type	token	counts
953	VERB	said	30
841	VERB	felt	5
851	VERB	found	5
934	VERB	recalled	5
1025	VERB	went	5
753	SYM	$	1
750	SPACE	\n\n	72

	type	token	counts
752	SPACE		40
751	SPACE	\n\n\n\n	1
749	SCONJ	like	2
709	PROPN	M	82
707	PROPN	Linda	27
692	PROPN	Elaniv	11
690	PROPN	Dr.	6
738	PROPN	Tony	6
655	NUM	2019	5
634	NUM	10	3
639	NUM	15	3
642	NUM	1990	3
637	NUM	13	2
540	NOUN	school	15
485	NOUN	parents	13
492	NOUN	percent	13
401	NOUN	health	11
235	NOUN	adolescents	10
217	INTJ	Hey	2
219	INTJ	Oh	2
220	INTJ	like	2
218	INTJ	Huh	1
216	CCONJ	plus	1
215	AUX	having	2
214	AUX	felt	1
191	ADV	later	4
206	ADV	sharply	4
187	ADV	home	3
208	ADV	socially	3
172	ADV	ago	2
167	ADP	like	4
86	ADJ	mental	9
64	ADJ	high	6

	type	token	counts
135	ADJ	social	6
1	ADJ	Black	4
10	ADJ	adolescent	4

1. Find a subject/object relationship through the dependency parser in any sentence.

Here are three sentences:

In [128…

```
n = 0
for sentence in NYT.sents:
    if n > 1 and n < 5:
        pr_tree(sentence.root, 0)
        print('-------------------------------------------')
    n += 1
```

```
                Moments - npadvmod
        earlier - advmod
                        the - det
                girl - poss
                        ’s - case
        mother - nsubj
                Linda - appos
        had - aux
stolen - ROOT
                a - det
        look - dobj
                at - prep
                                        her - poss
                                daughter - poss
                                        ’s - case
                        smartphone - pobj
-------------------------------------------
                The - det
        teenager - nsubj
        incensed - advcl
                by - agent
                                the - det
                        intrusion - pobj
        had - aux
grabbed - ROOT
                the - det
        phone - dobj
        and - cc
        fled - conj
-------------------------------------------
                The - det
        adolescent - nsubjpass
        is - aux
        being - auxpass
identified - ROOT
        by - agent
                        an - det
                initial - pobj
                        M - appos
```

```
                                 and - cc
                                         the - det
                                 parents - conj
                                         by - prep
                                                         first - amod
                                                 name - pobj
                         only - advmod
                 to - aux
         protect - advcl
                                 the - det
                         family - poss
                                 's - case
                 privacy - dobj
-------------------------------------------
```

1. Show the most common Entities and their types.

In [165…

```
ents = pd.DataFrame({'ent': [entity for entity in NYT.ents],
                    'ent_type':[entity.label_ for entity in NYT.ents]})
ents['ent'] = ents.apply(lambda x: str(x['ent']), axis=1)
most_common_ents = ents.groupby(['ent', 'ent_type']).size().reset_index(name='counts')
most_common_ents = most_common_ents.sort_values('counts', ascending=False).head(10)
most_common_ents
```

Out[165…

	ent	ent_type	counts
63	Linda	PERSON	26
87	Tony	PERSON	6
100	first	ORDINAL	5
117	the United States	GPE	4
66	M	ORG	3
48	Elaniv	PERSON	3
106	one	CARDINAL	3
47	Elaniv	ORG	3
11	1990	DATE	3
49	Elaniv Burnett	PERSON	2

1. Find Entites and their dependency (hint: entity.root.head)

Print the first 30 to save space in html doc:

In [331…

```
n = 0
print('Entity | Dependency')
print('--------------------')
for entity in NYT.ents:
    if n<30:
        print(entity, ' | ', entity.root.head)
    n += 1
```

```
Entity | Dependency
```

```
--------------------
One evening  |  sprang
last April  |  evening
13-year-old  |  girl
Minneapolis  |  in
Linda  |  mother
first  |  name
Linda  |  alarmed
Genocide Jack  |  were
the preceding two years  |  In
Linda  |  watched
American  |  adolescence
Three decades ago  |  came
the United States  |  in
2019  |  In
13 percent  |  reported
60 percent  |  increase
2007  |
2000 to 2007  |  from
nearly 60 percent  |  leaped
2018  |  by
the Centers for Disease Control and Prevention  |  to
December  |  In
U.S.  |  surgeon
Candice Odgers  |  said
the University of California  |  at
Irvine  |  University
Linda  |  realized
Linda  |  talked
Linda  |  jolted
Linda  |  said
```

1. Find the most similar noun (chunks) in the article

Due to runtime, I only compared the first 50 noun chunks

In [203…

```python
import numpy as np
```

In [318…

```python
def do_comparison(chunk1, chunk2):
    t1 = nlp(str(chunk1) +'  . This to make similarity work.')
    t2 = nlp(str(chunk2) +'  . This to make similarity work.')
    s1, s1p = t1.sents
    s2, s2p = t2.sents
    similarity = s1.similarity(s2)
    return similarity
```

In [327…

```python
noun_chunks = [noun_chunk for noun_chunk in NYT.noun_chunks][:50]
nc = pd.DataFrame({'noun_chunk':noun_chunks})
nc['noun_chunk'] = nc.apply(lambda x: str(x['noun_chunk']), axis = 1)
nc = nc.drop_duplicates()
nc = nc.merge(nc, how = 'cross', suffixes=['_1', '_2'])
nc = nc[nc['noun_chunk_1'] != nc['noun_chunk_2']]
nc['similarity'] = nc.apply(lambda x: do_comparison(x['noun_chunk_1'], x['noun_chunk_2'
```

Get rid of consecutive rows where the same two noun chunks are being compared:

In [329…

```
nc.sort_values('similarity', ascending=False).loc[[True, False]*int(nc.shape[0]/2)].hea
```

Out[329…

	noun_chunk_1	noun_chunk_2	similarity
1130	intentional self-harm	self-harm	0.945473
257	the backyard	the patio	0.885707
993	Some	Others	0.874791
130	the living room	the house	0.872049
347	the girl’s mother	her daughter’s smartphone	0.871296
512	The teenager	the girl’s mother	0.870744
1166	Others	who	0.864880
642	The adolescent	The teenager	0.863191
768	the parents	The teenager	0.861646
1703	me	Some	0.852058