

HW2

September 4, 2016

```
In [1]: %pylab inline

Populating the interactive namespace from numpy and matplotlib

In [2]: df = open("cacm.trec.filtered.txt")
        data = df.readlines()

In [3]: corpus = [_split() for _ in data]

In [4]: wordlist = unique(list(flatten([_split() for _ in data])))

        total of 88700 words and 1845 unique words

In [6]: import itertools
        all_word_combos = list(itertools.combinations(wordlist,2))

In [7]: from tqdm import tqdm
        dict_counts = {}
        for wordpair in tqdm(all_word_combos):
            for doc in corpus:
                if (wordpair[0] in doc and wordpair[1] in doc ):
                    if wordpair in dict_counts:
                        dict_counts[wordpair]+=1
                    else:
                        dict_counts[wordpair]=1

In [8]: top10keys = sorted(dict_counts, key=dict_counts.get, reverse=True)[:10]

In [18]: for i in top10keys:
         print str(i) + ":" + str(dict_counts[i])

('january', 'paper'):181
('language', 'programming'):153
('january', 'time'):150
('january', 'program'):149
('january', 'systems'):149
('data', 'january'):142
('january', 'presented'):141
('january', 'programming'):139
('program', 'programs'):133
('january', 'method'):125
```

1 3c) Mutual information

```
In [9]: X = wordpair[0]
        Y = wordpair[1]
```

```
In [10]: all_words = list(flatten([_.split() for _ in data]))
```

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

```
In [11]: def p(x):
        return (all_words.count(X)+0.5)/(len(all_words)+1.)
```

```
In [12]: p(X)*p(Y)
```

```
Out[12]: 4.3499716321668095e-08
```

```
In [13]: wordpair
```

```
Out[13]: ('yields', 'zeros')
```

```
In [14]: MI_dict = {}
```

```
In [15]: def pAB(wp):
        return dict_counts[wp]+0.25/(len(all_words)+1)
```

```
In [16]: for wp in tqdm(dict_counts):
        MI_dict[wp] = pAB(wp)*log(pAB(wp)/p(X)*p(Y))
```

```
In [19]: top10MI = sorted(MI_dict, key=MI_dict.get, reverse=True)[:10]
```

```
In [21]: for i in top10MI:
        print str(i) + ":" + str(MI_dict[i])
```

```
('january', 'paper'):940.927980129
('language', 'programming'):769.65701897
('january', 'time'):751.595311055
('january', 'program'):745.588016508
('january', 'systems'):745.588016508
('data', 'january'):703.727458966
('january', 'presented'):697.77516131
('january', 'programming'):685.891893431
('program', 'programs'):650.416450655
('january', 'method'):603.53923359
```

```
('january', 'paper'):181 ('language', 'programming'):153 ('january', 'time'):150 ('january', 'program'):149
('january', 'systems'):149 ('data', 'january'):142 ('january', 'presented'):141 ('january', 'programming'):139
('program', 'programs'):133 ('january', 'method'):125
```

i) The top 10 word pairs with the highest mutual information is the same as the top 10 pairs based on the co-occurrence counts.

ii) The top 5 words which have the highest mutual information with the word “programming” are:

```
In [23]: topMIs = sorted(MI_dict, key=MI_dict.get, reverse=True)[:30]
        for i in topMIs:
            if i[0] == 'programming' or i[1] == 'programming':
                print str(i) + ":" + str(MI_dict[i])

('language', 'programming'):769.65701897
('january', 'programming'):685.891893431
('program', 'programming'):586.090583816
('paper', 'programming'):562.940801951
('languages', 'programming'):545.667210954
```

The results are somewhat reasonable, programming language(s), program, programming all have word meaning that makes sense. We saw previously that january is a common word that appears with many other words, (possibly because this is a subset of articles that have January listed in its publication date), so it is not surprising that january also occur frequently with programming. ('paper', 'programming') is a bit surprising, but maybe the word 'paper' occurs frequently in the same phrase common throughout the collection (e.g. 'Paper published on July 1, 2016', 'Paper accepted to CACM journal' ..etc), so any papers containing the word 'programming' will be counted as co-occurrence.