In [27]:
```python
import spacy
```

In [28]:
```python
nlp = spacy.load('en_core_web_lg')
```

In [29]:
```python
nlp(u'lion').vector
```

Out[29]:
```
array([ 1.8963e-01, -4.0309e-01,  3.5350e-01, -4.7907e-01, -4.3311e-01
,
        2.3857e-01,  2.6962e-01,  6.4332e-02,  3.0767e-01,  1.3712e+00
,
       -3.7582e-01, -2.2713e-01, -3.5657e-01, -2.5355e-01,  1.7543e-02
,
        3.3962e-01,  7.4723e-02,  5.1226e-01, -3.9759e-01,  5.1333e-03
,
       -3.0929e-01,  4.8911e-02, -1.8610e-01, -4.1702e-01, -8.1639e-01
,
       -1.6908e-01, -2.6246e-01, -1.5983e-02,  1.2479e-01, -3.7276e-02
,
       -5.7125e-01, -1.6296e-01,  1.2376e-01, -5.5464e-02,  1.3244e-01
,
        2.7519e-02,  1.2592e-01, -3.2722e-01, -4.9165e-01, -3.5559e-01
,
       -3.0630e-01,  6.1185e-02, -1.6932e-01, -6.2405e-02,  6.5763e-01
,
       -2.7925e-01, -3.0450e-03, -2.2400e-02, -2.8015e-01, -2.1975e-01
,
       -4.3188e-01,  3.9864e-02, -2.2102e-01, -4.2693e-02,  5.2748e-02
,
        2.8726e-01,  1.2315e-01, -2.8662e-02,  7.8294e-02,  4.6754e-01
,
       -2.4589e-01, -1.1064e-01,  7.2250e-02, -9.4980e-02, -2.7548e-01
,
       -5.4097e-01,  1.2823e-01, -8.2408e-02,  3.1035e-01, -6.3394e-02
,
       -7.3755e-01, -5.4992e-01,  9.9999e-02, -2.0758e-01, -3.9674e-02
,
        2.0664e-01, -9.7557e-02, -3.7092e-01,  2.7901e-01, -6.2218e-01
,
       -1.0280e-01,  2.3271e-01,  4.3838e-01,  3.2445e-02, -2.9866e-01
,
       -7.3611e-02,  7.1594e-01,  1.4241e-01,  2.7770e-01, -3.9892e-01
,
        3.6656e-02,  1.5759e-01,  8.2014e-02, -5.7343e-01,  3.5457e-01
,
        2.2491e-01, -6.2699e-01, -8.8106e-02,  2.4361e-01,  3.8533e-01
,
       -1.4083e-01,  1.7691e-01,  7.0897e-02,  1.7951e-01, -4.5907e-01
```

```
,
       -8.2120e-01, -2.6631e-02,  6.2549e-02,  4.2415e-01, -8.9630e-02
,
       -2.4654e-01,  1.4156e-01,  4.0187e-01, -4.1232e-01,  8.4516e-02
,
       -1.0626e-01,  7.3145e-01,  1.9217e-01,  1.4240e-01,  2.8511e-01
,
       -2.9454e-01, -2.1948e-01,  9.0460e-01, -1.9098e-01, -1.0340e+00
,
       -1.5754e-01, -1.1964e-01,  4.9888e-01, -1.0624e+00, -3.2820e-01
,
       -1.1232e-02, -7.9482e-01,  3.7275e-01, -6.8710e-03, -2.5772e-01
,
       -4.7005e-01, -4.1387e-01, -6.4089e-02, -2.8033e-01, -4.0778e-02
,
       -2.4866e+00,  6.2494e-03, -1.0210e-02,  1.2752e-01,  3.4965e-01
,
       -1.2571e-01,  3.1570e-01,  4.1926e-01,  2.0056e-01, -5.5984e-01
,
       -2.2801e-01,  1.2012e-01, -2.0518e-03, -8.9764e-02, -8.0373e-02
,
        1.1969e-02, -2.6978e-01,  3.4829e-01,  7.3664e-03, -1.1137e-01
,
        6.3410e-01,  3.8449e-01, -6.2248e-01,  4.1145e-02,  2.5922e-01
,
        6.5811e-01, -4.9548e-01, -1.3030e-01, -3.8279e-01,  1.1156e-01
,
       -4.3085e-01,  3.4473e-01,  2.7109e-02, -2.5108e-01, -2.8011e-01
,
        2.1662e-01,  3.2660e-01,  5.5895e-02,  7.6077e-02, -5.2480e-02
,
        4.5928e-02, -2.5266e-01,  5.2845e-01, -1.3145e-01, -1.2453e-01
,
        4.0556e-01,  3.1877e-01,  2.4415e-02, -2.2620e-01, -6.1960e-01
,
       -4.0886e-01, -3.5534e-02, -5.5123e-03,  2.3438e-01,  8.7854e-01
,
       -2.5161e-01,  4.0600e-01, -4.4284e-01,  3.4934e-01, -5.6429e-01
,
       -2.3676e-01,  6.2199e-01, -2.8175e-01,  4.2024e-01,  1.0043e-01
,
       -1.4720e-01,  4.9593e-01, -3.5850e-01, -1.3998e-01, -2.7494e-01
,
        2.3827e-01,  5.7268e-01,  7.9025e-02,  1.7872e-02, -2.1829e-01
,
        5.5050e-02, -5.4200e-01,  1.6788e-01,  3.9065e-01,  3.0209e-01
,
        2.3040e-01, -3.9351e-02, -2.1078e-01, -2.7224e-01,  1.6907e-01
,
        5.4819e-01,  9.4888e-02,  7.9798e-01, -6.6158e-02,  1.9844e-01
```

```
,
         2.0307e-01,  4.4808e-02, -1.0240e-01, -6.9909e-02, -3.6756e-02
,
         9.5159e-02, -2.7830e-01, -1.0597e-01, -1.6276e-01, -1.8211e-01
,
        -3.1897e-01, -2.1633e-01,  1.4994e-01, -7.2057e-02,  2.2264e-01
,
        -4.5551e-01,  3.0341e-01,  1.8431e-01,  2.1681e-01, -3.1940e-01
,
         2.6426e-01,  5.8106e-01,  5.4635e-02,  6.3238e-01,  4.3169e-01
,
         9.0343e-02,  1.9494e-01,  3.5483e-01, -2.0706e-02, -7.3117e-01
,
         1.2941e-01,  1.7418e-01, -1.5065e-01,  5.3355e-02,  4.4794e-02
,
        -1.6600e-01,  2.2007e-01, -5.3970e-01, -2.4968e-01, -2.6464e-01
,
        -5.5515e-01,  5.8242e-01,  2.2295e-01,  2.4433e-01,  4.5275e-01
,
         3.4693e-01,  1.2255e-01, -3.9059e-02, -3.2749e-01, -2.7891e-01
,
         1.3766e-01,  3.8392e-01,  1.0543e-03, -1.0242e-02,  4.9205e-01
,
        -1.7922e-01,  4.1215e-02,  1.3547e-01, -2.0598e-01, -2.3194e-01
,
        -7.7701e-01, -3.8237e-01, -7.6383e-01,  1.9418e-01, -1.5441e-01
,
         8.9740e-01,  3.0626e-01,  4.0376e-01,  2.1738e-01, -3.8050e-01
],
      dtype=float32)
```

In [30]:
```
nlp(u'lion').vector.shape
```

Out[30]:
```
(300,)
```

In [31]:
```
nlp(u'The quick brown fox jumped').vector
```

Out[31]:
```
array([-2.09217995e-01, -2.78227981e-02, -3.57064009e-02,  1.55218393e
-01,
       -1.28050027e-02,  1.31627038e-01, -1.99465990e-01,  4.75811996e
-02,
        1.26798794e-01,  1.64792800e+00, -3.57592016e-01, -1.39875397e
-01,
       -1.26122087e-02, -2.02728346e-01, -2.25237608e-01,  2.15431936e
-02,
        7.78958052e-02,  9.29676056e-01, -2.75549982e-02, -3.71005982e
-01,
       -1.42800003e-01, -3.66641544e-02, -1.07376035e-02, -1.84352830e
-01,
```

```
        2.29006782e-02, -5.17717972e-02, -2.78652012e-01, -1.19738199e
-01,
        5.10960072e-03, -2.85990000e-01, -1.58261746e-01,  2.96241999e
-01,
        1.09597601e-01, -4.18331996e-02,  1.87256075e-02, -1.03439607e
-01,
       -5.10879979e-02, -3.51091917e-03, -6.81461841e-02, -2.05657601e
-01,
        1.66347414e-01, -9.31599736e-03, -4.61134054e-02, -1.05457589e
-01,
        2.31313989e-01,  1.80005193e-01, -2.06444815e-01, -1.37050152e
-02,
        1.70106202e-01, -2.19812002e-02, -2.14003205e-01,  1.07415602e
-01,
       -2.80592032e-02, -5.23634031e-02, -4.86331955e-02, -1.04047179e
-01,
        1.27018047e-02,  2.02107817e-01, -1.18217587e-01, -1.51981995e
-01,
        5.12168184e-02,  5.64177930e-02,  5.44355996e-02,  5.15560023e
-02,
        5.14240041e-02, -1.37740612e-01, -8.45800620e-03, -1.13289997e
-01,
        1.34828404e-01, -2.25100014e-02,  1.19754001e-01,  5.12372032e
-02,
        5.40098026e-02,  7.36430064e-02, -1.59269981e-02, -2.37861007e
-01,
       -7.90134817e-02, -1.30640596e-01,  4.45272028e-02, -4.98013981e
-02,
       -7.30358064e-02,  1.80923387e-01,  4.17133979e-02,  4.45965007e
-02,
        1.77781999e-01,  1.66720040e-02,  6.30389988e-01,  4.30720389e
-01,
        2.10017413e-01,  1.68576598e-01,  7.29599595e-03,  1.58338398e
-01,
        3.79000008e-02, -3.46915185e-01,  1.33294016e-01, -9.09054056e
-02,
       -6.90027997e-02, -5.71460137e-03,  6.47759885e-02, -2.06994608e
-01,
        1.64740205e-01, -2.13680025e-02, -1.30543396e-01, -3.02743949e
-02,
       -1.03020016e-02, -6.46848023e-01,  1.80351987e-01, -9.54739973e
-02,
       -1.40800001e-02, -3.42049934e-02, -4.12111953e-02, -1.11505605e
-01,
        1.27132609e-01, -1.88302785e-01,  1.13260604e-01,  1.80767015e
-01,
        6.53811991e-02, -1.58561952e-02,  9.68780071e-02,  7.01044053e
-02,
        4.83391993e-02, -1.63396388e-01, -4.51015905e-02, -9.61597934e
-02,
```

```
       -3.00761998e-01,  1.63111001e-01,  4.52036038e-02, -7.97460005e
-02,
       -1.00108802e-01, -5.79720028e-02, -1.80559454e-03, -8.24856013e
-02,
       -1.32837012e-01,  7.58986026e-02,  1.25125393e-01, -1.17681786e
-01,
       -2.02512026e-01, -5.02933972e-02, -3.09894979e-02, -1.96231812e
-01,
       -2.05938005e+00, -1.78128034e-02,  1.31141797e-01,  5.38920052e
-02,
        2.48921394e-01, -8.28451961e-02,  2.29120012e-02,  4.11577933e
-02,
        6.28991947e-02,  1.26971409e-01,  3.54279988e-02, -1.65512592e
-01,
        1.54437393e-01, -8.91011953e-02, -2.38956213e-01,  3.74409966e
-02,
       -1.47978395e-01,  1.23184405e-01,  7.47255981e-02, -8.21532011e
-02,
       -3.02814040e-02, -1.99475795e-01, -2.98164397e-01, -5.18049970e
-02,
       -5.98729961e-02, -1.95260614e-01, -1.14224993e-01, -7.07461983e
-02,
       -1.35402009e-01, -1.59228414e-01,  1.33558795e-01,  1.44477993e
-01,
       -2.50075996e-01, -2.07789987e-01, -3.69598001e-01, -1.38345197e
-01,
        2.83456177e-01, -4.93402767e-04, -2.13945992e-02,  1.83038004e
-02,
       -3.45086008e-02, -1.53184995e-01, -2.21591994e-01, -1.14851996e
-01,
        9.88069996e-02,  1.29306391e-01, -5.40879965e-02, -4.65750024e
-02,
       -4.59119631e-03,  1.25648379e-02,  2.73273796e-01,  3.86317968e
-02,
        6.49093613e-02,  3.70982066e-02,  1.26923593e-02,  1.85716420e
-01,
        1.69222593e-01,  1.06119812e-01,  1.43199565e-03, -7.52920061e
-02,
       -1.51146010e-01,  4.95879985e-02, -2.87192404e-01, -2.74599995e
-02,
        2.10097998e-01,  1.37594968e-01, -1.26467999e-02, -1.47451401e
-01,
       -6.98073283e-02,  1.16517963e-02,  7.65941963e-02,  8.24979991e
-02,
       -3.28646004e-02,  2.22982407e-01,  8.53662044e-02,  1.13203190e
-01,
       -3.06712002e-01,  6.65521994e-02, -2.42485963e-02,  1.85453802e
-01,
        6.33899868e-03,  2.36580381e-03, -6.98355958e-02,  9.46911126e
-02,
```

```
       -2.32161760e-01, -1.91601396e-01,  2.09780186e-01,  1.91669196e
-01,
        2.42458023e-02,  4.20044035e-01, -1.50683997e-02,  4.16760286e
-03,
       -7.33660609e-02,  5.19229956e-02, -1.21735949e-02, -8.69527981e
-02,
       -1.26489595e-01, -6.72094822e-02,  1.63832814e-01,  2.75269568e
-01,
       -1.64249986e-01, -1.52959794e-01, -5.64128049e-02, -9.03348029e
-02,
        3.59305963e-02, -1.62403792e-01, -9.81536135e-02,  3.03588063e
-02,
       -2.10355401e-01,  5.90659976e-02, -3.81862000e-02, -1.31029800e
-01,
       -1.98952004e-01,  1.02112450e-01,  1.69432998e-01,  4.52830084e
-02,
        2.06268996e-01,  1.30002588e-01,  3.64079997e-02,  1.67501979e
-02,
        1.52415991e-01,  8.69902000e-02,  1.43315807e-01, -1.03890002e
-01,
        2.32943982e-01, -2.00447604e-01,  8.02273974e-02, -4.50324044e
-02,
       -2.33428001e-01,  9.64580029e-02, -8.72388482e-04,  2.65751779e
-01,
       -4.15487401e-02, -1.38139397e-01, -3.27680036e-02, -4.31547984e
-02,
        3.97300012e-02,  1.05935797e-01,  9.52792179e-04,  3.66709009e
-02,
        1.43723994e-01,  9.49178040e-02, -1.12830400e-01,  1.76364794e
-01,
       -2.80858018e-02, -3.65898088e-02,  7.58539960e-02,  7.08954111e
-02,
       -7.30042011e-02,  1.27999904e-02, -2.66412795e-01,  1.81497976e
-01,
       -2.59018000e-02, -1.16812006e-01, -7.90375918e-02,  2.10512038e
-02,
        2.83426046e-02,  4.26702015e-02, -1.21463798e-01, -1.45020094e
-02],
      dtype=float32)
```

In [32]:
```
nlp(u'The quick brown fox jumped').vector.shape
```

Out[32]: (300,)

In [33]:
```
nlp(u'fox').vector.shape
```

Out[33]: (300,)

In [34]:
```
tokens = nlp(u'lion cat pet')
```

In [35]:
```
for token1 in tokens:
    for token2 in tokens:
        print(token1.text, token2.text, token1.similarity(token2))
```

```
lion lion 1.0
lion cat 0.5265437
lion pet 0.39923772
cat lion 0.5265437
cat cat 1.0
cat pet 0.7505456
pet lion 0.39923772
pet cat 0.7505456
pet pet 1.0
```

In [36]:
```
tokens = nlp(u'like love hate')

for token1 in tokens:
    for token2 in tokens:
        print(token1.text, token2.text, token1.similarity(token2))
```

```
like like 1.0
like love 0.65790397
like hate 0.6574652
love like 0.65790397
love love 1.0
love hate 0.6393099
hate like 0.6574652
hate love 0.6393099
hate hate 1.0
```

Observation: Though love and hate have opposite meaning, they are used in the same context. Eg. You love a movie or you have a movie. So the similarity between love and hate is high.

In [37]:
```
# Number of vocab vectors in the library
len(nlp.vocab.vectors)
```

Out[37]:
```
684831
```

In [38]:
```
print(nlp.vocab.vectors.shape)
```

```
(684831, 300)
```

In [39]:
```
tokens = nlp(u'dog cat nargle')
for token in tokens:
    print(token.text, token.has_vector, token.vector_norm, token.is_oov)
```

```
dog True 7.0336733 False
cat True 6.6808186 False
nargle False 0.0 True
```

In [40]:
```
tokens = nlp('John Joan Jayashri')
for token in tokens:
    print(token.text, token.has_vector, token.vector_norm, token.is_oov)
```

```
John True 6.533578 False
Joan True 6.178602 False
Jayashri False 0.0 True
```

In [41]:
```
from scipy import spatial

cosine_similarity = lambda x, y : 1 - spatial.distance.cosine(x, y)

king = nlp.vocab['king'].vector
man = nlp.vocab['man'].vector
woman = nlp.vocab['woman'].vector

# Compute a new vector
new_vector = king - man + woman
computed_similarities = []

for word in nlp.vocab:
    if word.has_vector:
        if word.is_lower:
            if word.is_alpha:
                similarity = cosine_similarity(new_vector, word.vector)
                computed_similarities.append((word, similarity))
```

In [42]:
```
# Sorting computed_similarities in descending order from the most simila

computed_similarities = sorted(computed_similarities, key=lambda item: -
print([w[0].text for w in computed_similarities[0:10]])
```

```
['king', 'woman', 'she', 'lion', 'who', 'fox', 'brown', 'when', 'dare'
, 'cat']
```

In [ ]:

In [ ]:

In [ ]: