Brody Vogel NLP HW due Nov. 5

October 31, 2018

Brody Vogel
NLP Assignment Due November 5th

Overall, it seems like NER does a decent job of giving a high-level summary for some kinds of articles. It does a good job with news articles, for example; for the news articles that I checked, the word clouds did an OK job of summarizing the contents of the article with named entities. For others, NER didn't do a good job at all. Articles that were longer and / or had more abstract topics weren't handled well; for example, this method didn't at all summarize an article I found about "walkability" in cities.

Using fewer named entities in the word clouds seemed to help reduce the clutter in the word clouds for the longer articles. When I did this, though, the word clouds for some of the articles lost a lot of important details. The same can be said for when I restricted the types of named entities that went into the word clouds to be either people ("PERSON") or organizations ("ORG").

What I did, specifically, then:

I grabbed 10 articles / news pieces from the web. Their general and specific topics are:

(Sports) Golden Tate being traded to the Philadelphia Eagles.

(Conservative Politics) Matt Shea running for re-election in Washington state.

(National News) The first funeral for a victim of the Pittsburgh Synagogue shooting.

(Data Science) Popular data science interview questions.

(Climate Change) Cheap ways to remove CO2 from the atmosphere.

(Interview) A New York Times interview with Jerry Seinfeld about scandals in comedy.

(Special Interest) An article looking at the impacts of the "walkability" of a city.

(Public Health) A deep look at the opiod problem and attempts at a solution in Stanwood, Washington.

(Mental Health) An article comparing Facebook groups to group therapy.

(Professional Interest) A list of soft skills programmers should have.

The article about soft skills for programmers is kind of a challenge problem for the algorithm: the article is about computer programmers, but not about actual computer programming; it's about other skills programmers should have. It'll be interesting to see how the algorithm does on that one.

/Also sorry for the length - couldn't figure out how to make the word clouds smaller/

```
In [33]: # import statements
         import spacy
         import os
         from wordcloud import WordCloud
         from matplotlib import pyplot as plt
         import random
```

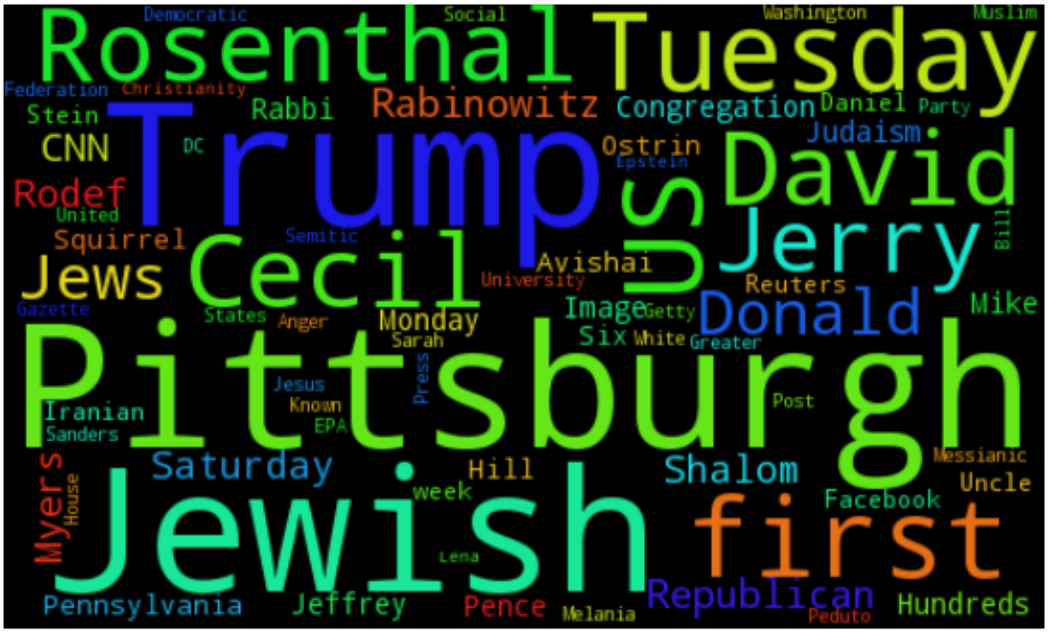
This first chunk creates word clouds from the named entities for each of the above-listed articles. It does a pretty good job on the shorter articles, but the longer pieces produced very jumbled word clouds. It's pretty clear, for example, what's going on in article about data science interview questions, but the focus of the long article about Facebook groups is fairly unclear.

As an aside, the article about soft skills for programmers seems to be causing some problems.

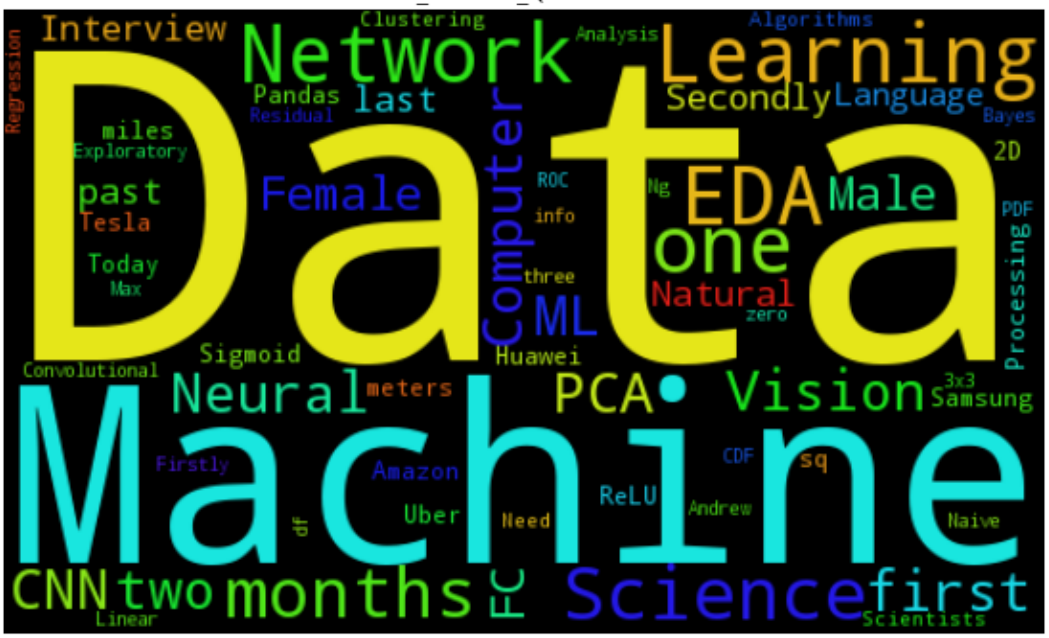
```
In [47]: # get the list of articles from my machine
         articles = os.listdir('/Users/brodyvogel/Desktop/580 Spacy Docs/')

         # load the English model from SpaCy
         nlp = spacy.load('en')

         # loop through the articles and build the word clouds
         for article in articles:
             # read the text of the article
             text = open('/Users/brodyvogel/Desktop/580 Spacy Docs/' + article).read()
             # apply the model from SpaCy
             doc = nlp(text)
             # make a long string for the WordCloud plug-in
             for_word_cloud = ' '.join([str(x) for x in doc.ents])
             # build the Word Cloud
             wordcloud = WordCloud(width = 500, height = 300).generate(for_word_cloud)
             # show the Word Cloud
             plt.figure(figsize = (10, 6))
             plt.title(article)
             plt.imshow(wordcloud, interpolation = "bilinear")
             plt.axis("off")
             plt.show()
```

Conservative_Politics.txt

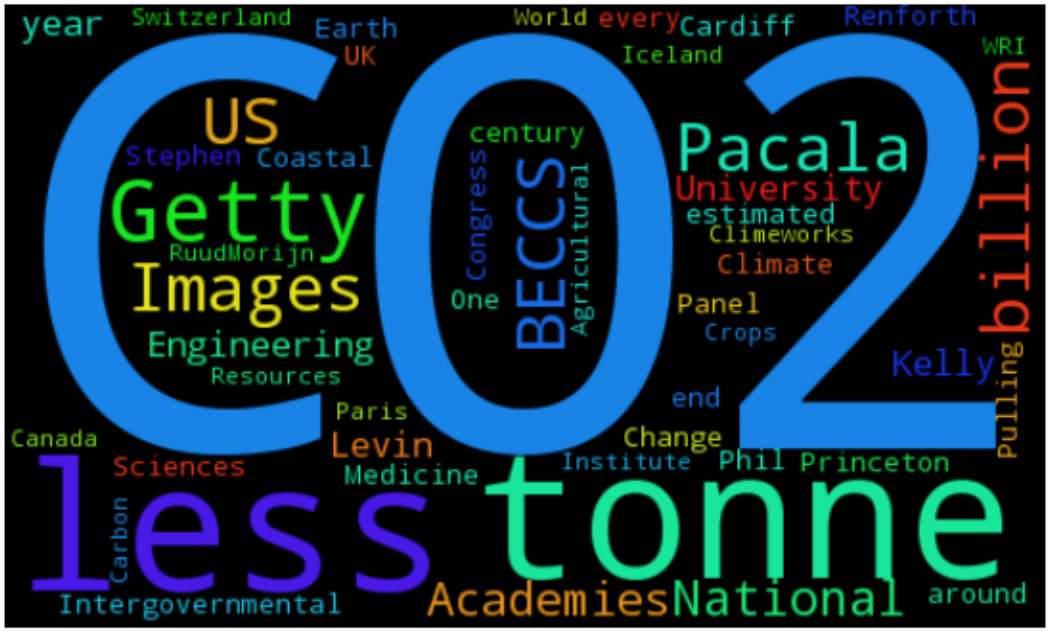
Pittsburgh_Shooting.txt



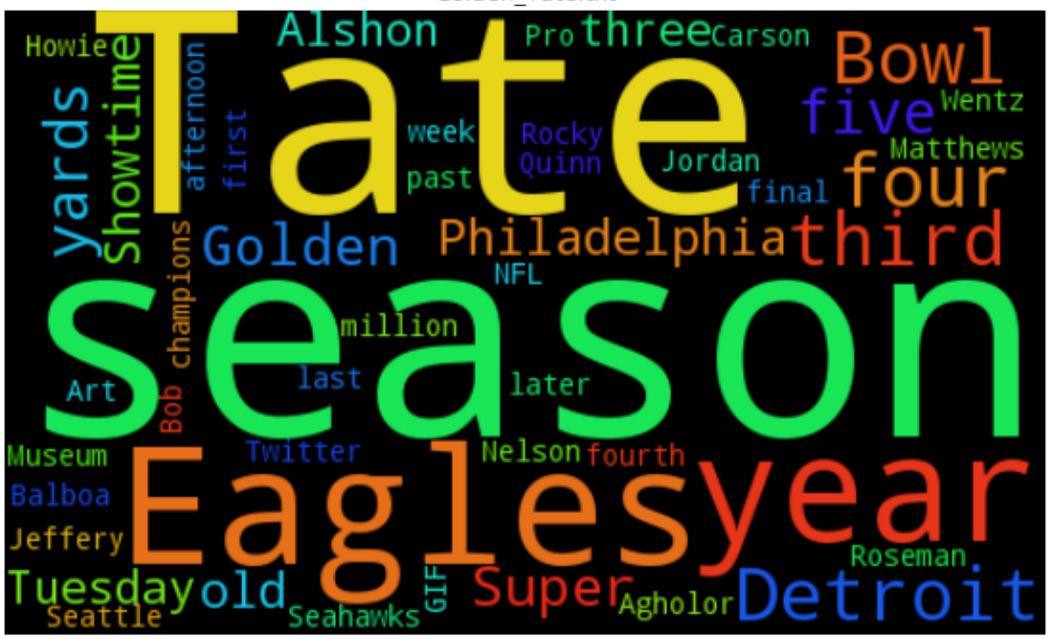
Data_Science_Questions.txt



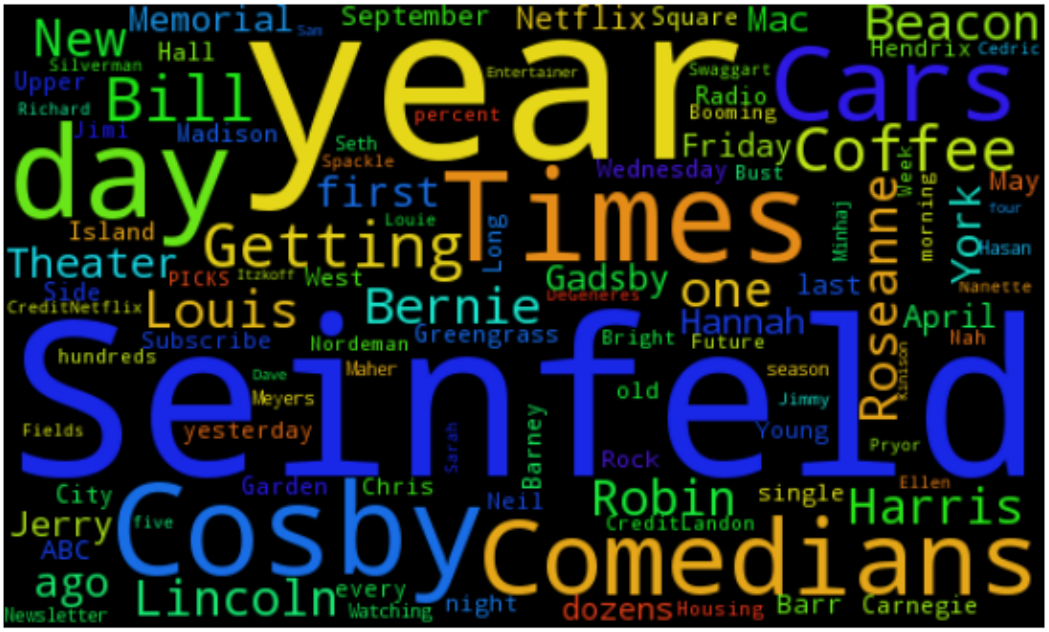
Climate_Change.txt



Golden_Tate.txt

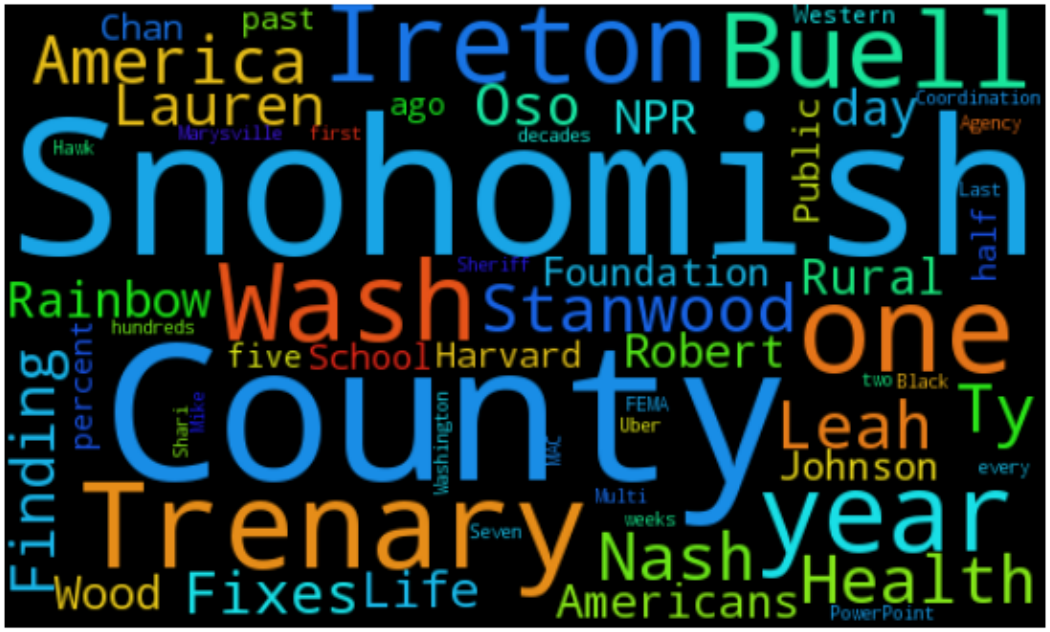


Jerry_Seinfeld.txt

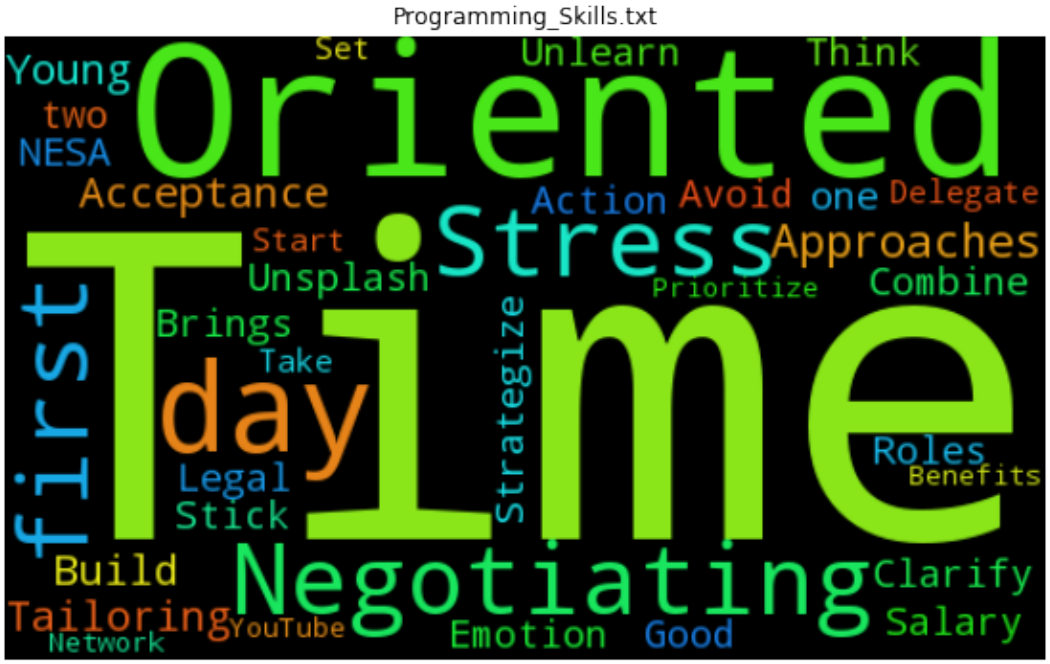


Walkability.txt

Opiates.txt



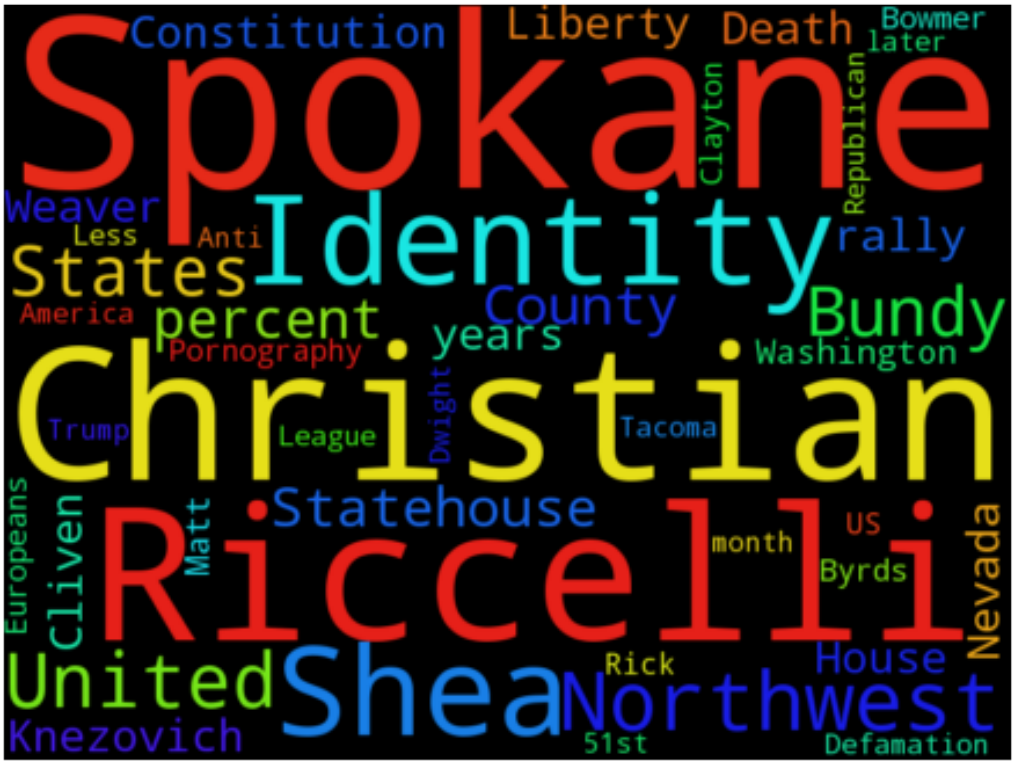
Facebook_Groups.txt



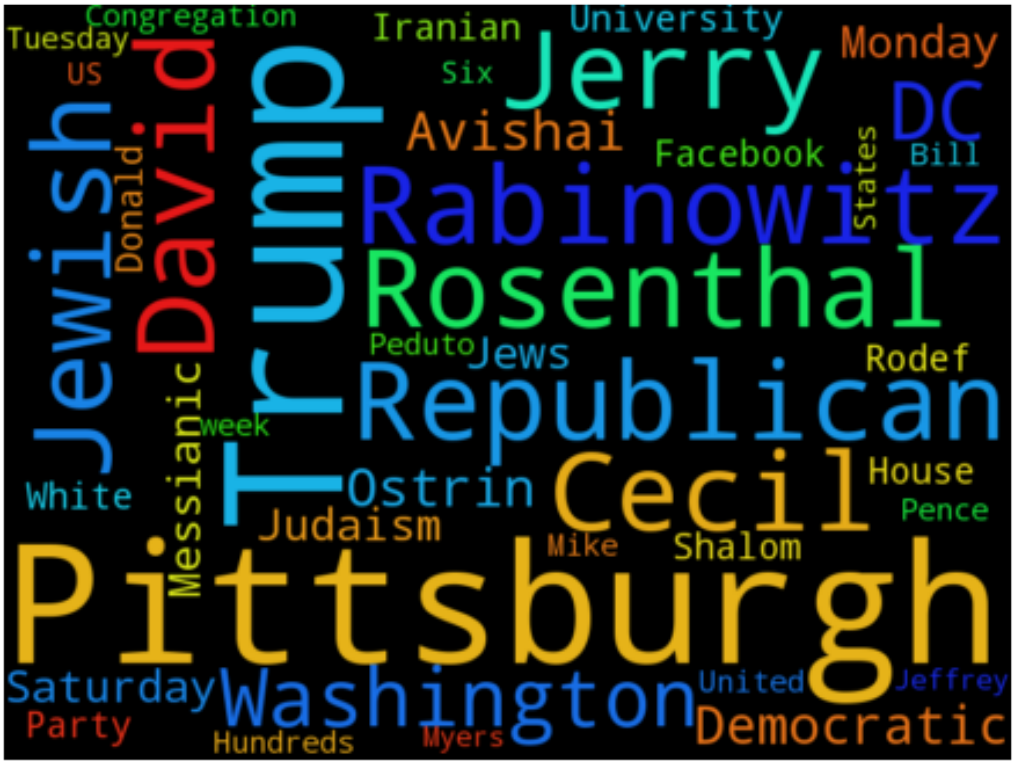
In this second chunk, I restricted the number of named entities in each word cloud to be 50. I did this by randomly sampling 50 named entities from the list of entities for each piece. This helps a little bit in focusing the word clouds for the longer articles. But, the focus of the more abstract article - like the one on "walkability" in cities - is still pretty unclear. I think this is because summaries for these types of pieces depend on much more than named entities to make sense of the information in the article.

```
In [41]: # build a word cloud with only 50 named entities
         for article in articles:
             text = open('/Users/brodyvogel/Desktop/580 Spacy Docs/' + article).read()
             doc = nlp(text)
             # here is where I randomly sample 50 named entities from all in the document
             ents = [str(x) for x in doc.ents]
                 # random sample
             sampled_ents = random.sample(ents, 50)
             for_word_cloud = ' '.join(sampled_ents)
             # build the Word Cloud
             wordcloud= WordCloud(width = 500, height = 300).generate(for_word_cloud)
             # show the Word Cloud
             plt.figure(figsize = (10, 6))
             plt.title("sampled " + article)
             plt.imshow(wordcloud, interpolation = "bilinear")
             plt.axis("off")
             plt.show()
```

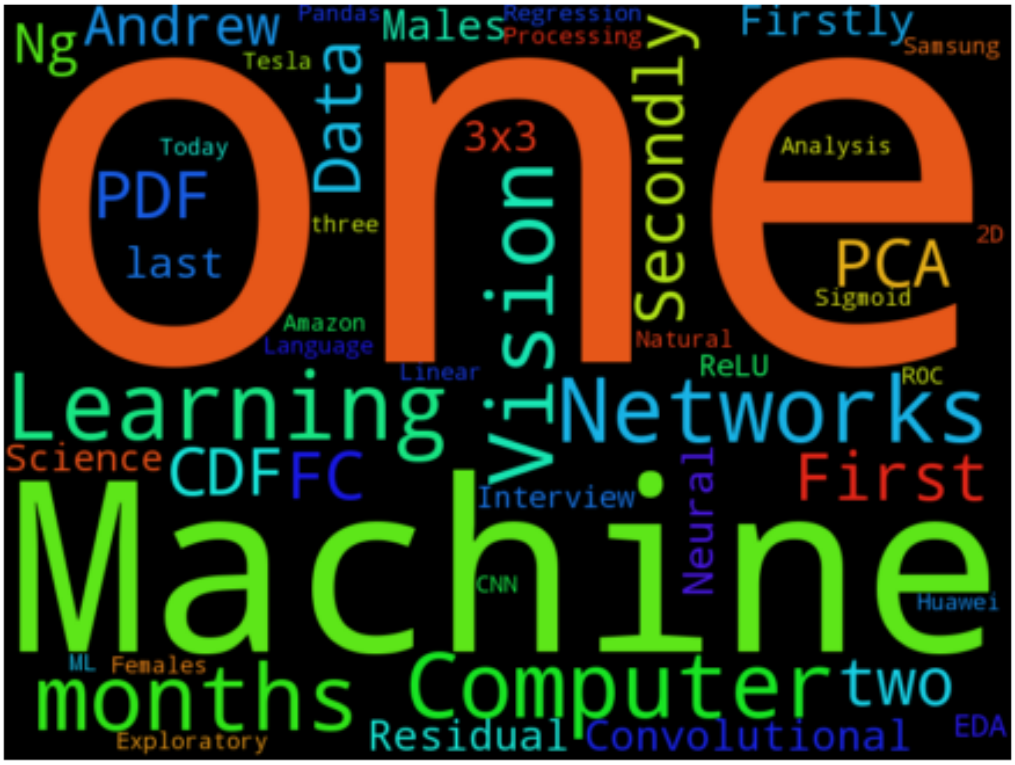
sampled Conservative_Politics.txt



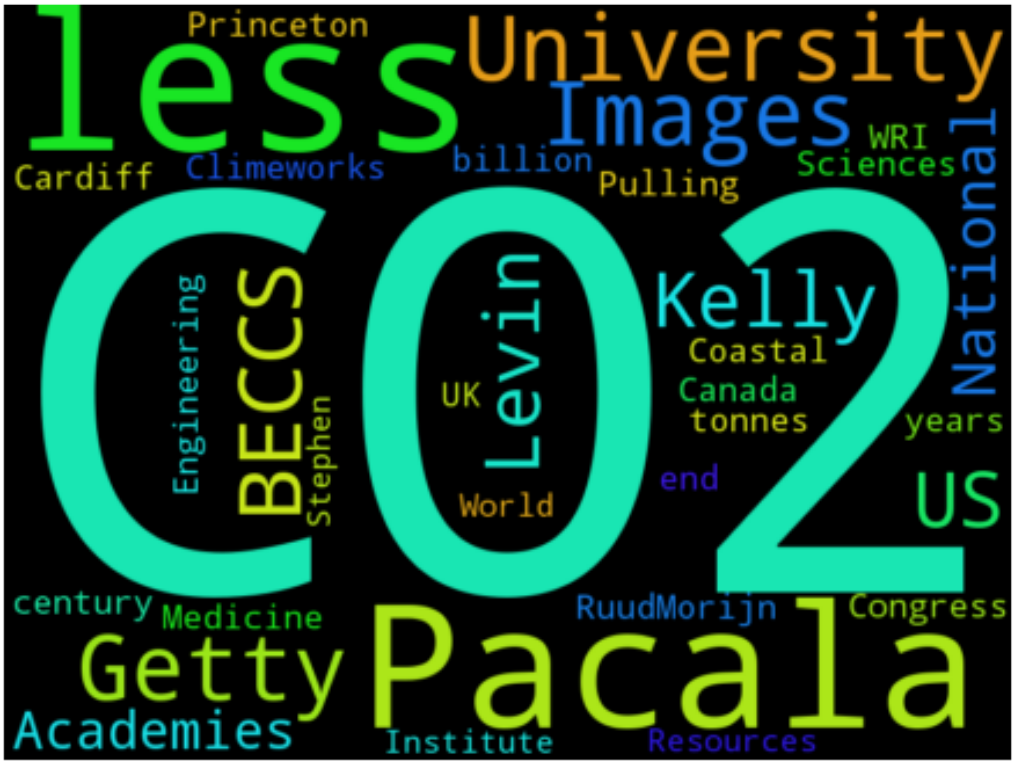
sampled Pittsburgh_Shooting.txt



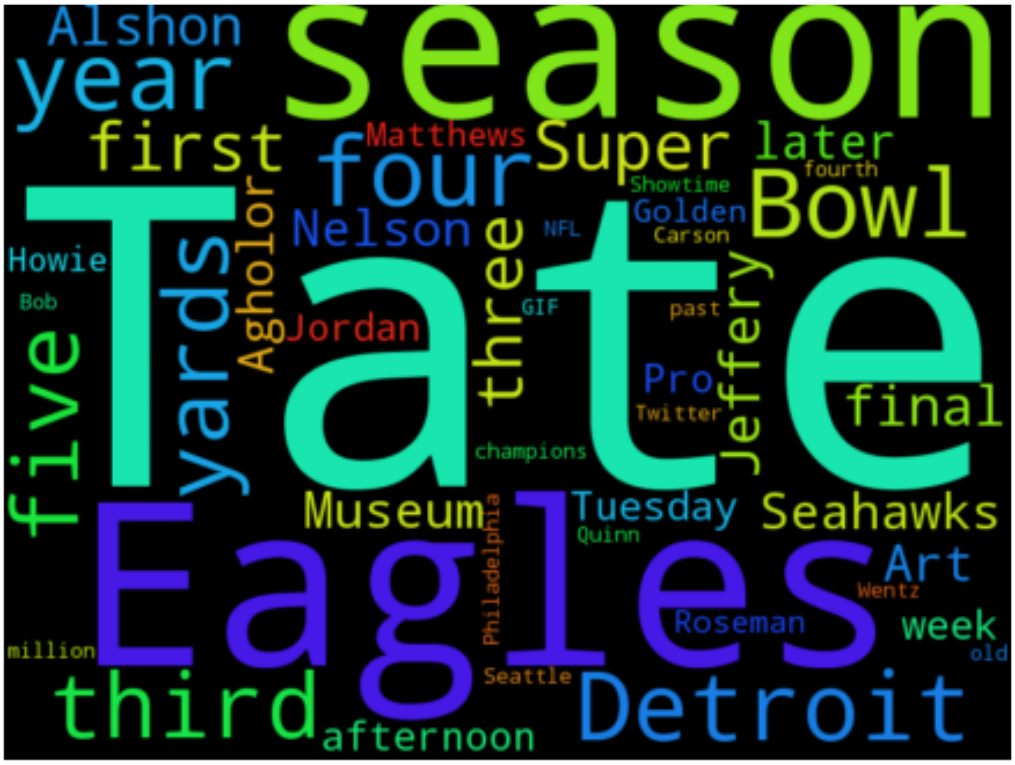
sampled Data_Science_Questions.txt



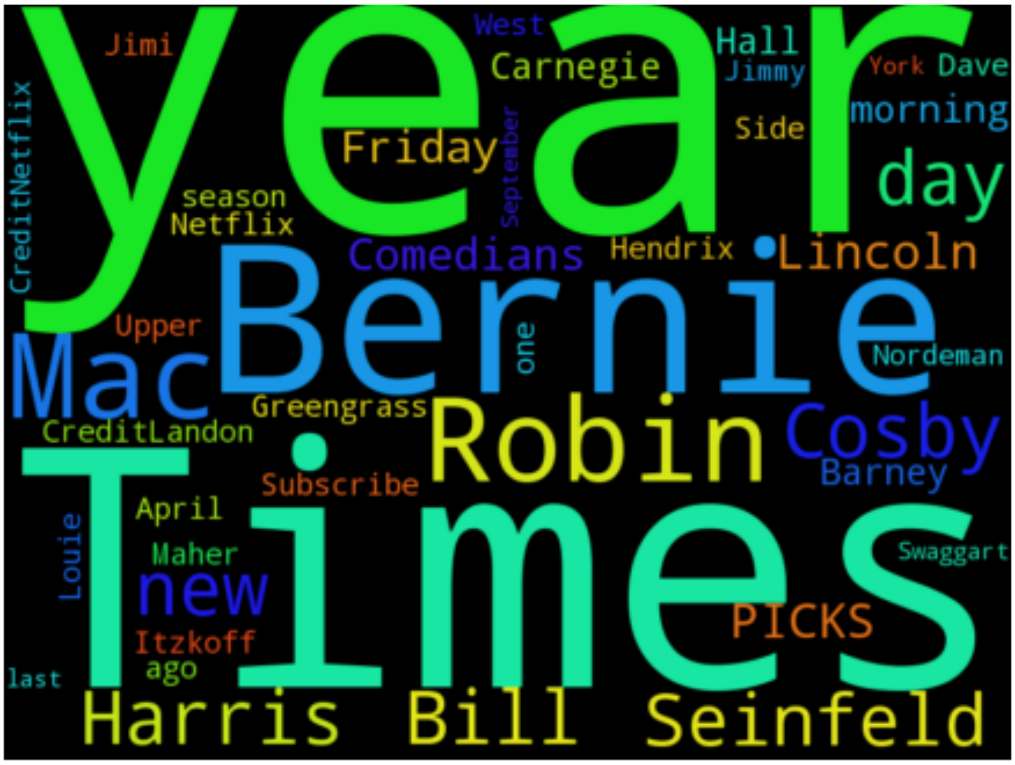
sampled Climate_Change.txt



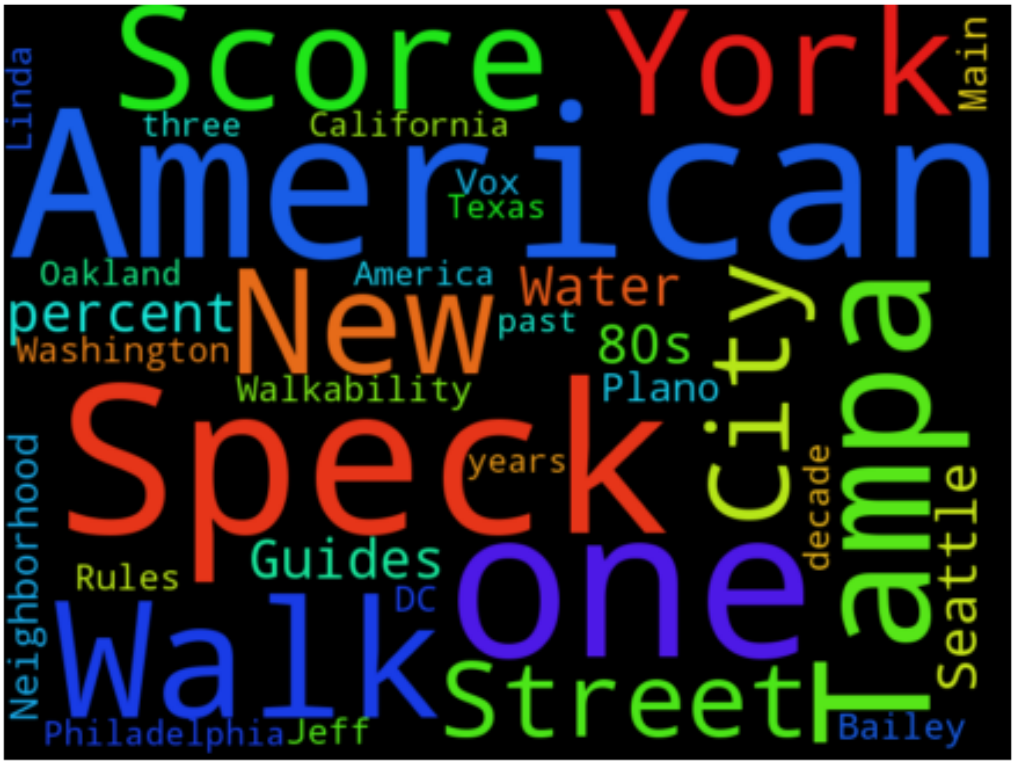
sampled Golden_Tate.txt



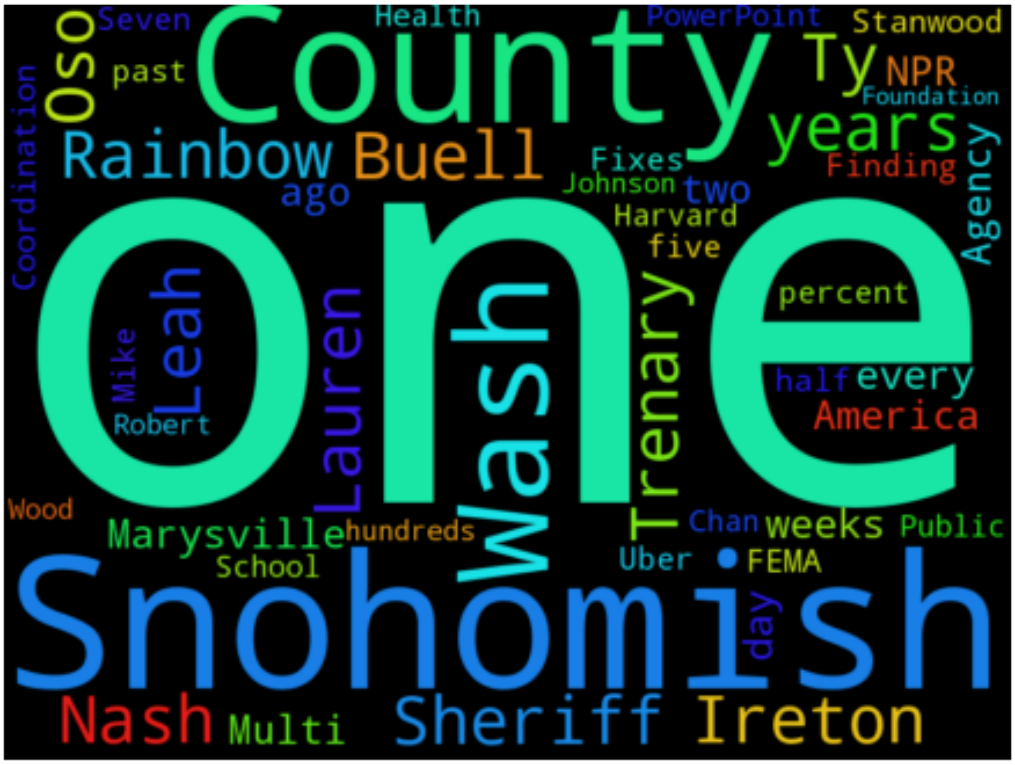
sampled Jerry_Seinfeld.txt



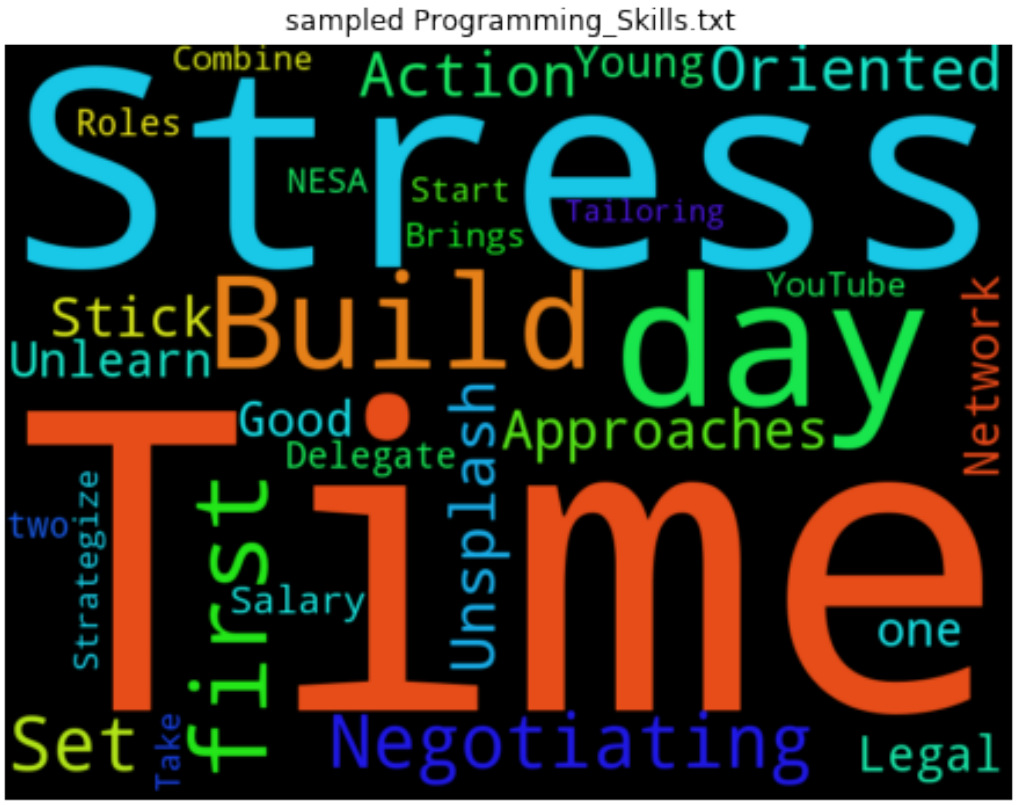
sampled Walkability.txt



sampled Opiates.txt



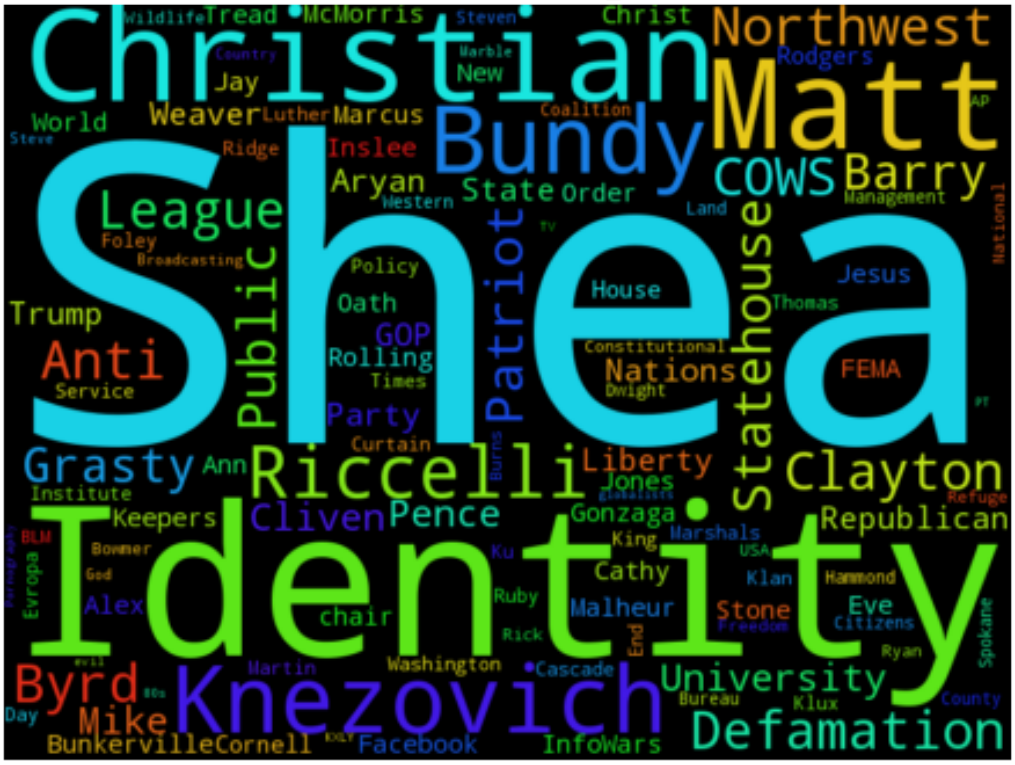
sampled Facebook_Groups.txt



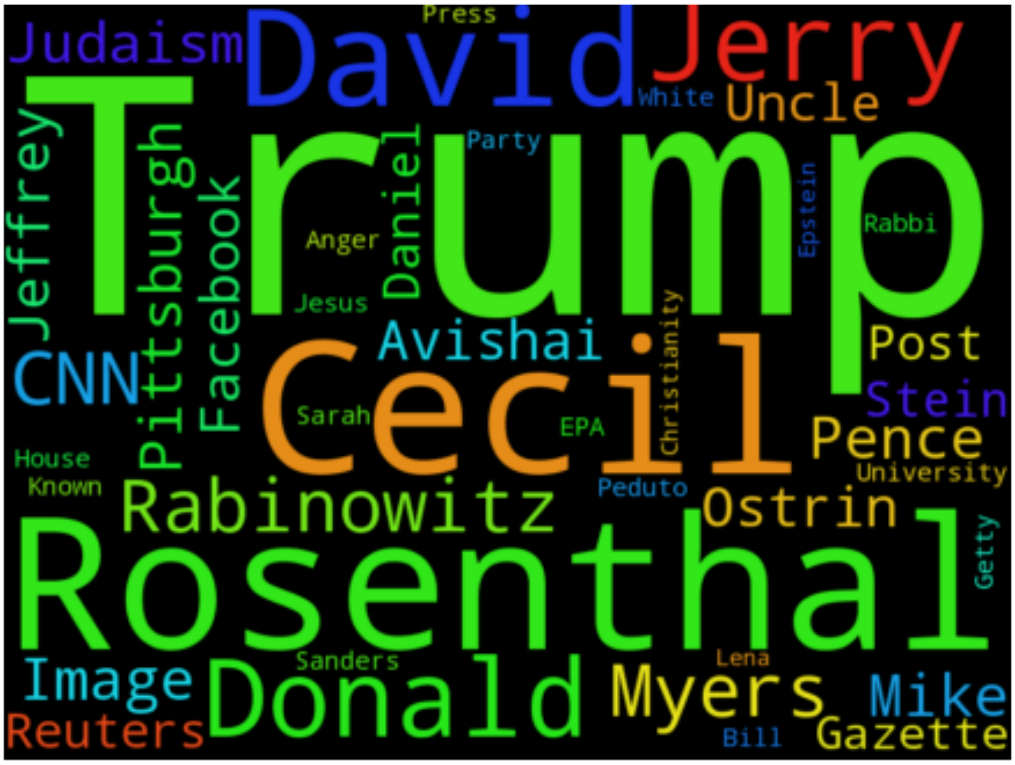
For this last chunk, then, I constrained the named entities in the word clouds to only be those that the algorithm labeled a person ("PER") or organization ("ORG"). This didn't make much difference for the more straightforward news articles (like the Golden Tate trade and the Pittsburgh shooting), but the word clouds for the abstract articles lose a lot of important details.

```
In [43]: # build a word cloud with only "PERSON" and "ORG" labelled entities
         for article in articles:
             text = open('/Users/brodyvogel/Desktop/580 Spacy Docs/' + article).read()
             doc = nlp(text)
             # get the named entities that were labeled "PERSON" or "ORG"
             constrained_ents = [str(x) for x in doc.ents if x.label_ in ['PERSON', 'ORG']]
             for_word_cloud = ' '.join(constrained_ents)
             # build the Word Cloud
             wordcloud= WordCloud(width = 500, height = 300).generate(for_word_cloud)
             # show the Word Cloud
             plt.figure(figsize = (10, 6))
             plt.title("constrained " + article)
             plt.imshow(wordcloud, interpolation = "bilinear")
             plt.axis("off")
             plt.show()
```

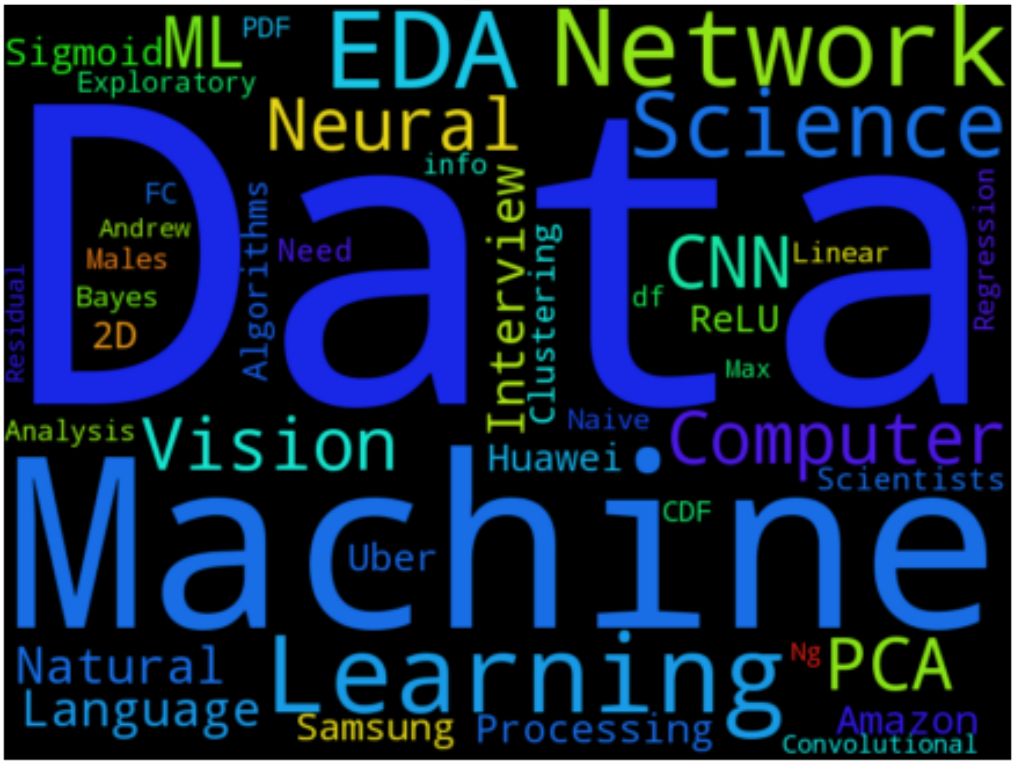
constrained Conservative_Politics.txt



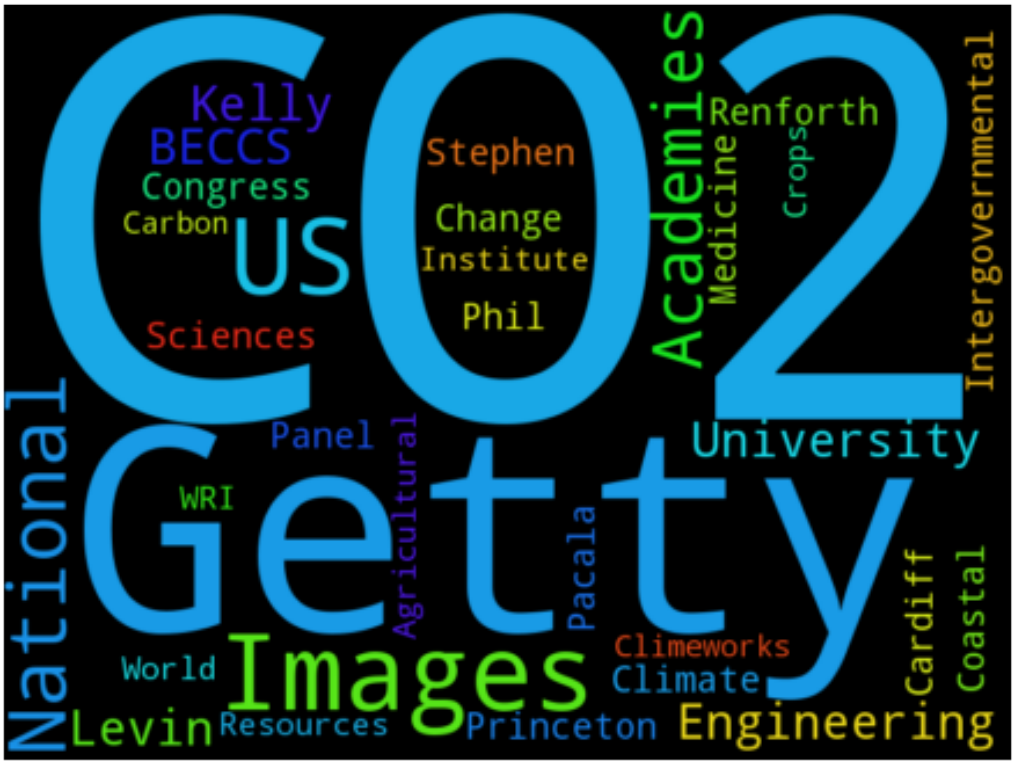
constrained Pittsburgh_Shooting.txt



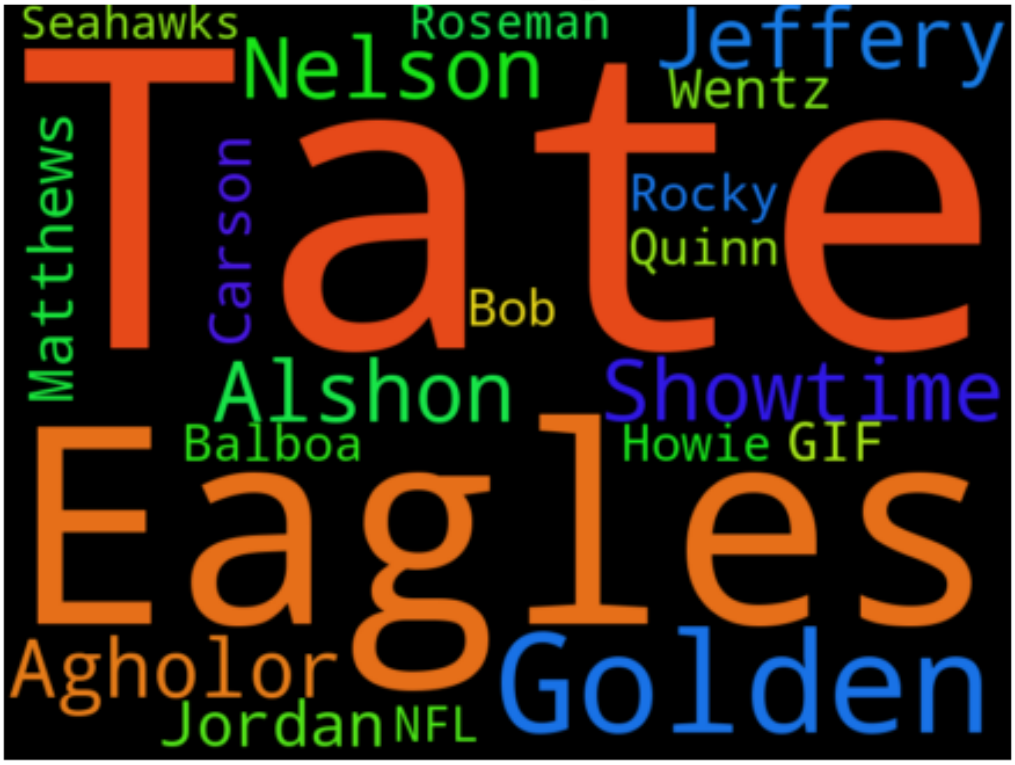
constrained Data_Science_Questions.txt



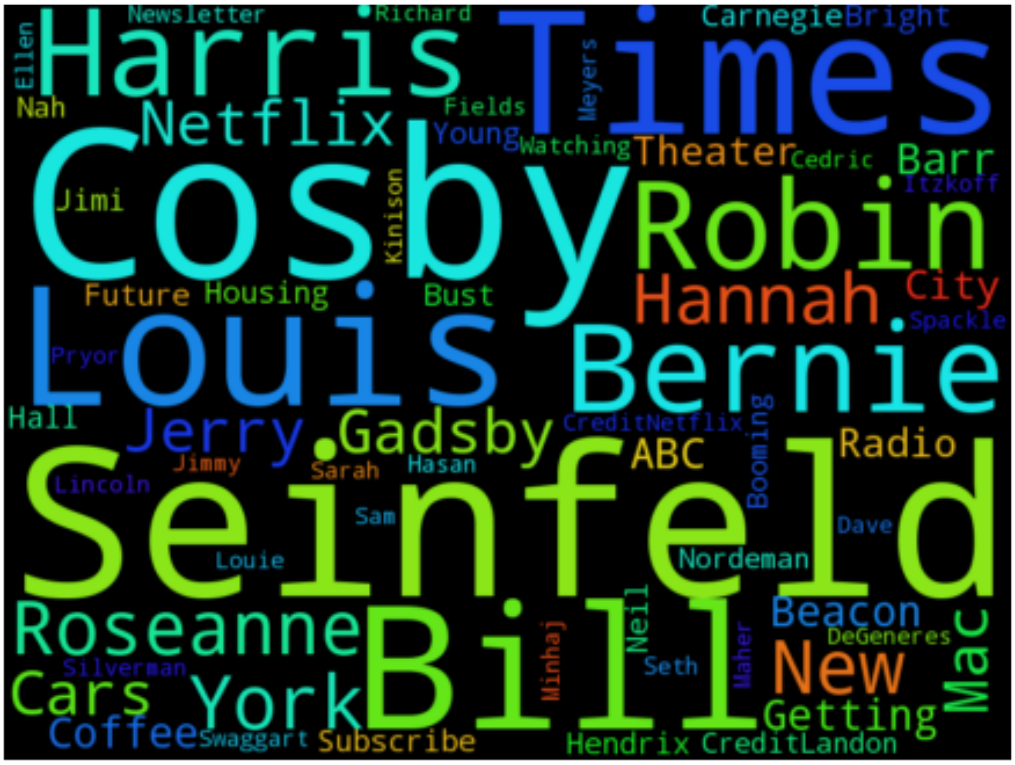
constrained Climate_Change.txt



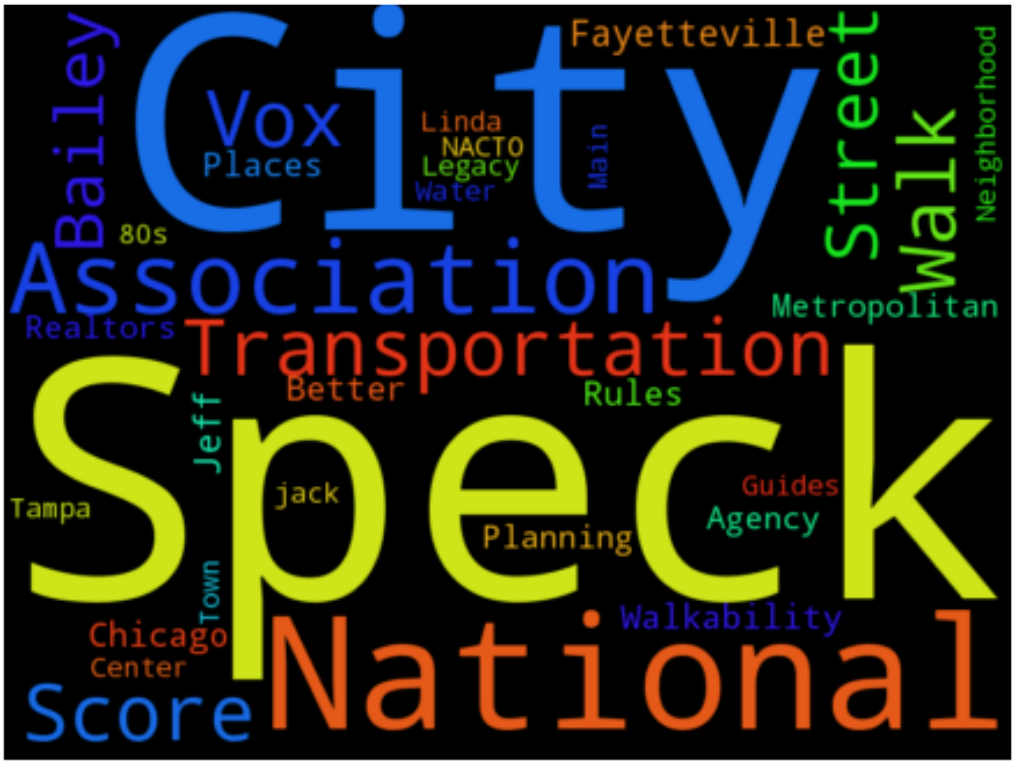
constrained Golden_Tate.txt



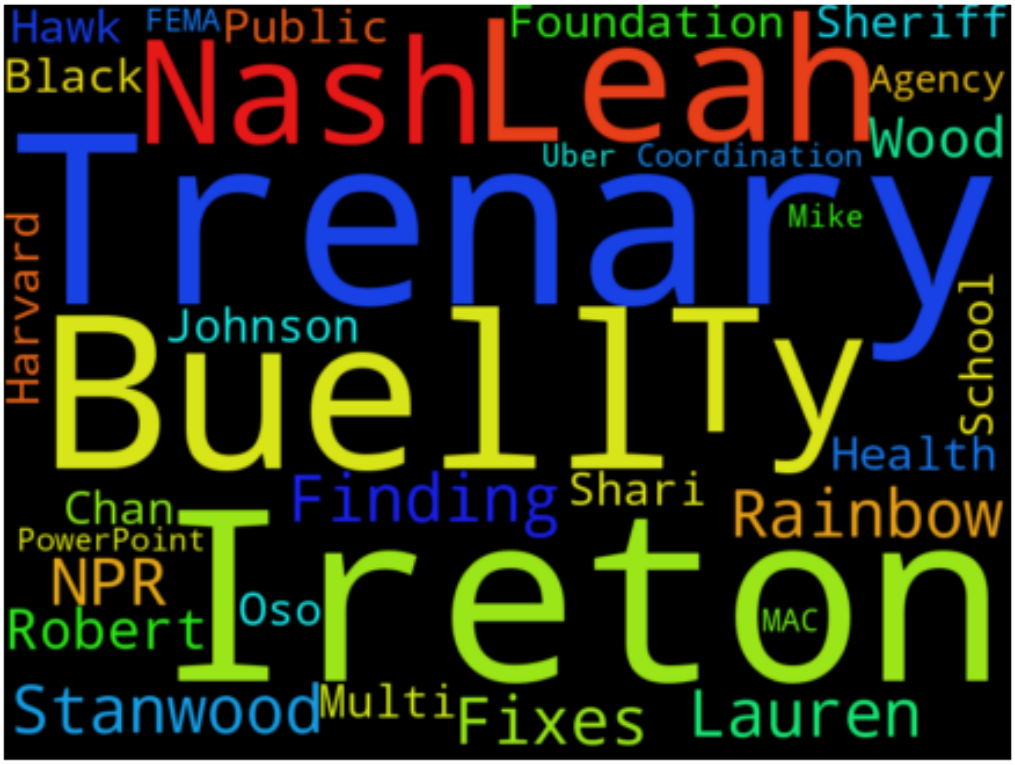
constrained Jerry_Seinfeld.txt



constrained Walkability.txt



constrained Opiates.txt



constrained Facebook_Groups.txt

constrained Programming_Skills.txt