問68 英語の文書のcos類似度の比較 uberについて書かれた記事2本とlyftについて書かれた記事1本の類似度を比較してみる。

- uber1.txt (https://medium.com/free-code-camp/dark-genius-how-programmers-at-uber-volkswagen-and-zenefits-helped-their-employers-break-the-law-b7a7939c6591)
- uber2.txt (https://medium.com/sandpapersuit/side-hustle-as-a-sign-of-the-apocalypse-e7027a889fc2)
- lyft1.txt (https://medium.com/@johnzimmer/all-lyft-rides-are-now-carbon-neutral-55693af04f36)

文章の量を一致させるため一部削っている。

In [167]:

```
pip install nltk
```

```
Requirement already satisfied: nltk in /anaconda3/lib/python3.7/site-packages (3.4.
1)
Requirement already satisfied: six in /anaconda3/lib/python3.7/site-packages (from n
ltk) (1.12.0)
Note: you may need to restart the kernel to use updated packages.
```

In [168]:

```
import numpy as np
import nltk
from functools import reduce
import collections
nltk.download('punkt')

def word_list(file_name):
    word_list1 = []
    with open(file_name) as f:
        for s_line in f:
            word_list1.append(nltk.word_tokenize(s_line))
    word_list1 = reduce(lambda a, b: a + b, word_list1)
    return word_list1

def count_f(word_list, all_word_list):
    count_list = []
    for i in range(all_word_list.size):
        word = all_word_list[i]
        count = np.count_nonzero(word_list == word)
#       t1 = (word, count)
        t1 = count
        count_list.append(t1)
    return count_list

def cos(f, g):
    return np.dot(f, g)/(np.linalg.norm(f)*np.linalg.norm(g))
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/taishieguchi/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [169]:

```
uber1_list = np.array(word_list('uber1.txt'))
uber2_list = np.array(word_list('uber2.txt'))
lyft1_list = np.array(word_list('lyft1.txt'))

all_word_list = np.unique(np.hstack((uber1_list, uber2_list, lyft1_list)))

uber1_vec = np.array(count_f(uber1_list, all_word_list))
uber2_vec = np.array(count_f(uber2_list, all_word_list))
lyft1_vec = np.array(count_f(lyft1_list, all_word_list))
```

In [170]:

```
cos(uber1_vec, uber1_vec)
```

Out[170]:

```
1.0
```

In [171]:

```
cos(uber1_vec, uber2_vec)
```

Out[171]:

```
0.6395783804293877
```

In [172]:

```
cos(uber2_vec, lyft1_vec)
```

Out[172]:

```
0.6849740392966016
```

In [173]:

```
cos(uber1_vec, lyft1_vec)
```

Out[173]:

```
0.7263543672138268
```

uber1とuber2の値が近くなることを期待したが精度はイマイチであった（むしろlyftとuberの方が近づいてしまっている）。

別の文書(newstimesの記事)で比較してみる。

- realestate1.txt (https://medium.com/free-code-camp/dark-genius-how-programmers-at-uber-volkswagen-and-zenefits-helped-their-employers-break-the-law-b7a7939c6591)
- realestate2.txt (https://medium.com/sandpapersuit/side-hustle-as-a-sign-of-the-apocalypse-e7027a889fc2)
- sports1.txt (https://medium.com/@johnzimmer/all-lyft-rides-are-now-carbon-neutral-55693af04f36)

In [178]:

```
realestate1_list = np.array(word_list('realestate1.txt'))
realestate2_list = np.array(word_list('realestate2.txt'))
sports1_list = np.array(word_list('sports1.txt'))

all_word_list = np.unique(np.hstack((realestate1_list, realestate2_list, sports1_list)))

realestate1_vec = np.array(count_f(realestate1_list, all_word_list))
realestate2_vec = np.array(count_f(realestate2_list, all_word_list))
sports1_vec = np.array(count_f(sports1_list, all_word_list))
```

In [179]:

```
cos(realestate1_vec, realestate1_vec)
```

Out[179]:

```
1.0
```

In [180]:

```
cos(realestate1_vec, realestate2_vec)
```

Out[180]:

```
0.8383622904249884
```

In [181]:

```
cos(realestate1_vec, sports1_vec)
```

Out[181]:

```
0.7632852370301972
```

In [182]:

```
cos(realestate2_vec, sports1_vec)
```

Out[182]:

```
0.7255375589022625
```

不動産の記事とスポーツの記事だが、きちんとその類似度を分類できた。

残りの課題については締め切りに間に合わなかったため以下のリンクに貼った。
https://github.com/shierote/NLP_practice (https://github.com/shierote/NLP_practice)