CIS434 Homework 4 - Pin Li

Food Trend

Introduction

In this assignment, the cooccurrence matrix is used to find the food trend. The reason is that as we observe words that co-occur, there would be a high possibility of trend rising. To be specific, a seasonal rise or a rapid rise of cooccurrence will relate to trends. I also tried LDA for this assignment since it would show the trend of certain topics. However, when it comes to discovering the trends of specific ingredients, the words in the topics will seem redundant and increase the computation unnecessarily. Thus, under comparison, cooccurrence is used here.

Summary

In the process, I firstly find the words that usually appear together. To validate, I used 'vegetable noodle' and 'cauliflower rice' to validate. As we could observe from the picture below, pumpkin pie and pecan pie show high seasonal correlations, which could be explained by holiday servings. Lemon juice and chicken soup show the rapid growth of cooccurrence, which is another kind of trend.

In [ ]:

```
import os
import numpy as np
import pandas as pd
import re
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from tqdm import tqdm
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:

```
def cooccurrence_symmetric_window( sentlist, targets, weights ): # cooccurrence
 based on weighted moving window
    m = len(weights)
    cooc = np.zeros((len(targets), len(targets)), np.float64)
    w2i = {w:i for (i,w) in enumerate(targets,0)}
    for num,sent in enumerate(sentlist,1):
        words = sent.split()
        n = len(words)
        for i in range(n):
            end = min(n-1, i+m)
            for j in range(i+1, end+1):
                if(words[i] in w2i.keys() and words[j] in w2i.keys()): cooc[w2i[
words[i]], w2i[words[j]]] += weights[j-i-1]
    np.fill_diagonal(cooc, 0)
    return cooc+cooc.T # necessary since symmetry is exploited to save computati
on during construction
```

In [6]:

```
# convert list to string
def listToString(s):
    string = ""
    for ele in s:
        string += ele
    return string
```

In [7]:

```
# load facebook data
import csv
file = []
data = []
for j in range(1,6):
    for i in range(1,13):
        file.append(open('/Users/Pin/Desktop/Social Media/Data/facebook/fb201'+s
tr(j)+'/fpost-201'+str(j)+'-'+str(i)+'.csv', 'r'))
for i in range(1,61):
    data.append(file[i-1].readlines())
```

In [8]:

```
# load ingredient data
ingredients = open("/Users/Pin/Desktop/Social Media/Data/facebook/ingredients.tx
t","r")
ingredients = ingredients.readlines()
ingredients = listToString([x.lower() for x in ingredients])
```

In [9]:

```
# ingredients - word tokenization
newing = nltk.word_tokenize(ingredients)
newing = list(set(list(set(newing))))
newing.remove("and")
newing.remove("of")
newing.remove("con")
newing.remove("de")
newing.remove("sel")
```

In [10]:

```
# get the cooccurence matrix
def getCooc(data):
    symbols = listToString([x.lower() for x in data])
    sentencelist = sent_tokenize( symbols )
    sentencelist = [re.sub("[^a-zA-Z]", " ", x) for x in sentencelist]
    cooc = cooccurrence_symmetric_window( sentencelist, newing, 1/np.arange(1,10
))
    return cooc
```

In [11]:

```
cooc = []
for i in tqdm(range(0,60)):
    cooc.append(getCooc(data[i]))
```

```
100%|██████████| 60/60 [16:26<00:00, 16.43s/it]
```

In [12]:

```
#3-D stack
foodlist = np.stack(cooc)
```

In [13]:

```
# save the data
from tempfile import TemporaryFile
outfile = TemporaryFile()
x = foodlist
np.save(outfile, x)
```

In [14]:

```
foodindex = {w: i for (i,w) in enumerate(newing)}
```

In [17]:

```
# define the plot function
def plotTrend(f1,f2):
    fig, ax = plt.subplots(figsize=(15, 5))
    ax.plot(foodlist[:,foodindex[f1],foodindex[f2]])
    ax.set_xticks(np.arange(60,step = 12))
    ax.set_xticklabels(['2011-01','2012-01','2013-01','2014-01','2015-01'], font
size=18)
    ax.set_title('The Trend of '+ f1 +' and '+f2,fontsize=30)
    ax.set_xlabel("Month",fontsize=20)
    ax.set_ylabel("Number",fontsize=20)
```

In [20]:

```
# show the most popular pair of ingredients in each month
def getTrendList(data):
    cooc = getCooc(data)
    m = pd.DataFrame(cooc, index=newing, columns=newing)
    m = m[~np.all(m == 0, axis=1)]
    month = pd.DataFrame(m.stack()).reset_index()
    month.columns = ['ing1','ing2','trend']
    month = month.sort_values(by='trend', ascending = False)[month['trend']>=10]
[0::2]
    month['food'] = month['ing1'] +' ,' + month['ing2']
    month = month.drop(['ing1','ing2'],axis=1)
    return month
```

In [21]:

```
#find the words that is of popular coocurrence
getTrendList(data[10]).head(20)
```

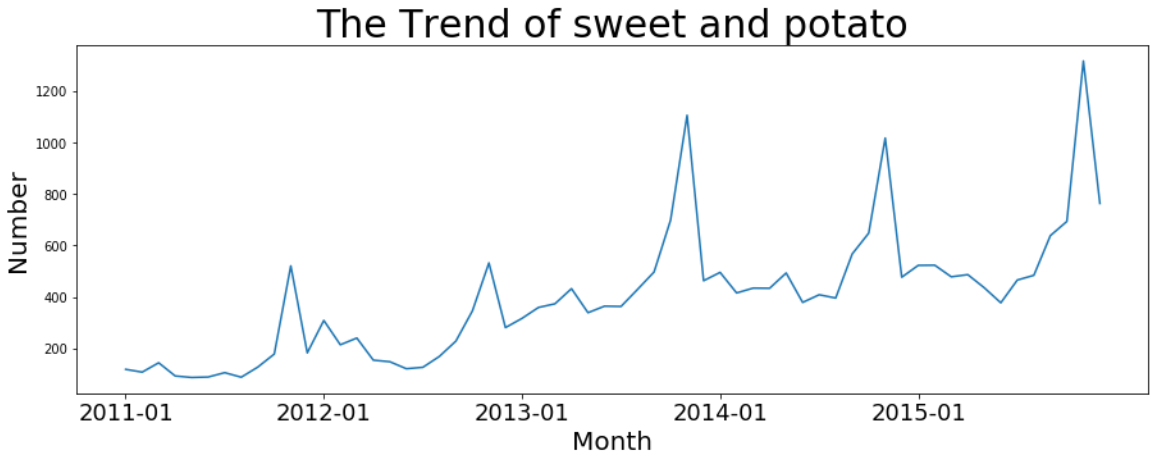
```
//anaconda3/envs/cis434/lib/python3.6/site-packages/ipykernel_launch
er.py:8: UserWarning: Boolean Series key will be reindexed to match
DataFrame index.
```

Out[21]:

	trend	food
462879	520.849206	sweet ,potato
548419	487.855952	pie ,pumpkin
559873	332.441667	mashed ,potatoes
696769	300.919841	potatoes ,sweet
197969	258.395635	cranberry ,sauce
640985	247.999206	cheese ,cream
333184	245.753175	cream ,ice
393171	222.306349	butter ,peanut
499422	190.209524	apple ,pie
308117	186.185714	olive ,oil
115362	184.506746	turkey ,breast
670792	173.638889	leaf ,maple
599649	167.263492	chocolate ,cake
547946	158.780159	pie ,pecan
169835	147.211508	beans ,green
208793	130.441270	squash ,butternut
176293	122.706349	sugar ,brown
598723	115.958333	soup ,chicken
574266	103.149603	pepper ,salt
700213	96.406349	juice ,lemon

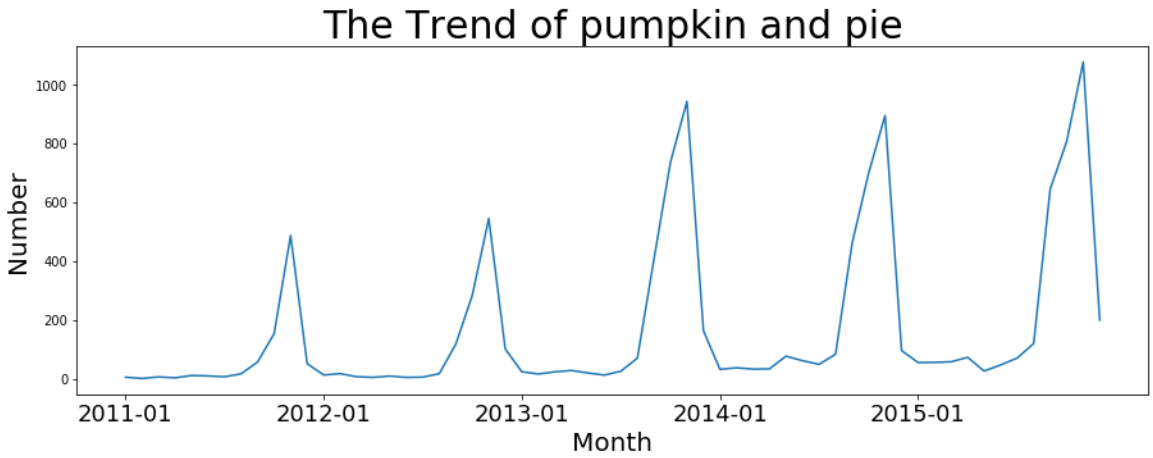
In [22]:

```
plotTrend('sweet','potato')
```



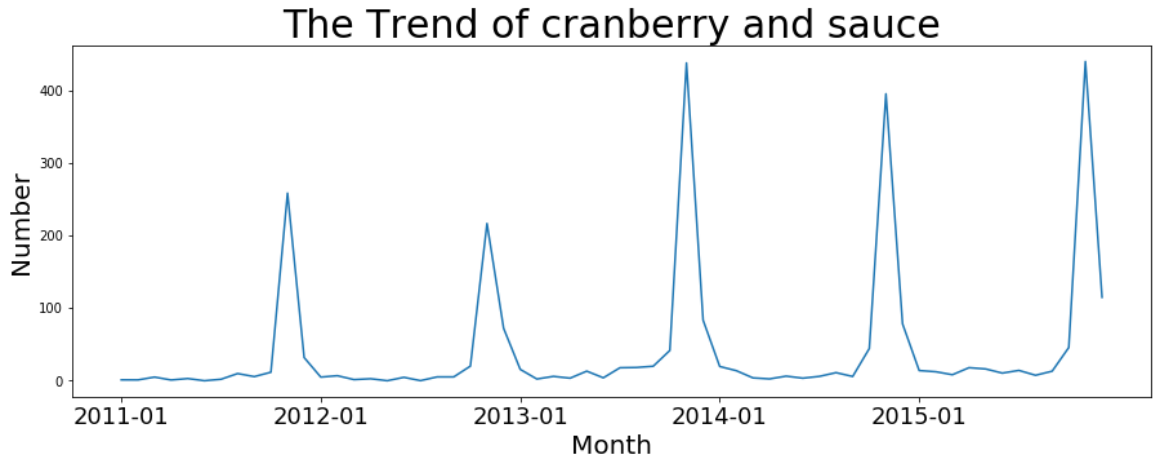
In [23]:

```
plotTrend('pumpkin','pie')
```



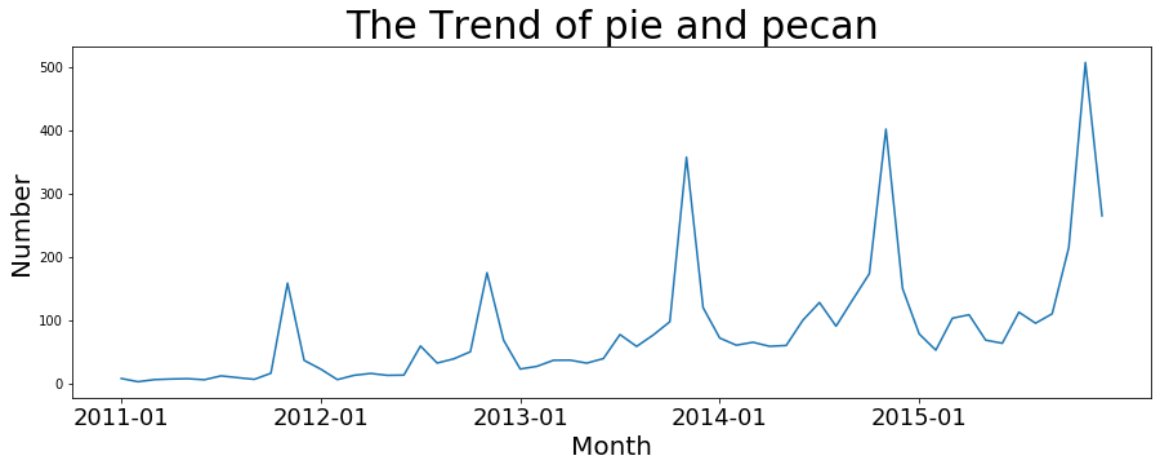
In [24]:

```
plotTrend('cranberry','sauce')
```



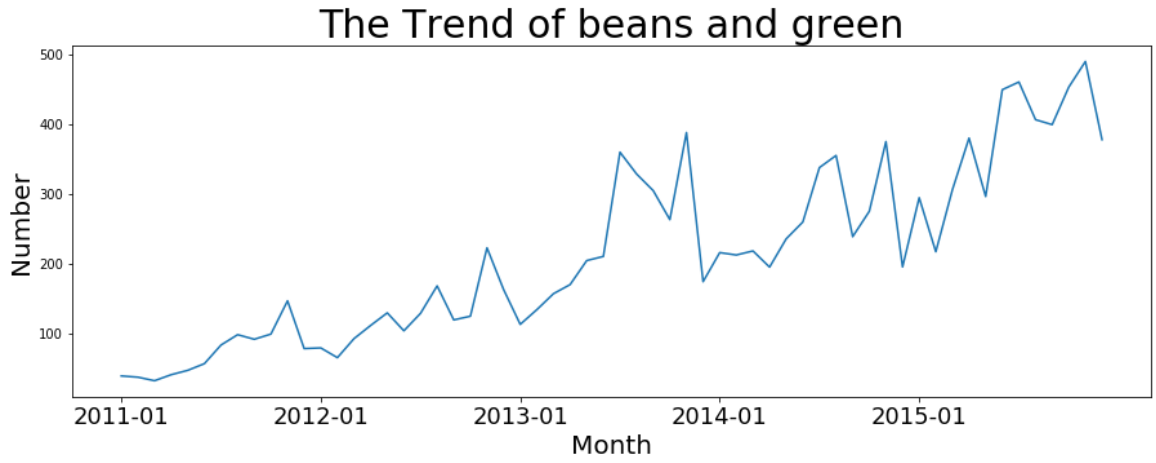
In [25]:

```
plotTrend('pie','pecan')
```



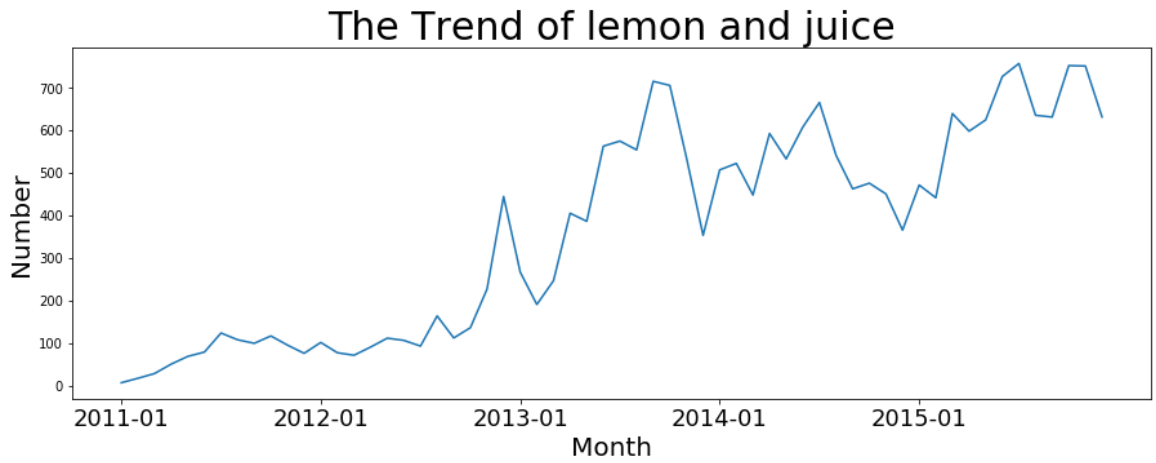
In [26]:

```
plotTrend('beans','green')
```



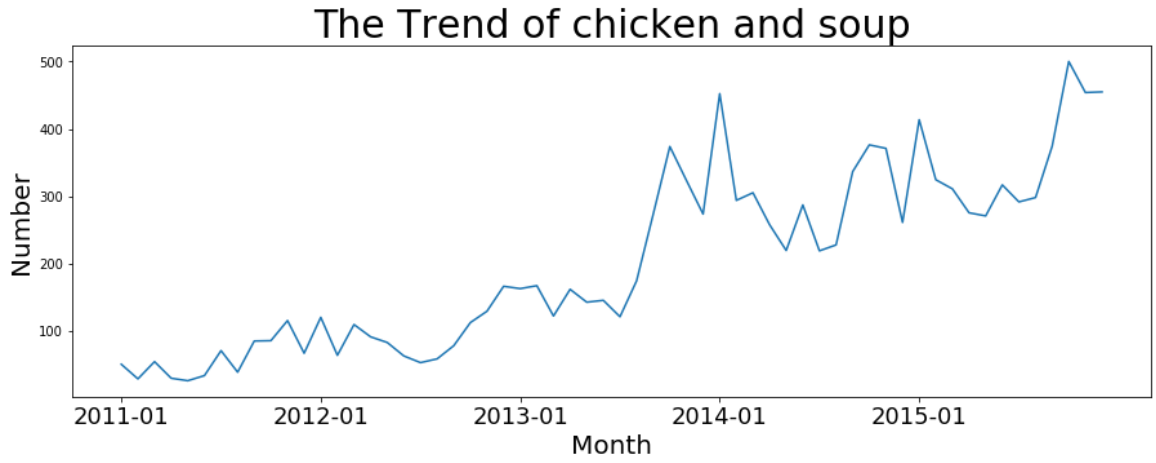
In [27]:

```
plotTrend('lemon','juice')
```



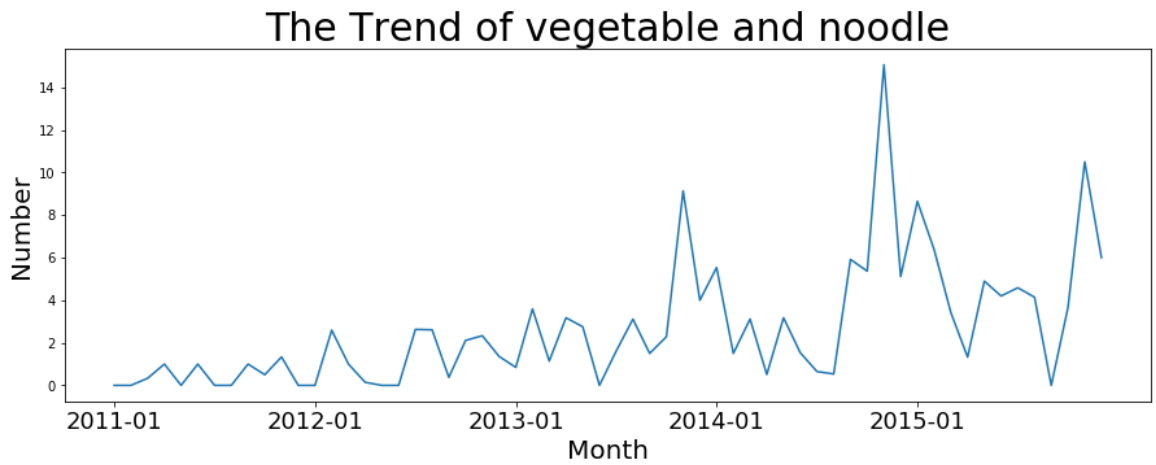
In [30]:

```
plotTrend('chicken','soup')
```



In [28]:

```
plotTrend('vegetable','noodle')
```



In [29]:

```
plotTrend('cauliflower','rice')
```