In [ ]:

```
import gensim.downloader as api
wv = api.load('word2vec-google-news-300')
# wv = api.load('/Users/florrie/gensim-data/word2vec-google-news-300')
```

In [10]:

```
for index, word in enumerate(wv.index_to_key):
    if index == 10:
        break
    print(f"word #{index}/{len(wv.index_to_key)} is {word}")
```

```
word #0/3000000 is </s>
word #1/3000000 is in
word #2/3000000 is for
word #3/3000000 is that
word #4/3000000 is is
word #5/3000000 is on
word #6/3000000 is ##
word #7/3000000 is The
word #8/3000000 is with
word #9/3000000 is said
```

In [17]:

```
vec_king = wv['king']
print(vec_king)
```

```
[ 1.25976562e-01  2.97851562e-02  8.60595703e-03  1.39648438e-01
 -2.56347656e-02 -3.61328125e-02  1.11816406e-01 -1.98242188e-01
  5.12695312e-02  3.63281250e-01 -2.42187500e-01 -3.02734375e-01
 -1.77734375e-01 -2.49023438e-02 -1.67968750e-01 -1.69921875e-01
  3.46679688e-02  5.21850586e-03  4.63867188e-02  1.28906250e-01
  1.36718750e-01  1.12792969e-01  5.95703125e-02  1.36718750e-01
  1.01074219e-01 -1.76757812e-01 -2.51953125e-01  5.98144531e-02
  3.41796875e-01 -3.11279297e-02  1.04492188e-01  6.17675781e-02
  1.24511719e-01  4.00390625e-01 -3.22265625e-01  8.39843750e-02
  3.90625000e-02  5.85937500e-03  7.03125000e-02  1.72851562e-01
  1.38671875e-01 -2.31445312e-01  2.83203125e-01  1.42578125e-01
  3.41796875e-01 -2.39257812e-02 -1.09863281e-01  3.32031250e-02
 -5.46875000e-02  1.53198242e-02 -1.62109375e-01  1.58203125e-01
 -2.59765625e-01  2.01416016e-02 -1.63085938e-01  1.35803223e-03
 -1.44531250e-01 -5.68847656e-02  4.29687500e-02 -2.46582031e-02
  1.85546875e-01  4.47265625e-01  9.58251953e-03  1.31835938e-01
  9.86328125e-02 -1.85546875e-01 -1.00097656e-01 -1.33789062e-01
 -1.25000000e-01  2.83203125e-01  1.23046875e-01  5.32226562e-02
 -1.77734375e-01  8.59375000e-02 -2.18505859e-02  2.05078125e-02
```

In [12]:

```
try:
    vec_cameroon = wv['cameroon']
except KeyError:
    print("The word 'cameroon' does not appear in this model")
```

```
The word 'cameroon' does not appear in this model
```

Word2Vec supports several word similarity tasks out of the box. You can see how the similarity intuitively

decreases as the words get less and less similar.

In [23]:

```
pairs = [
    ('car', 'minivan'),   # a minivan is a kind of car
    ('car', 'bicycle'),   # still a wheeled vehicle
    ('car', 'airplane'),  # ok, no wheels, but still a vehicle
    ('car', 'cereal'),    # ... and so on
    ('car', 'communism'),
]
for w1, w2 in pairs:
    print('%r\t%r\t%.2f' % (w1, w2, wv.similarity(w1, w2)))
```

```
'car'   'minivan'       0.69
'car'   'bicycle'       0.54
'car'   'airplane'      0.42
'car'   'cereal'        0.14
'car'   'communism'     0.06
'man'   'king'  0.23
'woman' 'queen' 0.32
```

Print the 5 most similar words to “car” or “minivan”

In [25]:

```
print(wv.most_similar(positive=['car', 'minivan'], topn=5))
```

```
[('SUV', 0.8532192707061768), ('vehicle', 0.8175783753395081), ('picku
p_truck', 0.7763688564300537), ('Jeep', 0.7567334175109863), ('Ford_Ex
plorer', 0.7565720081329346)]
```

In [27]:

```
print(wv.doesnt_match(['fire', 'water', 'land', 'sea', 'air', 'car']))
```

```
boot
```