13_Word_Frequency_&_Vector

August 20, 2017

In [1]:
```
import os
from string import punctuation
import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import RegexpTokenizer
from nltk.stem.wordnet import WordNetLemmatizer
```

In [2]:
```
df_temp = pd.read_excel('2_foidev_column_full_list_standardized.xlsx', sheetname='foid
temp = list(df_temp['Standardized term'])
temp = list(set(temp))
temp.remove(np.nan)
temp.remove('UNK')
manufacturer = []
for i in temp:
    i = i.upper()
    i = i.replace(' CORP.', '').replace(' CORPRATION', '').replace(' LTD', '')\
        .replace(' INC.', '').replace(', INC.', '').replace(' INCORPORATED', '').strip
    manufacturer.append(i)
    print(i)
del df_temp, temp
```

```
HUTCHISON INTERNATIONAL
MEDICAL ENGINEERING CORPORATION
ALLERGAN
BAXTER HEALTHCARE
NAGOR
MEDICAL ENGINEERING
COX-UPHOFF INTERNATIONAL
PMT
MCGHAN
NATURAL Y SURGICAL SPECIALTIES
ALLEGIANCE HEALTHCARE
EUROMED
BIOPLASTY
DOW CORNING
MENTOR
```

```
SILIMED
SIENTRA
POLY IMPLANT PROTHESE
INAMED
BIOSIL
IDEAL IMPLANT
```

```
In [3]: fill_type = ['SALINE', 'SILICONE', 'GEL', 'COHESIVE']
        surface_type = ['SMOOTH', 'TEXTURED', 'BIOCELL', 'MICROCELL', 'POLYURETHANE']
        implantation_indication = ['AUGMENTATION', 'RECONSTRUCTION', 'COSMETIC', 'REVISION']
        ALCL = ['ALCL', 'ANAPLASTIC LARGE CELL LYMPHOMA', 'LYMPHOMA', 'T-CELL LYMPHOMA', 'B-CE
                'CANCER', 'TUMOR', 'SUSPECT', 'CONFIRM']
        side = ['LEFT', 'RIGHT', 'BILATERAL', 'BOTH SIDES']
        biomarker = ['CD 30', 'ALK', 'NEGATIVE', 'POSITIVE', 'CD30-', 'CD30+', 'ALK-', 'ALK+']
        symptom = ['breast pain', 'breast swelling', 'breast cyst', 'breast calcification', 'ca
                   'lymph node enlargement', 'firmness of breast', 'hematoma', 'mass', 'lump',
                   'infection', 'abscess', 'leukopenia', 'nodules', 'skin discoloration', 'skin
                   'effusion', 'fluid', 'edema', 'leak', 'redness', 'tenderness', 'erythema',

In [4]: df_BI_DEV = pd.read_csv('WF/DEV_BI_FULL_TABLE.txt', sep='|', header=0, encoding='ISO-88
        df_BI_DEV = df_BI_DEV.drop_duplicates()
        df_BI_TEXT = pd.read_csv('WF/TEXT_FULL_BI_LIST.txt', sep='|', header=0, encoding='ISO-8
        df_BI_TEXT = df_BI_TEXT.drop_duplicates('MDR_TEXT_KEY')
        print('Device #:', df_BI_DEV.shape[0])
        print('Text #:', df_BI_TEXT.shape[0])
        df_BI = df_BI_TEXT.merge(df_BI_DEV, on='MDR_REPORT_KEY', how='inner')
        df_BI['MANUFACTURER_D_NAME'] = df_BI['MANUFACTURER_D_NAME'].astype(str)
        df_BI['BRAND_NAME'] = df_BI['BRAND_NAME'].astype(str)
        df_BI['GENERIC_NAME'] = df_BI['GENERIC_NAME'].astype(str)
        df_BI['FOI_TEXT'] = df_BI['FOI_TEXT'].astype(str)
        print('Full table #:', df_BI.shape[0])

Device #: 28182
Text #: 27137
Full table #: 28301

In [13]: def text_search(text, k):
             if k in text:
                 return True
             else:
                 return False

         def filter_merge(f1, f2, f3):
             return (f1 | f2 | f3)

In [14]: print('Manufacturer word frequecy:\n')
         for m in manufacturer:
```

```
            df_BI['filter'] = np.vectorize(text_search)(df_BI['MANUFACTURER_D_NAME'], m)
            temp = df_BI.loc[df_BI['filter']==True, :]
            del df_BI['filter']
            temp.to_csv('WF_FULL_TABLE/manufacturer/'+m+'.txt', header=True, sep='|', index=Fa
            print(m, ':', temp.shape[0])
            del temp
```

```
Manufacturer word frequecy:

HUTCHISON INTERNATIONAL : 2
MEDICAL ENGINEERING CORPORATION : 3
ALLERGAN : 2655
BAXTER HEALTHCARE : 351
NAGOR : 4
MEDICAL ENGINEERING : 1542
COX-UPHOFF INTERNATIONAL : 2
PMT : 22
MCGHAN : 2184
NATURAL Y SURGICAL SPECIALTIES : 1
ALLEGIANCE HEALTHCARE : 14
EUROMED : 1
BIOPLASTY : 23
DOW CORNING : 3630
MENTOR : 3536
SILIMED : 14
SIENTRA : 46
POLY IMPLANT PROTHESE : 3
INAMED : 234
BIOSIL : 1760
IDEAL IMPLANT : 57
```

```
In [15]: print('Surface type word frequecy:\n')
         for s in surface_type:
             df_BI['filter_1'] = np.vectorize(text_search)(df_BI['BRAND_NAME'], s)
             df_BI['filter_2'] = np.vectorize(text_search)(df_BI['GENERIC_NAME'], s)
             df_BI['filter_3'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], s)
             df_BI['filter'] = np.vectorize(filter_merge)(df_BI['filter_1'], df_BI['filter_2']
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter'], df_BI['filter_1'], df_BI['filter_2'], df_BI['filter_3']
             temp.to_csv('WF_FULL_TABLE/surface/'+s+'.txt', header=True, sep='|', index=False)
             print(s, ':', temp.shape[0])
             del temp
```

```
Surface type word frequecy:

SMOOTH : 1056
TEXTURED : 821
```

```
BIOCELL : 80
MICROCELL : 1
POLYURETHANE : 391
```

```
In [16]: print('Fill type word frequecy:\n')
         for f in fill_type:
             df_BI['filter_1'] = np.vectorize(text_search)(df_BI['BRAND_NAME'], f)
             df_BI['filter_2'] = np.vectorize(text_search)(df_BI['GENERIC_NAME'], f)
             df_BI['filter_3'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], f)
             df_BI['filter'] = np.vectorize(filter_merge)(df_BI['filter_1'], df_BI['filter_2']
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter'], df_BI['filter_1'], df_BI['filter_2'], df_BI['filter_3']
             temp.to_csv('WF_FULL_TABLE/fill/'+f+'.txt', header=True, sep='|', index=False)
             print(f, ':', temp.shape[0])
             del temp
```

```
Fill type word frequecy:

SALINE : 8322
SILICONE : 11394
GEL : 11466
COHESIVE : 364
```

```
In [17]: print('Implantation indication word frequecy:\n')
         for i in implantation_indication:
             df_BI['filter'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], i)
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter']
             temp.to_csv('WF_FULL_TABLE/implantation_indication/'+i+'.txt', header=True, sep='
             print(i, ':', temp.shape[0])
             del temp
```

```
Implantation indication word frequecy:

AUGMENTATION : 1985
RECONSTRUCTION : 1108
COSMETIC : 267
REVISION : 344
```

```
In [18]: print('ALCL word frequecy:\n')
         for m in ALCL:
             df_BI['filter'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], m)
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter']
             temp.to_csv('WF_FULL_TABLE/ALCL/'+m+'.txt', header=True, sep='|', index=False)
             print(m, ':', temp.shape[0])
             del temp
```

```
ALCL word frequecy:

ALCL : 907
ANAPLASTIC LARGE CELL LYMPHOMA : 492
LYMPHOMA : 1072
T-CELL LYMPHOMA : 74
B-CELL LYMPHOMA : 1
CANCER : 867
TUMOR : 167
SUSPECT : 447
CONFIRM : 732
```

```
In [19]: print('Side word frequecy:\n')
         for s in side:
             df_BI['filter'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], s)
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter']
             temp.to_csv('WF_FULL_TABLE/side/'+s+'.txt', header=True, sep='|', index=False)
             print(s, ':', temp.shape[0])
             del temp
```

```
Side word frequecy:

LEFT : 6952
RIGHT : 7041
BILATERAL : 7314
BOTH SIDES : 98
```

```
In [20]: print('Biomarker word frequecy:\n')
         for b in biomarker:
             df_BI['filter'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], b)
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter']
             temp.to_csv('WF_FULL_TABLE/biomarker/'+b+'.txt', header=True, sep='|', index=Fals
             print(b, ':', temp.shape[0])
             del temp
```

```
Biomarker word frequecy:

CD 30 : 7
ALK : 432
NEGATIVE : 359
POSITIVE : 569
CD30- : 8
CD30+ : 86
ALK- : 113
```

```
ALK+ : 0
```

```
In [21]: print('Symptom word frequecy:\n')
         for s in symptom:
             s = s.upper()
             df_BI['filter'] = np.vectorize(text_search)(df_BI['FOI_TEXT'], s)
             temp = df_BI.loc[df_BI['filter']==True, :]
             del df_BI['filter']
             temp.to_csv('WF_FULL_TABLE/symptom/'+s+'.txt', header=True, sep='|', index=False)
             print(s, ':', temp.shape[0])
             del temp
```

```
Symptom word frequecy:

BREAST PAIN : 713
BREAST SWELLING : 33
BREAST CYST : 6
BREAST CALCIFICATION : 2
CAPSULAR CONTRACTURE : 3104
LYMPH NODE ENLARGEMENT : 3
FIRMNESS OF BREAST : 5
HEMATOMA : 795
MASS : 624
LUMP : 713
RUPTURE : 6918
DEFLATED : 1186
INFECTION : 2003
ABSCESS : 47
LEUKOPENIA : 2
NODULES : 84
SKIN DISCOLORATION : 13
SKIN LESION : 16
SEROMA : 1141
EFFUSION : 45
FLUID : 684
EDEMA : 85
LEAK : 2053
REDNESS : 210
TENDERNESS : 377
ERYTHEMA : 94
ASYMMETRY : 688
BREAST ENLARGEMENT : 10
```

```
In [24]: full_list = {'manufacturer':manufacturer, 'fill':fill_type, 'surface':surface_type,
                      'implantation_indication':implantation_indication, 'ALCL':ALCL, 'side':si
                      'biomarker':biomarker, 'symptom':symptom}
```

```
df_vector = pd.DataFrame()
df_vector['MDR_REPORT_KEY'] = df_BI['MDR_REPORT_KEY']

for key, value in full_list.items():
    for w in value:
        temp = pd.read_csv(os.path.join('WF_FULL_TABLE', key, w+'.txt'), sep='|', hea
                           encoding='ISO-8859-1', error_bad_lines=False)
        key_list = list(temp['MDR_REPORT_KEY'])
        del temp
        for k in key_list:
            df_vector.loc[df_vector['MDR_REPORT_KEY']==k, w] = 1
        try:
            df_vector[w] = df_vector[w].fillna(0)
        except:
            continue

for c in df_vector.columns.values:
    df_vector[c] = df_vector[c].astype(int)

df_vector.to_csv('keyword_vector.txt', sep='|', header=True, index=False)
df_vector.head(20)
```

Out[24]:

	MDR_REPORT_KEY	HUTCHISON INTERNATIONAL	MEDICAL ENGINEERING CORPORATION	\
0	6730886	0	0	
1	6730886	0	0	
2	6734192	0	0	
3	6734192	0	0	
4	6283766	0	0	
5	6283766	0	0	
6	6533466	0	0	
7	6533466	0	0	
8	6315557	0	0	
9	6315557	0	0	
10	6747770	0	0	
11	6747770	0	0	
12	6739134	0	0	
13	6739134	0	0	
14	6749011	0	0	
15	6749011	0	0	
16	6748046	0	0	
17	6748046	0	0	
18	6275181	0	0	
19	6275181	0	0	

	ALLERGAN	BAXTER HEALTHCARE	NAGOR	MEDICAL ENGINEERING	\
0	1	0	0	0	
1	1	0	0	0	

2	1	0	0	0
3	1	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	1	0	0	0
9	1	0	0	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	0	0	0	0
15	0	0	0	0
16	1	0	0	0
17	1	0	0	0
18	1	0	0	0
19	1	0	0	0

	COX-UPHOFF	INTERNATIONAL	PMT	MCGHAN	...	seroma	\
0		0	0	0	...	1	
1		0	0	0	...	1	
2		0	0	0	...	1	
3		0	0	0	...	1	
4		0	0	0	...	0	
5		0	0	0	...	0	
6		0	0	0	...	0	
7		0	0	0	...	0	
8		0	0	0	...	1	
9		0	0	0	...	1	
10		0	0	0	...	1	
11		0	0	0	...	1	
12		0	0	0	...	1	
13		0	0	0	...	1	
14		0	0	0	...	0	
15		0	0	0	...	0	
16		0	0	0	...	1	
17		0	0	0	...	1	
18		0	0	0	...	0	
19		0	0	0	...	0	

	effusion	fluid	edema	leak	redness	tenderness	erythema	asymmetry	\
0	0	0	0	1	0	0	0	1	
1	0	0	0	1	0	0	0	1	
2	0	0	0	0	0	0	0	1	
3	0	0	0	0	0	0	0	1	
4	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	

```
6          0      0      0      0      0          0          0          0
7          0      0      0      0      0          0          0          0
8          0      1      0      0      0          0          0          0
9          0      1      0      0      0          0          0          0
10         0      0      0      0      0          0          0          1
11         0      0      0      0      0          0          0          1
12         0      0      0      0      0          0          0          1
13         0      0      0      0      0          0          0          1
14         0      0      0      1      0          0          0          0
15         0      0      0      1      0          0          0          0
16         0      0      0      0      0          0          0          1
17         0      0      0      0      0          0          0          1
18         0      0      0      0      0          0          0          0
19         0      0      0      0      0          0          0          0
```

```
    breast enlargement
0                    0
1                    0
2                    0
3                    0
4                    0
5                    0
6                    0
7                    0
8                    0
9                    0
10                   0
11                   0
12                   0
13                   0
14                   0
15                   0
16                   0
17                   0
18                   0
19                   0

[20 rows x 83 columns]
```

In [ ]: