IR Exercise 3

April 24, 2020

1 IR Exercise 3

1.1 setting up environment

[1]:
```
from json import load
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
invindx_file = open('/home/hakim/Documents/semester 8/IR/HW_3/code/invert_index.
 ↪json', 'r')
invindx = load(invindx_file)
invindx_file.close()
```

[3]:
```
md_file = open('/home/hakim/Documents/semester 8/IR/HW_3/code/parsed_data.
 ↪json', 'r')
md = load(md_file)
md_file.close()
```

1.2 Basic Statics (question #2)

[4]:
```
all_cars = []
all_cars_year = []
all_cars_model = []
for d in md:
    car = md[d]['file_name']

    all_cars.append(car)
    all_cars_year.append(car.split('_')[0])
    all_cars_model.append(car.split('_')[1])

all_cars = pd.array(all_cars).unique()
all_cars_year = pd.array(all_cars_year)
all_cars_model = pd.array(all_cars_model)
```

[5]:
```
pd.DataFrame((['All Document Number',len(md)],
              ['All Word Number',len(invindx)],
              ['All Cars/File Number',len(all_cars)],
             ['All Cars Number: ',len(all_cars)],
             ['All Cars Models: ', len(all_cars_model.unique())])
              ,columns=['Variable','Counts'])
```

[5]:

	Variable	Counts
0	All Document Number	42288
1	All Word Number	21605
2	All Cars/File Number	597
3	All Cars Number:	597
4	All Cars Models:	30

1.2.1 All Year comment Statics

[6]:
```
year_stat = pd.DataFrame(all_cars_year.value_counts(), columns=['Count'])
all_year_count = 0
for year in all_cars_year.value_counts():
    all_year_count += int(year)
year_stat.loc['All Years'] = all_year_count
year_stat
```

[6]:

	Count
2007	18903
2008	15438
2009	7947
All Years	42288

1.2.2 Brand comment (Document) number

[7]:
```
pd.DataFrame(pd.array(all_cars_model).value_counts(), columns=['Count'])
```

[7]:

	Count
toyota	4720
honda	4570
nissan	3003
chevrolet	2864
ford	2775
hyundai	2502
mazda	1819
dodge	1682
volkswagen	1676
mercedes-benz	1592
acura	1269

```
jeep            1189
saturn          1168
bmw             1113
pontiac         1086
subaru          1015
lexus            935
infiniti         840
gmc              831
chrysler         751
scion            700
kia              699
audi             637
mitsubishi       636
cadillac         510
buick            504
suzuki           421
volvo            402
mini             196
smart            183
```

1.2.3 Top 20 word in all comments (Documents)

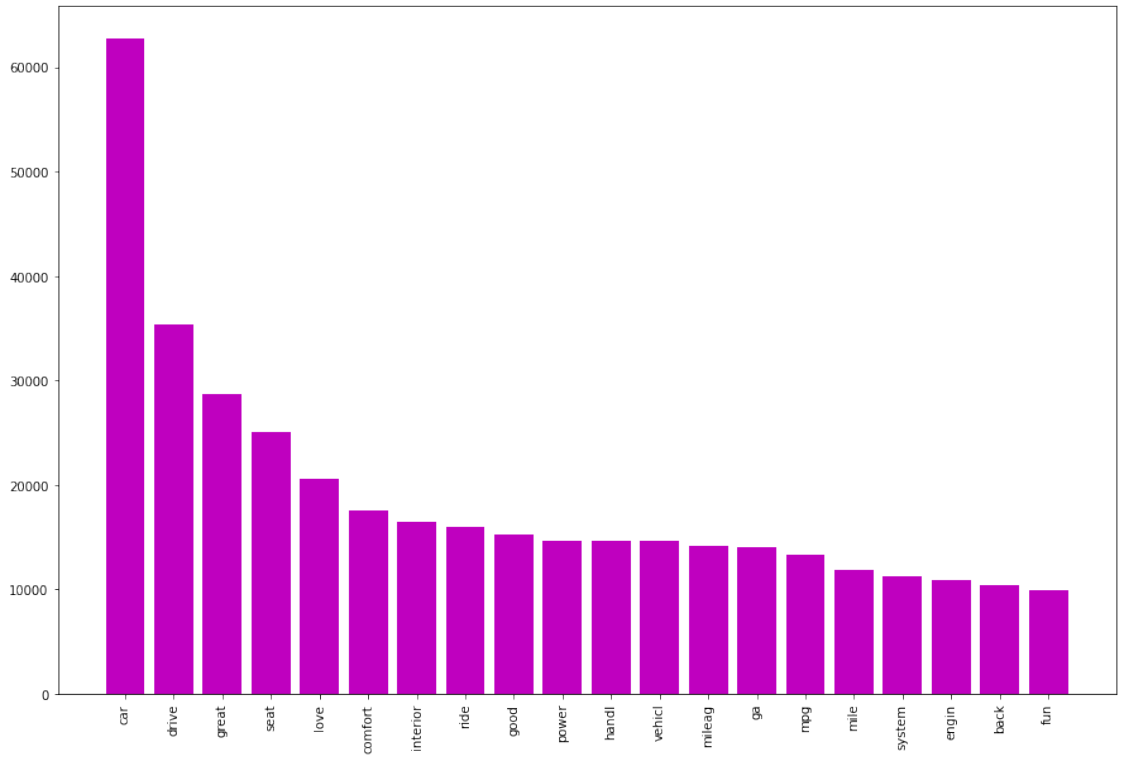
[8]:
```
all_freq = []
for w in invindx:
    total_tf = 0
    for tf in invindx[w]['DocTF']:
        total_tf += int(invindx[w]['DocTF'][tf])
    total_tf = [total_tf,w]
    all_freq.append(total_tf)


all_freq = pd.DataFrame(all_freq, columns=['count', 'word'])

all_freq = all_freq.sort_values(by='count', ascending=False, ignore_index=True)

top_20_word = all_freq.head(20)
top_20_word
```

[8]:

	count	word
0	62707	car
1	35334	drive
2	28701	great
3	25043	seat
4	20560	love
5	17555	comfort
6	16432	interior

```
7   16012      ride
8   15277      good
9   14671     power
10  14631     handl
11  14604    vehicl
12  14153    mileag
13  14105        ga
14  13333       mpg
15  11922      mile
16  11247    system
17  10884     engin
18  10370      back
19   9993       fun
```

[9]:
```
plt.figure(figsize=(15,10))
plt.bar(top_20_word['word'], top_20_word['count'], color='m')
plt.xticks(rotation='vertical')
plt.show()
```



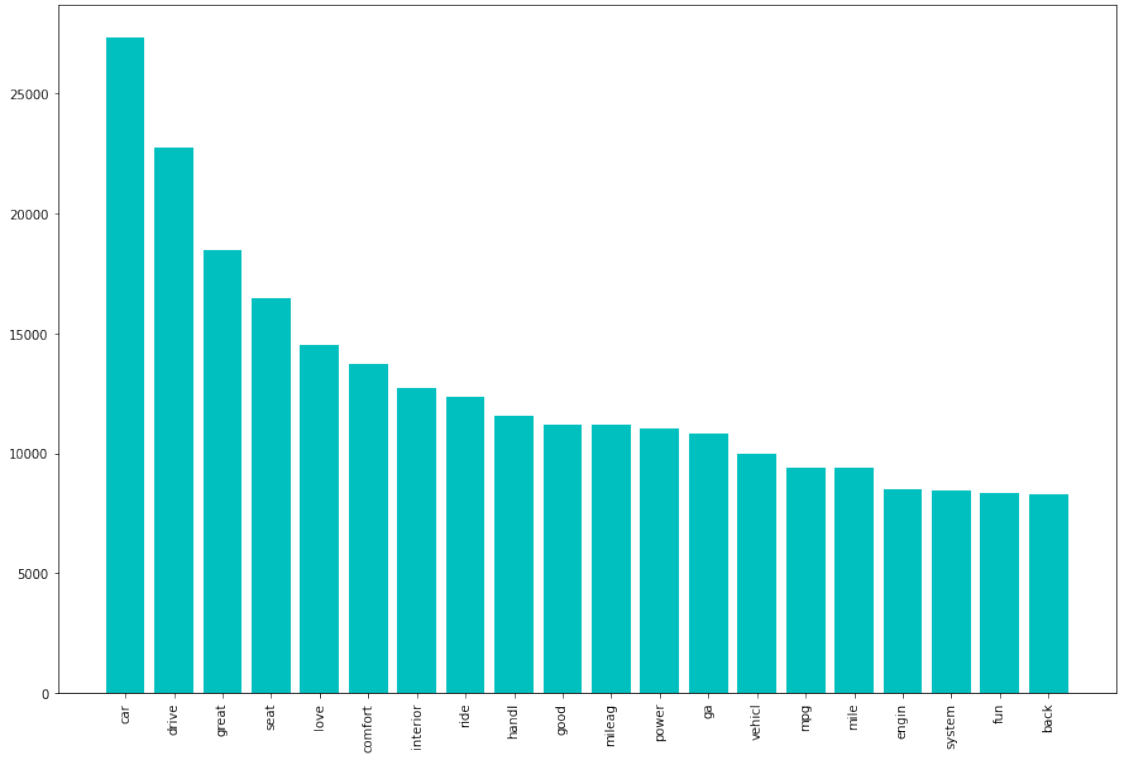
1.2.4 Top 20 Words in Comments (Documents)

[10]:
```
# print(invindx['move'])
all_df = []
total_df = []
for w in invindx:
    df = invindx[w]['DF']
    total_df = [w,df]
    all_df.append(total_df)

all_df = pd.DataFrame(all_df, columns=['word', 'count'])
all_df = all_df.sort_values(by='count' , ascending=False, ignore_index=True)

top_20_df_word = all_df.head(20)
top_20_df_word
```

[10]:

	word	count
0	car	27328
1	drive	22775
2	great	18463
3	seat	16489
4	love	14501
5	comfort	13750
6	interior	12745
7	ride	12360
8	handl	11555
9	good	11200
10	mileag	11186
11	power	11048
12	ga	10824
13	vehicl	9971
14	mpg	9411
15	mile	9405
16	engin	8501
17	system	8452
18	fun	8330
19	back	8313

[11]:
```
plt.figure(figsize=(15,10))
plt.bar(top_20_df_word['word'], top_20_df_word['count'], color='c')
plt.xticks(rotation='vertical')
plt.show()
```


25000
20000
15000
10000
5000
0
car
drive
great
seat
love
comfort
interior
ride
handl
good
mileag
power
ga
vehicl
mpg
mile
engin
system
fun
back