Assignment 10

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 words in the article

In [82]:
```
!pip3 install spacy
```

```
Requirement already satisfied: spacy in /Library/Frameworks/Python.framework/Ver
sions/3.6/lib/python3.6/site-packages
Requirement already satisfied: numpy>=1.7 in /Library/Frameworks/Python.framewor
k/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: murmurhash<0.27,>=0.26 in /Library/Frameworks/Pyt
hon.framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: cymem<1.32,>=1.30 in /Library/Frameworks/Python.f
ramework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: preshed<2.0.0,>=1.0.0 in /Library/Frameworks/Pyth
on.framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: thinc<6.6.0,>=6.5.0 in /Library/Frameworks/Python
.framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: plac<1.0.0,>=0.9.6 in /Library/Frameworks/Python.
framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: six in /Library/Frameworks/Python.framework/Versi
ons/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: pathlib in /Library/Frameworks/Python.framework/V
ersions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: ujson>=1.35 in /Library/Frameworks/Python.framewo
rk/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: dill<0.3,>=0.2 in /Library/Frameworks/Python.fram
ework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /Library/Frameworks/Py
thon.framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: regex==2017.4.5 in /Library/Frameworks/Python.fra
mework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: ftfy<5.0.0,>=4.4.2 in /Library/Frameworks/Python.
framework/Versions/3.6/lib/python3.6/site-packages (from spacy)
Requirement already satisfied: wrapt in /Library/Frameworks/Python.framework/Ver
sions/3.6/lib/python3.6/site-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already satisfied: tqdm<5.0.0,>=4.10.0 in /Library/Frameworks/Python
.framework/Versions/3.6/lib/python3.6/site-packages (from thinc<6.6.0,>=6.5.0->s
pacy)
Requirement already satisfied: cytoolz<0.9,>=0.8 in /Library/Frameworks/Python.f
ramework/Versions/3.6/lib/python3.6/site-packages (from thinc<6.6.0,>=6.5.0->spa
cy)
Requirement already satisfied: termcolor in /Library/Frameworks/Python.framework
/Versions/3.6/lib/python3.6/site-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.
framework/Versions/3.6/lib/python3.6/site-packages (from requests<3.0.0,>=2.13.0
->spacy)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /Library/Frameworks/Pyth
on.framework/Versions/3.6/lib/python3.6/site-packages (from requests<3.0.0,>=2.1
3.0->spacy)
Requirement already satisfied: idna<2.6,>=2.5 in /Library/Frameworks/Python.fram
ework/Versions/3.6/lib/python3.6/site-packages (from requests<3.0.0,>=2.13.0->sp
acy)
Requirement already satisfied: urllib3<1.22,>=1.21.1 in /Library/Frameworks/Pyth
on.framework/Versions/3.6/lib/python3.6/site-packages (from requests<3.0.0,>=2.1
3.0->spacy)
Requirement already satisfied: html5lib in /Library/Frameworks/Python.framework/
Versions/3.6/lib/python3.6/site-packages (from ftfy<5.0.0,>=4.4.2->spacy)
Requirement already satisfied: wcwidth in /Library/Frameworks/Python.framework/V
ersions/3.6/lib/python3.6/site-packages (from ftfy<5.0.0,>=4.4.2->spacy)
Requirement already satisfied: toolz>=0.8.0 in /Library/Frameworks/Python.framew
ork/Versions/3.6/lib/python3.6/site-packages (from cytoolz<0.9,>=0.8->thinc<6.6.
0,>=6.5.0->spacy)
Requirement already satisfied: webencodings in /Library/Frameworks/Python.framew
ork/Versions/3.6/lib/python3.6/site-packages (from html5lib->ftfy<5.0.0,>=4.4.2-
>spacy)
Requirement already satisfied: setuptools>=18.5 in /Library/Frameworks/Python.fr
amework/Versions/3.6/lib/python3.6/site-packages (from html5lib->ftfy<5.0.0,>=4.
```

In [83]:
```
!pip install --upgrade spacy
```

```
Requirement already up-to-date: spacy in /Users/jMac/anaconda/lib/python3.6/site
-packages
Requirement already up-to-date: six in /Users/jMac/anaconda/lib/python3.6/site-p
ackages (from spacy)
Requirement already up-to-date: regex<2017.12.1,>=2017.4.1 in /Users/jMac/anacon
da/lib/python3.6/site-packages (from spacy)
Requirement already up-to-date: preshed<2.0.0,>=1.0.0 in /Users/jMac/anaconda/li
b/python3.6/site-packages (from spacy)
Requirement already up-to-date: murmurhash<0.27,>=0.26 in /Users/jMac/anaconda/l
ib/python3.6/site-packages (from spacy)
Requirement already up-to-date: ftfy<5.0.0,>=4.4.2 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from spacy)
Requirement already up-to-date: dill<0.3,>=0.2 in /Users/jMac/anaconda/lib/pytho
n3.6/site-packages (from spacy)
Requirement already up-to-date: plac<1.0.0,>=0.9.6 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from spacy)
Requirement already up-to-date: pip<10.0.0,>=9.0.0 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from spacy)
Requirement already up-to-date: thinc<6.6.0,>=6.5.0 in /Users/jMac/anaconda/lib/
python3.6/site-packages (from spacy)
Requirement already up-to-date: numpy>=1.7 in /Users/jMac/anaconda/lib/python3.6
/site-packages (from spacy)
Requirement already up-to-date: cymem<1.32,>=1.30 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from spacy)
Requirement already up-to-date: requests<3.0.0,>=2.13.0 in /Users/jMac/anaconda/
lib/python3.6/site-packages (from spacy)
Requirement already up-to-date: pathlib in /Users/jMac/anaconda/lib/python3.6/si
te-packages (from spacy)
Requirement already up-to-date: ujson>=1.35 in /Users/jMac/anaconda/lib/python3.
6/site-packages (from spacy)
Requirement already up-to-date: html5lib in /Users/jMac/anaconda/lib/python3.6/s
ite-packages (from ftfy<5.0.0,>=4.4.2->spacy)
Requirement already up-to-date: wcwidth in /Users/jMac/anaconda/lib/python3.6/si
te-packages (from ftfy<5.0.0,>=4.4.2->spacy)
Requirement already up-to-date: wrapt in /Users/jMac/anaconda/lib/python3.6/site
-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already up-to-date: tqdm<5.0.0,>=4.10.0 in /Users/jMac/anaconda/lib/
python3.6/site-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already up-to-date: cytoolz<0.9,>=0.8 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already up-to-date: termcolor in /Users/jMac/anaconda/lib/python3.6/
site-packages (from thinc<6.6.0,>=6.5.0->spacy)
Requirement already up-to-date: idna<2.6,>=2.5 in /Users/jMac/anaconda/lib/pytho
n3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already up-to-date: urllib3<1.22,>=1.21.1 in /Users/jMac/anaconda/li
b/python3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already up-to-date: chardet<3.1.0,>=3.0.2 in /Users/jMac/anaconda/li
b/python3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already up-to-date: certifi>=2017.4.17 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already up-to-date: webencodings in /Users/jMac/anaconda/lib/python3
.6/site-packages (from html5lib->ftfy<5.0.0,>=4.4.2->spacy)
Requirement already up-to-date: setuptools>=18.5 in /Users/jMac/anaconda/lib/pyt
hon3.6/site-packages (from html5lib->ftfy<5.0.0,>=4.4.2->spacy)
```

In [84]:
```
!python3 -m spacy download en
```

```
    Downloading en_core_web_sm-1.2.0/en_core_web_sm-1.2.0.tar.gz

Collecting https://github.com/explosion/spacy-models/releases/download/en_core_w
eb_sm-1.2.0/en_core_web_sm-1.2.0.tar.gz
  Downloading https://github.com/explosion/spacy-models/releases/download/en_cor
e_web_sm-1.2.0/en_core_web_sm-1.2.0.tar.gz (52.2MB)
    100% |████████████████████████████████| 52.2MB 2.5MB/s ta 0:00:0111
  Requirement already satisfied (use --upgrade to upgrade): en-core-web-sm==1.2.
0 from https://github.com/explosion/spacy-models/releases/download/en_core_web_s
m-1.2.0/en_core_web_sm-1.2.0.tar.gz in /Users/jMac/anaconda/lib/python3.6/site-p
ackages
Requirement already satisfied: spacy<2.0.0,>=1.7.0 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from en-core-web-sm==1.2.0)
Requirement already satisfied: numpy>=1.7 in /Users/jMac/anaconda/lib/python3.6/
site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: cymem<1.32,>=1.30 in /Users/jMac/anaconda/lib/pyt
hon3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: ftfy<5.0.0,>=4.4.2 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: pathlib in /Users/jMac/anaconda/lib/python3.6/sit
e-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: ujson>=1.35 in /Users/jMac/anaconda/lib/python3.6
/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: murmurhash<0.27,>=0.26 in /Users/jMac/anaconda/li
b/python3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: pip<10.0.0,>=9.0.0 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: regex<2017.12.1,>=2017.4.1 in /Users/jMac/anacond
a/lib/python3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: plac<1.0.0,>=0.9.6 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: dill<0.3,>=0.2 in /Users/jMac/anaconda/lib/python
3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: six in /Users/jMac/anaconda/lib/python3.6/site-pa
ckages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /Users/jMac/anaconda/l
ib/python3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: thinc<6.6.0,>=6.5.0 in /Users/jMac/anaconda/lib/p
ython3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: preshed<2.0.0,>=1.0.0 in /Users/jMac/anaconda/lib
/python3.6/site-packages (from spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: wcwidth in /Users/jMac/anaconda/lib/python3.6/sit
e-packages (from ftfy<5.0.0,>=4.4.2->spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0)
Requirement already satisfied: html5lib in /Users/jMac/anaconda/lib/python3.6/si
te-packages (from ftfy<5.0.0,>=4.4.2->spacy<2.0.0,>=1.7.0->en-core-web-sm==1.2.0
)
Requirement already satisfied: idna<2.6,>=2.5 in /Users/jMac/anaconda/lib/python
3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy<2.0.0,>=1.7.0->en-core-we
b-sm==1.2.0)
Requirement already satisfied: certifi>=2017.4.17 in /Users/jMac/anaconda/lib/py
thon3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy<2.0.0,>=1.7.0->en-cor
e-web-sm==1.2.0)
Requirement already satisfied: urllib3<1.22,>=1.21.1 in /Users/jMac/anaconda/lib
/python3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy<2.0.0,>=1.7.0->en-
core-web-sm==1.2.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /Users/jMac/anaconda/lib
/python3.6/site-packages (from requests<3.0.0,>=2.13.0->spacy<2.0.0,>=1.7.0->en-
core-web-sm==1.2.0)
Requirement already satisfied: cytoolz<0.9,>=0.8 in /Users/jMac/anaconda/lib/pyt
hon3.6/site-packages (from thinc<6.6.0,>=6.5.0->spacy<2.0.0,>=1.7.0->en-core-web
-sm==1.2.0)
Requirement already satisfied: termcolor in /Users/jMac/anaconda/lib/python3.6/s
```

In [85]:
```
import spacy
```

In [86]:
```
processor = spacy.load('en')
```

After downloading the model, we can use some text from an article found on Forbes.com (https://www.forbes.com/sites/rogergroves/2017/07/20/what-celebrity-athletes-can-learn-from-the-o-j-simpson-parole-hearing/2/#5d9647978e18). This is an article on, from a sports law standpoint, what celebirty athletes can learn from OJ Simpson's parole hearing.

In [87]:
```
text2 = """Probably no other celebrity athlete will ever be in the same exact sit
uation as O.J. Simpson — 70 years old asking for parole after a conviction for ro
bbery, kidnapping, and conspiracy, some 23 years after one of the most sensationa
lized murder acquittals in American history.

But well over half of NBA and NFL players file for bankruptcy protection within 5
years of retirement. Many will have diagnosed or undiagnosed brain injuries that
prevent or hamper employment, and increase depression, mood swings, and bad judge
ment. And bad judgment is what Simpson admitted as a cause for his botched attemp
t to reclaim intimate family photos and other memorabilia. Many players may face
a near-perfect storm in a similar setting — a desperate need for money and a misg
uided attempt to retrieve it, especially since the property involves his own name
, image and likeness.

Some former athletes have also lived in a cocoon of celebrity, fighting the demon
s of being self-absorbed. That can lead to a failure to be humble, to understand
the role of contrition and remorse. Those character attributes are important when
begging a judge, jury, or a parole board for forgiveness. We know crimes by forme
r athletes defrocked from fame occur in greater number than we would like. So a v
ersion of these circumstances and lessons therefrom is not far-fetched. If convic
ted, instead of saying, “I’m not the bad guy, somebody else was,” the more strate
gic verse is “Yes, I was the bad guy, but I am not any more because…”

If athletes are paying careful attention to Simpson’s own statements, a celeb sin
ce he was 19, they should notice that he essentially attempted to be his own coun
sel and re-litigate the trial he lost. Instead, athletes should follow the advice
of counsel to be humble, contrite, and remorseful. That is a Simpson mistake no s
imilarly situated player should replicate.

For example, Simpson’s underlying theme throughout the hearing was that his “secu
rity” team held the guns, and perpetrated the violence, not him. Simpson repeated
ly admitted the error of bad judgment in bringing thugs with him. But Simpson ris
ked a parole board perception that he failed to take responsibility for his orche
stration of the events that led to his conviction. There was a conspiracy count,
so he did not have to hold a gun to be guilty. Additionally, Simpson did not test
ify during the robbery trial. The jury heard other testimony not so favorable. Si
mpson and players facing a parole hearing should remember the Board is looking at
his character, and admission of the full extent of his errors. How else can one a
ssess whether he is a risk for repeating his errors unless they appreciate the gr
avity and full extent of those errors.

Simpson was successful at the parole hearing. But other former stars should not b
e confused. Simpson had the advantage of (1) a sentence that was extraordinarily
long (9.5 to 33 years for what CNN commentator Mark Geragos said is typically a o
ne-year sentence), (2) both victims apparently being at peace with parole, (3) an
d incident-free actions during incarceration. Future players are not likely to ha
ve all those advantages.

Instead, athletes should focus on the strategy of humility, contrition, and remor
se. More importantly, they should avoid appearing delusional about their own sord
id history if one exists. Even ignoring the murder trial, Simpson still had the c
onviction for domestic violence, after multiple 911 calls from his then-wife. Yet
Simpson’s opening statement included his self-assessment that he basically led a
“conflict-free” life. He probably was thinking about the adoration of stardom he
received prior to the murder trial. The delusion is that repeated violence agains
t a woman is not a linear equation in the character analysis. It is a crime that
colors the entire perception of the person, regardless of the year perpetrated.

As one board member cautioned, Simpson will receive strict scrutiny on yet-to-be-
finalized conditions to his parole. I expect those conditions to have muscularity
Simpson never faced on the field. He talked of being conflict-free. Starting in O
```

In [88]:
```
processed_text2 = processor(text2)
```

In [89]:
```
processed_text2
```

Out[89]: Probably no other celebrity athlete will ever be in the same exact situation as O.J. Simpson — 70 years old asking for parole after a conviction for robbery, kidnapping, and conspiracy, some 23 years after one of the most sensationalized murder acquittals in American history.

But well over half of NBA and NFL players file for bankruptcy protection within 5 years of retirement. Many will have diagnosed or undiagnosed brain injuries that prevent or hamper employment, and increase depression, mood swings, and bad judgement. And bad judgment is what Simpson admitted as a cause for his botched attempt to reclaim intimate family photos and other memorabilia. Many players may face a near-perfect storm in a similar setting — a desperate need for money and a misguided attempt to retrieve it, especially since the property involves his own name, image and likeness.

Some former athletes have also lived in a cocoon of celebrity, fighting the demons of being self-absorbed. That can lead to a failure to be humble, to understand the role of contrition and remorse. Those character attributes are important when begging a judge, jury, or a parole board for forgiveness. We know crimes by former athletes defrocked from fame occur in greater number than we would like. So a version of these circumstances and lessons therefrom is not far-fetched. If convicted, instead of saying, “I’m not the bad guy, somebody else was,” the more strategic verse is “Yes, I was the bad guy, but I am not any more because…”

If athletes are paying careful attention to Simpson’s own statements, a celeb since he was 19, they should notice that he essentially attempted to be his own counsel and re-litigate the trial he lost. Instead, athletes should follow the advice of counsel to be humble, contrite, and remorseful. That is a Simpson mistake no similarly situated player should replicate.

For example, Simpson’s underlying theme throughout the hearing was that his “security” team held the guns, and perpetrated the violence, not him. Simpson repeatedly admitted the error of bad judgment in bringing thugs with him. But Simpson risked a parole board perception that he failed to take responsibility for his orchestration of the events that led to his conviction. There was a conspiracy count, so he did not have to hold a gun to be guilty. Additionally, Simpson did not testify during the robbery trial. The jury heard other testimony not so favorable. Simpson and players facing a parole hearing should remember the Board is looking at his character, and admission of the full extent of his errors. How else can one assess whether he is a risk for repeating his errors unless they appreciate the gravity and full extent of those errors.

Simpson was successful at the parole hearing. But other former stars should not be confused. Simpson had the advantage of (1) a sentence that was extraordinarily long (9.5 to 33 years for what CNN commentator Mark Geragos said is typically a one-year sentence), (2) both victims apparently being at peace with parole, (3) and incident-free actions during incarceration. Future players are not likely to have all those advantages.

Instead, athletes should focus on the strategy of humility, contrition, and remorse. More importantly, they should avoid appearing delusional about their own sordid history if one exists. Even ignoring the murder trial, Simpson still had the conviction for domestic violence, after multiple 911 calls from his then-wife. Yet Simpson’s opening statement included his self-assessment that he basically led a “conflict-free” life. He probably was thinking about the adoration of stardom he received prior to the murder trial. The delusion is that repeated violence against a woman is not a linear equation in the character analysis. It is a crime that colors the entire perception of the person, regardless of the year perpetrated.

As one board member cautioned, Simpson will receive strict scrutiny on yet-to-be-finalized conditions to his parole. I expect those conditions to have muscularity Simpson never faced on the field. He talked of being conflict-free. Starting

1. Find the most common words in the article

In [90]:
```
from spacy import attrs
from spacy.attrs import ORTH
from spacy.tokens.doc import Doc
from collections import defaultdict, Counter
```

Note: Imported various modules to help with the assignment (many found from going through the tutorials on Spacy.io (https://spacy.io/docs/usage/resources).

In [91]:
```
#Store all the words that are not stop words or punctuations in a dictionary usin
g a for loop and if statement

words = [token.text for token in processed_text2 if token.is_stop != True and tok
en.is_punct != True]

#call on Counter module to determine word frequency

word_frequency = Counter(words)

#call and print the most common words, those top 30 words (count of the words)

common_words = word_frequency.most_common(30)
common_words
```

Out[91]:
```
[('Simpson', 16),
 ('parole', 8),
 ('\n\n', 8),
 ('athletes', 6),
 ('players', 5),
 ('bad', 5),
 ('board', 5),
 ('years', 4),
 ('conviction', 4),
 ('trial', 4),
 ('free', 4),
 ('murder', 3),
 ('character', 3),
 ('jury', 3),
 ('crimes', 3),
 ('’s', 3),
 ('hearing', 3),
 ('violence', 3),
 ('errors', 3),
 ('celebrity', 2),
 ('athlete', 2),
 ('robbery', 2),
 ('conspiracy', 2),
 ('history', 2),
 ('judgment', 2),
 ('admitted', 2),
 ('attempt', 2),
 ('need', 2),
 ('especially', 2),
 ('self', 2)]
```

Observation: It appears from the Counter, that Simpson is the most common word with 16 mentions. Would make sense as he's the subject of the article.

2. Most common words under a part of speech.

Note: For this portion, it's a similar approach, however we'll have to call on the parts of speech (nouns, verbs, etc.). Let's try this for a couple parts of speech, say adjectives and nouns?

In [92]:
```
#Store all the words under noun and adjective parts of speech that are not stop w
ords or punctuations in a dictionary using a for loop and if statement
#use token.pos

nouns = [token.text for token in processed_text2 if token.is_stop != True and tok
en.is_punct != True and token.pos_ =="NOUN"]
adjectives = [token.text for token in processed_text2 if token.is_stop != True an
d token.is_punct != True and token.pos_ =="ADJ"]
adverbs = [token.text for token in processed_text2 if token.is_stop != True and t
oken.is_punct != True and token.pos_ =="ADV"]

#call on Counter module to determine frequencies

noun_frequency = Counter(nouns)
adjective_frequency = Counter(adjectives)
adverb_frequency = Counter(adverbs)
```

In [93]:
```
#call and print the most common nouns, top 15 (count of the nouns)

common_nouns = noun_frequency.most_common(15)
common_nouns
```

Out[93]:
```
[('parole', 8),
 ('athletes', 6),
 ('players', 5),
 ('board', 5),
 ('years', 4),
 ('conviction', 4),
 ('trial', 4),
 ('murder', 3),
 ('character', 3),
 ('jury', 3),
 ('crimes', 3),
 ('hearing', 3),
 ('violence', 3),
 ('errors', 3),
 ('celebrity', 2)]
```

In [94]:
```
#call and print the most common adjectives, top 15 (count of the adjectives)

common_adjectives = adjective_frequency.most_common(15)
common_adjectives
```

Out[94]:
```
[('bad', 5),
 ('free', 4),
 ('humble', 2),
 ('exact', 1),
 ('old', 1),
 ('sensationalized', 1),
 ('American', 1),
 ('undiagnosed', 1),
 ('botched', 1),
 ('intimate', 1),
 ('perfect', 1),
 ('similar', 1),
 ('desperate', 1),
 ('misguided', 1),
 ('important', 1)]
```

In [95]:
```
#call and print the most common adverbs, top 15 (count of the adverbs)

common_adverbs = adverb_frequency.most_common(15)
common_adverbs
```

Out[95]:
```
[('especially', 2),
 ('Instead', 2),
 ('Probably', 1),
 ('near', 1),
 ('far', 1),
 ('instead', 1),
 ('essentially', 1),
 ('similarly', 1),
 ('repeatedly', 1),
 ('Additionally', 1),
 ('extraordinarily', 1),
 ('typically', 1),
 ('apparently', 1),
 ('importantly', 1),
 ('basically', 1)]
```

Observation: Both interesting results (particularly as I have not read the article in full). From the adjective and adverb frequency counts, it would appear that this is a "lessons learned" article, warning other athlets/players from parole, conviction, trial, crimes, errors (as seen in the noun frequencies).

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

Note: For this portion, let's use the pr_tree against our sentences. We can find the roots and the dependencies in our processed.text2 sentences.

In [96]:
```
#first, how many sentences are we looking at in our article's text?

sentences = [s for s in processed_text2.sents]
print(len(sentences)) #length of sentences
```

```
46
```

In [97]:
```
#print out the 46 sentences

n = 0
for sentence in processed_text2.sents:
    print(n, sentence)
    n+=1
```

0 Probably no other celebrity athlete will ever be in the same exact situation a
s O.J. Simpson — 70 years old asking for parole after a conviction for robbery,
kidnapping, and conspiracy, some 23 years after one of the most sensationalized
murder acquittals in American history.

1 But well over half of NBA and NFL players file for bankruptcy protection withi
n 5 years of retirement.
2 Many will have diagnosed or undiagnosed brain injuries that prevent or hamper
employment, and increase depression, mood swings, and bad judgement.
3 And bad judgment is what Simpson admitted as a cause for his botched attempt t
o reclaim intimate family photos and other memorabilia.
4 Many players may face a near-perfect storm in a similar setting — a desperate
need for money and a misguided attempt to retrieve it, especially since the prop
erty involves his own name, image and likeness.

5 Some former athletes have also lived in a cocoon of celebrity, fighting the de
mons of being self-absorbed.
6 That can lead to a failure to be humble, to understand the role of contrition
and remorse.
7 Those character attributes are important when begging a judge, jury, or a paro
le board for forgiveness.
8 We know crimes by former athletes defrocked from fame occur in greater number
than we would like.
9 So a version of these circumstances and lessons therefrom is not far-fetched.
10 If convicted, instead of saying, “
11 I’m not the bad guy, somebody else was,” the more strategic verse is “
12 Yes, I was the bad guy, but I am not any more because…”

13 If athletes are paying careful attention to Simpson’s own statements, a celeb
since he was 19, they should notice that he essentially attempted to be his own
counsel and re-litigate the trial he lost.
14 Instead, athletes should follow the advice of counsel to be humble, contrite,
and remorseful.
15 That is a Simpson mistake no similarly situated player should replicate.

16 For example, Simpson’s underlying theme throughout the hearing was that his “
security”
17 team held the guns, and perpetrated the violence, not him.
18 Simpson repeatedly admitted the error of bad judgment in bringing thugs with
him.
19 But Simpson risked a parole board perception that he failed to take responsib
ility for his orchestration of the events that led to his conviction.
20 There was a conspiracy count, so he did not have to hold a gun to be guilty.
21 Additionally, Simpson did not testify during the robbery trial.
22 The jury heard other testimony not so favorable.
23 Simpson and players facing a parole hearing should remember the Board is look
ing at his character, and admission of the full extent of his errors.
24 How else can one assess whether he is a risk for repeating his errors unless
they appreciate the gravity and full extent of those errors.

25 Simpson was successful at the parole hearing.
26 But other former stars should not be confused.
27 Simpson had the advantage of (1) a sentence that was extraordinarily long (9.
5 to 33 years for what CNN commentator Mark Geragos said is typically a one-year
sentence), (2) both victims apparently being at peace with parole, (3) and incid
ent-free actions during incarceration.
28 Future players are not likely to have all those advantages.

In [98]:
```
#explore the various Parts of Speech and words used in the article

n = 0
for sentence in processed_text2.sents:
    for token in sentence:
        print(n, token, token.pos_, token.lemma_)
        n+=1
```

```
0 Probably ADV probably
1 no DET no
2 other ADJ other
3 celebrity NOUN celebrity
4 athlete NOUN athlete
5 will VERB will
6 ever ADV ever
7 be VERB be
8 in ADP in
9 the DET the
10 same ADJ same
11 exact ADJ exact
12 situation NOUN situation
13 as ADP as
14 O.J. PROPN o.j.
15 Simpson PROPN simpson
16 — PART —
17 70 NUM 70
18 years NOUN year
19 old ADJ old
20 asking VERB ask
21 for ADP for
22 parole NOUN parole
23 after ADP after
24 a DET a
25 conviction NOUN conviction
26 for ADP for
27 robbery NOUN robbery
28 , PUNCT ,
29 kidnapping NOUN kidnapping
30 , PUNCT ,
31 and CCONJ and
32 conspiracy NOUN conspiracy
33 , PUNCT ,
34 some DET some
35 23 NUM 23
36 years NOUN year
37 after ADP after
38 one NUM one
39 of ADP of
40 the DET the
41 most ADV most
42 sensationalized ADJ sensationalized
43 murder NOUN murder
44 acquittals NOUN acquittal
45 in ADP in
46 American ADJ american
47 history NOUN history
48 . PUNCT .
49

 SPACE


50 But CCONJ but
51 well INTJ well
52 over ADP over
53 half NOUN half
54 of ADP of
55 NBA PROPN nba
56 and CCONJ and
57 NFL PROPN nfl
```

In [99]:
```
#let's do a dependency parsing tree for the entire article

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 [100]:
```
#print results of the parsing tree for the article

for sentence in processed_text2.sents:
    pr_tree(sentence.root, 0)
    print('====================Next Tree================')
```

```
        Probably - advmod
                no - det
                other - amod
                celebrity - compound
        athlete - nsubj
        will - aux
        ever - advmod
be - ROOT
        in - prep
                        the - det
                        same - amod
                        exact - amod
                situation - pobj
====================Next Tree=================
        But - cc
        well - nsubj
                over - prep
                        half - pobj
                                of - prep
                                                NBA - nmod
                                                        and - cc
                                                        NFL - conj
                                        players - pobj
file - ROOT
        for - prep
                        bankruptcy - compound
                protection - pobj
        within - prep
                        5 - nummod
                years - pobj
                        of - prep
                                retirement - pobj
====================Next Tree=================
        Many - nsubj
        will - aux
        have - aux
diagnosed - ROOT
        or - cc
                undiagnosed - amod
                brain - compound
        injuries - conj
                        that - nsubj
                prevent - relcl
                        or - cc
                        hamper - conj
                                employment - dobj
                                and - cc
                                        increase - compound
                                depression - conj
                                        mood - compound
                                swings - dobj
                                        and - cc
                                                bad - amod
                                        judgement - conj
====================Next Tree=================
        And - cc
                bad - amod
        judgment - nsubj
is - ROOT
                what - dobj
                Simpson - nsubj
        admitted - ccomp
```

Observation: This shows the various relationships throughout the article. The fourth tree, related to Simpson (the subject) is interesting-- showing the relationship of the words, "admitted, cause, botched attempt, etc." and probably refers to his own judgment or judgment made for his case.

Note: Let's take a closer look at this, just by taking out the first sentence and seeing the subject/object relationships.

In [101]:
```
subtext = """Probably no other celebrity athlete will ever be in the same exact s
ituation as O.J. Simpson — 70 years old asking for parole after a conviction for
robbery, kidnapping, and conspiracy, some 23 years after one of the most sensatio
nalized murder acquittals in American history."""
```

In [102]:
```
processed_subtext = processor(subtext)
processed_subtext
```

Out[102]:
```
Probably no other celebrity athlete will ever be in the same exact situation as
O.J. Simpson — 70 years old asking for parole after a conviction for robbery, ki
dnapping, and conspiracy, some 23 years after one of the most sensationalized mu
rder acquittals in American history.
```

In [103]:
```
for sentence in processed_subtext.sents:
    pr_tree(sentence.root, 0)
    print('====================End Tree=================')
```

```
        Probably - advmod
                no - det
                other - amod
                celebrity - compound
        athlete - nsubj
        will - aux
        ever - advmod
be - ROOT
        in - prep
                        the - det
                        same - amod
                        exact - amod
                situation - pobj
====================End Tree=================
```

Observation: It appears that there aren't many objects for the subject. It appears that the situation could apply to other "celebrity athletes".

4. Show the most common entities and their types

In [104]:
```
#explore the various entities and their labeled types in the article

for entity in processed_text2.ents:
    print(entity, entity.label_)
```

```
O.J. Simpson — PERSON
70 years old DATE
some 23 years DATE
one CARDINAL
American NORP
well over half CARDINAL
NBA ORG
NFL ORG
within 5 years DATE
Simpson PERSON
Simpson PERSON
19 DATE
Simpson PERSON
Simpson's PERSON
Simpson PERSON
Simpson PERSON
Simpson PERSON
Simpson PERSON
Board ORG
Simpson PERSON
Simpson PERSON
1 CARDINAL
9.5 CARDINAL
33 CARDINAL
CNN ORG
Mark Geragos PERSON
one-year DATE
2 CARDINAL
3 CARDINAL
one CARDINAL
Simpson PERSON
911 CARDINAL
Simpson's PERSON
the year DATE
one CARDINAL
Simpson PERSON
Simpson PERSON
October, 2017 DATE
Simpson PERSON
```

In [105]:
```
#from the above list, it appears that Simpson is the most common entity (type = P
ERSON)
#let's create a dictionary and store the various PERSON entities in it

persons = []

for entity in processed_text2.ents:
    if entity.label_ == 'PERSON':
        persons.append(entity)
```

In [ ]:
```
# to show the most frequent PERSON entities, let's use the Counter

persons_frequency = Counter(persons)
common_persons = persons_frequency.most_common(5)
common_persons
```

Observation: As suspected, Simpson (or OJ Simpson) is in the top 5 of frequent entities.

5. Find entities and their dependency (hint: entity.root.head)

Note: From the question above, it appears there are very few entities (a couple of PERSON, a few CARDINALs, a couple of DATEs, and a couple ORGs). Probably best to explore the PERSON and ORG entity types for this question.

In [ ]:
```
import numpy as np
import pandas as pd
```

In [ ]:
```
def entity_dep(entity, level):

    dependencies = []
    for ent in processed_text2.ents:
        if ent.root.head == 'PERSON':
            dependencies.append(token.lemma_)

    for ent in processed_text2.ents:
        if ent.root.dep_ =='adj':
            for child in ent.root.head.children:
                dependencies.append(child.lemma_)

    return dependencies
```

In [ ]:
```
print(entity_dep(processed_text2, 'Simpson'))
```

In [ ]:
```
#check all adjectives used with Simpson

def pos_words(sentence, token, ptag):
    sentences = [sent for sent in sentence.sents if token in sent.string]
    pwords = []
    for sent in sentences:
        for word in sent:
            if character in word.string:
                pwords.extend([child.string.strip() for child in word.children
                                                    if child.post_ == ptag])

    return Counter(pwords).most_common(10)

pos_words(processed_text2, 'Simpson', 'ADJ')
```

In [ ]:
```
# extract all  sentences that contains the term - Simpson
simpson = [sent for sent in document.sents if 'simpson' in sent.string.lower()]

# create dependency tree
sentence = simpson[2] for word in sentence:
print word, ': ', str(list(word.children))
```

In [ ]:

In [ ]:

In [ ]:

6. Find the most similar words in the article.

Note: Best to use word vectorization to compare similarity.

In [106]:
```
#this will print out the various vectors for each word in the article

for sentences in processed_text2.sents:
    for token in sentences:
        print(token, token.vector)
```

```
Probably [ -2.57519990e-01   2.74899989e-01  -2.48999998e-01  -3.22129987e-02
   1.87330004e-02   1.55190006e-01  -5.14380001e-02  -3.31860006e-01
  -1.01669997e-01   2.78189993e+00   6.91149980e-02   9.31370035e-02
   5.49589992e-01   3.08389992e-01  -1.79790005e-01  -5.98729998e-02
  -1.69119999e-01   5.06889999e-01   6.97999969e-02  -1.95439994e-01
  -2.51770001e-02  -1.01640001e-02  -9.32170004e-02  -1.17950000e-01
  -1.41360000e-01  -6.52960017e-02  -2.64710009e-01  -3.39100003e-01
   1.89070001e-01  -3.04369986e-01  -2.06819996e-01   1.78929999e-01
   1.79940000e-01   5.78230014e-03  -1.75129995e-01   1.44390002e-01
   1.96419999e-01   2.61599988e-01   1.85579993e-02  -1.49619997e-01
  -9.53060016e-02   3.61950010e-01  -2.57610008e-02   1.06830001e-01
   3.13080013e-01  -4.62340005e-02  -1.68660000e-01  -3.53839993e-01
   7.80370012e-02   1.54200003e-01  -1.23029999e-01  -1.58789996e-02
   1.38229996e-01  -5.01709990e-02   1.77640006e-01  -2.79570013e-01
   5.04559986e-02  -4.02619988e-01   3.76659989e-01   1.48130000e-01
  -1.05219997e-01  -3.94520015e-01  -1.27510005e-03   7.88250029e-01
   6.23590015e-02  -2.19559997e-01  -2.06159994e-01  -4.05919999e-02
   1.70189999e-02   2.39910007e-01   9.06710029e-02   1.64409995e-01
   5.60149997e-02  -1.75380006e-01   1.67459995e-01   1.32939994e-01
   5.70940003e-02   9.60009992e-02  -4.05160010e-01   3.53480011e-01
  -3.87159996e-02  -1.70190006e-01  -1.66529998e-01  -8.47230032e-02
   5.75819984e-02  -3.80719990e-01   3.33810002e-01  -5.18519998e-01
   3.83910000e-01  -1.18529998e-01  -4.18419987e-02  -1.49749994e-01
  -1.80960000e-01   5.27449995e-02   3.61999989e-01  -1.03830002e-01
   2.43019998e-01  -7.66170025e-02   1.56000003e-01   5.08520007e-01
  -3.19849998e-01  -2.64699999e-02   6.59700036e-02   1.18490003e-01
   4.34439987e-01  -1.06280005e+00   1.15050003e-02  -2.39419997e-01
  -5.80630004e-02   1.02210000e-01   1.38219997e-01  -3.66959989e-01
   1.08780004e-02   1.32269993e-01  -1.72800004e-01   1.12709999e-01
   3.14179994e-02  -2.63040006e-01  -3.83749992e-01  -8.76659974e-02
   2.40750000e-01  -1.11369997e-01   4.68479991e-02  -3.71870011e-01
   1.34950001e-02   2.25130007e-01   8.56720004e-03  -4.55449998e-01
  -4.20829989e-02  -1.56420007e-01   2.28850003e-02   2.19090000e-01
  -5.85829979e-03   2.39050001e-01   1.96319997e-01  -3.83390009e-01
  -1.19360000e-01  -2.24910006e-01  -3.06430012e-01  -2.03450006e-02
  -1.82669997e+00   3.63709986e-01   1.81329995e-01   1.21440001e-01
   2.21369993e-02  -3.58559996e-01  -3.30049992e-01  -1.35810003e-01
   2.71560013e-01   4.60259989e-02   1.75449997e-02   3.76890004e-02
   2.88249999e-02   1.32229999e-01  -1.81979999e-01  -1.96679994e-01
   2.06870005e-01  -1.92049995e-01  -4.81679998e-02  -3.96220013e-03
  -8.89469981e-02   6.46590022e-04  -4.88609999e-01  -2.54319996e-01
  -3.34919989e-01  -2.17590004e-01  -5.99399991e-02   2.39749998e-02
   2.39419997e-01  -1.72729999e-01  -7.68219978e-02   1.96810007e-01
   5.11379987e-02  -1.42260000e-01  -4.80250001e-01   1.35230005e-01
  -9.35830027e-02  -7.30080009e-02   1.50179997e-01  -2.51060009e-01
   8.27889964e-02   4.98749986e-02  -5.09090006e-01  -1.65800005e-01
  -7.80069977e-02  -2.18700007e-01  -7.09339976e-02  -9.70249996e-02
   3.96189988e-02   2.16690004e-01   3.11390013e-01   2.10350007e-01
  -3.28229994e-01   1.98080003e-01   5.02099991e-01  -1.64150000e-01
  -1.91459998e-01  -1.55980006e-01   6.69329986e-02   3.90099990e-03
   2.30969995e-01  -2.54970014e-01  -3.61400004e-03  -2.40620002e-01
  -4.47399989e-02  -1.04889996e-01  -1.34890003e-03  -3.42819989e-01
   1.53229997e-01   9.36539993e-02   1.03050001e-01  -6.79380000e-02
  -1.82710007e-01  -5.28779984e-01   3.39739993e-02  -5.56100011e-02
  -5.82080008e-03   1.36879995e-01  -3.34609985e-01  -3.80019993e-01
   4.26730007e-01  -5.60950004e-02   1.64460003e-01  -1.95360005e-01
   1.78259999e-01  -1.54510006e-01   2.35300004e-01   6.29810011e-03
  -7.87559971e-02   5.83360009e-02  -2.28430003e-01  -9.56640020e-02
   3.50899994e-01   1.40740007e-01   1.17239997e-01  -2.14430004e-01
   4.31860000e-01  -2.62169987e-01  -1.88500002e-01   1.65989995e-01
  -5.35180010e-02   5.46360016e-02  -1.89720001e-02   4.00330007e-01
  -5.31499982e-02  -1.83009997e-01   7.59169981e-02  -1.01520002e-01
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.
```

In [149]:
```
for noun_chunk in processed_text2.noun_chunks:
    print(noun_chunk)
```

```
no other celebrity athlete
the same exact situation
O.J. Simpson
parole
a conviction
robbery
kidnapping
conspiracy
the most sensationalized murder acquittals
American history
half
NBA and NFL players
bankruptcy protection
5 years
retirement
undiagnosed brain injuries
employment
mood swings
bad judgement
bad judgment
what
Simpson
a cause
his botched attempt
intimate family photos
other memorabilia
Many players
a near-perfect storm
setting
a desperate need
money
a misguided attempt
it
the property
his own name
image
likeness
Some former athletes
a cocoon
celebrity
the demons
a failure
the role
contrition
remorse
Those character attributes
a judge
jury
a parole board
forgiveness
We
crimes
former athletes
fame
greater number
we
a version
these circumstances
lessons
therefrom
I
the bad guy
```

In [139]:
```
proc_vec = processor('''Probably no other celebrity athlete will ever be in the s
ame exact situation as O.J. Simpson — 70 years old asking for parole after a conv
iction for robbery, kidnapping, and conspiracy, some 23 years after one of the mo
st sensationalized murder acquittals in American history.

But well over half of NBA and NFL players file for bankruptcy protection within 5
years of retirement. Many will have diagnosed or undiagnosed brain injuries that
prevent or hamper employment, and increase depression, mood swings, and bad judge
ment. And bad judgment is what Simpson admitted as a cause for his botched attemp
t to reclaim intimate family photos and other memorabilia. Many players may face
a near-perfect storm in a similar setting — a desperate need for money and a misg
uided attempt to retrieve it, especially since the property involves his own name
, image and likeness.

Some former athletes have also lived in a cocoon of celebrity, fighting the demon
s of being self-absorbed. That can lead to a failure to be humble, to understand
the role of contrition and remorse. Those character attributes are important when
begging a judge, jury, or a parole board for forgiveness. We know crimes by forme
r athletes defrocked from fame occur in greater number than we would like. So a v
ersion of these circumstances and lessons therefrom is not far-fetched. If convic
ted, instead of saying, “I’m not the bad guy, somebody else was,” the more strate
gic verse is “Yes, I was the bad guy, but I am not any more because…”

If athletes are paying careful attention to Simpson’s own statements, a celeb sin
ce he was 19, they should notice that he essentially attempted to be his own coun
sel and re-litigate the trial he lost. Instead, athletes should follow the advice
of counsel to be humble, contrite, and remorseful. That is a Simpson mistake no s
imilarly situated player should replicate.

For example, Simpson’s underlying theme throughout the hearing was that his “secu
rity” team held the guns, and perpetrated the violence, not him. Simpson repeated
ly admitted the error of bad judgment in bringing thugs with him. But Simpson ris
ked a parole board perception that he failed to take responsibility for his orche
stration of the events that led to his conviction. There was a conspiracy count,
so he did not have to hold a gun to be guilty. Additionally, Simpson did not test
ify during the robbery trial. The jury heard other testimony not so favorable. Si
mpson and players facing a parole hearing should remember the Board is looking at
his character, and admission of the full extent of his errors. How else can one a
ssess whether he is a risk for repeating his errors unless they appreciate the gr
avity and full extent of those errors.

Simpson was successful at the parole hearing. But other former stars should not b
e confused. Simpson had the advantage of (1) a sentence that was extraordinarily
long (9.5 to 33 years for what CNN commentator Mark Geragos said is typically a o
ne-year sentence), (2) both victims apparently being at peace with parole, (3) an
d incident-free actions during incarceration. Future players are not likely to ha
ve all those advantages.

Instead, athletes should focus on the strategy of humility, contrition, and remor
se. More importantly, they should avoid appearing delusional about their own sord
id history if one exists. Even ignoring the murder trial, Simpson still had the c
onviction for domestic violence, after multiple 911 calls from his then-wife. Yet
Simpson’s opening statement included his self-assessment that he basically led a
“conflict-free” life. He probably was thinking about the adoration of stardom he
received prior to the murder trial. The delusion is that repeated violence agains
t a woman is not a linear equation in the character analysis. It is a crime that
colors the entire perception of the person, regardless of the year perpetrated.

As one board member cautioned, Simpson will receive strict scrutiny on yet-to-be-
finalized conditions to his parole. I expect those conditions to have muscularity
Simpson never faced on the field. He talked of being conflict-free. Starting in O
```

In [147]:
```
celebrity = proc_vec.vocab['celebrity']
athlete = proc_vec.vocab['athlete']
crimes = proc_vec.vocab['crimes']
celeb = proc_vec.vocab['celeb']
remorse = proc_vec.vocab['remorse']
humility = proc_vec.vocab['humility']
conviction = proc_vec.vocab['conviction']
judge = proc_vec.vocab['judge']
judgment = proc_vec.vocab['judgment']
stars = proc_vec.vocab['stars']
decision = proc_vec.vocab['decision']
behavior = proc_vec.vocab['behavior']
charges = proc_vec.vocab['charges']
criminal = proc_vec.vocab['criminal']
misguided = proc_vec.vocab['misguided']
absorbed = proc_vec.vocab['absorbed']
hearing = proc_vec.vocab['hearing']
```

In [148]:
```
celebrity.similarity(absorbed)
```

Out[148]:
```
0.076347151338033409
```

In [ ]:

In [ ]:
```
proc_text = processed_text2
```

In [ ]:
```
Probably = proc_text.vocab['Probably']
no = proc_text.vocab['no']
```

In [ ]:
```
Probably.similarity(no)
```

In [ ]:
```
proc_text.sents
```

In [ ]:
```
for sentence in processed_text2.sents:
    print(n, sentence)
    n+=1
```

In [ ]:
```
proc_vectext = []

for sentences in processed_text2.sents:
    for token in sentences:
        proc_vectext.append(token)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
# Find words (adjectives) that describe Simpson

def get_adjectives(processed_text2, person_lemma):

    Simpson_adj = [] #create an empty list and then for loop to append to list
    for ent in processed_text2.ents:
        if ent.lemma_== person_lemma:
            for token in ent.subtree:
                if token.pos_ == 'ADJ':
                    Simpson_adj.append(token.lemma_)


    for ent in processed_text2.ents:
        if ent.lemma_ == person_lemma:
            if ent.root.dep_ == 'nsubj':
                for child in ent.root.head.children:
                    if child.dep_ == 'acomp':
                        Simpson_adj.append(child.lemma_)


    return Simpson_adj

print(get_adjectives(processed_text2, 'simpson'))
```