sentence selection

March 20, 2019

In [1]:
```
import re
from pandas import Series, DataFrame
import pandas as pd
from numpy.random import randn
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import os
import rake_nltk
from rake_nltk import Rake
import nltk
# from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

# use wordnet find the first level keywords
from nltk.corpus import wordnet as wn
```

0.1 Load data.

0.1.1 Merge sample homes and their reviews into one dataframe.

In [2]:
```
# Load Necessary Data: reviews
reviews_df = pd.read_csv("./Data/reviews.csv", encoding="utf-8")
reviews_df.columns = ['home_id', 'review_id', 'date', 'reviewer_id', 'reviewer_name',
reviews_df.dropna()
reviews_df.head(2)
```

Out[2]:

	home_id	review_id	date	reviewer_id	reviewer_name	\
0	7202016	38917982	2015-07-19	28943674	Bianca	
1	7202016	39087409	2015-07-20	32440555	Frank	

	comments
0	Cute and cozy place. Perfect location to every...
1	Kelly has a great room in a very central locat...

In [3]:
```
sample1_df = pd.read_csv("./Data/sample_data_for_testing", sep='\t', encoding="utf-8")
sample1_df = sample1_df.drop("Unnamed: 0", axis=1)
sample1_df.head(2)
```

Out[3]:

	home_id	property_type	room_type	price	number_of_reviews	scores_overall_rating	scores_accuracy	scores_cleanliness	scores_checkin	scores_communication	scores_location	scores_value	price_int
0	241032	Apartment	Entire home/apt	$85.00	207	95.0	10.0	10.0	10.0	10.0	9.0	10.0	85
1	953595	Apartment	Entire home/apt	$150.00	43	96.0	10.0	10.0	10.0	10.0	10.0	10.0	150

In [4]:
```
# Merge the reviews and homes in the sample data.
df1 = sample1_df[['home_id', 'scores_cleanliness']]
df2 = reviews_df[['home_id', 'review_id', 'comments']]

print("-" * 40
      + '\nTotal number of reviews: '
      + str(sample1_df['number_of_reviews'].sum())
      + "\n" + "-" * 40)

sample1_rh_df = pd.merge(df1, df2, on="home_id")
sample1_rh_df.head(3)

# sample1_rh_df.stack()[0].comments
```

```
----------------------------------------
Total number of reviews: 68638
----------------------------------------
```

Out[4]:

	home_id	scores_cleanliness	review_id	comments
0	241032	10.0	682061	Excellent all the way around. \r\n\r\nMaija wa...
1	241032	10.0	691712	Maija's apartment was a wonderful place to sta...
2	241032	10.0	702999	one of the most pleasant stays i've had in my ...

0.2 Analyze `cleanliness` aspect

0.2.1 Use the selected aspect keywords to analyze reviews

0.2.2 Brief summary

In [5]:
```
# sample2_rh_df is a copy of sample1_rh_df to
# in case unexpected modification for original data.
sample2_rh_df = sample1_rh_df
print('*' * 40 + '\nThere are:\n' + '-' * 40)
print(str(len(sample2_rh_df.groupby('home_id'))) + " Airbnb homes in total.\n" + '-' *
print(str(len(sample2_rh_df)) + " reviews in total.\n" + '-' * 40)
```

```
****************************************
There are:
----------------------------------------
2225 Airbnb homes in total.
----------------------------------------
68638 reviews in total.
----------------------------------------
```

In [6]:
```
# group by scores_cleanliness
df1 = pd.DataFrame(sample2_rh_df.groupby(['scores_cleanliness'])['home_id'].nunique())
df2 = pd.DataFrame(sample2_rh_df.groupby(['scores_cleanliness'])['review_id'].nunique()
summary_df = pd.merge(df1, df2, on = 'scores_cleanliness')
summary_df.columns = ['number of homes', 'number of reveiws']
summary_df
```

Out[6]:

scores_cleanliness	number of homes	number of reveiws
3.0	1	2
4.0	4	9
5.0	3	17
6.0	23	86
7.0	20	236
8.0	113	2813
9.0	512	16571
10.0	1549	48904

0.2.3 Chose 1 cleanliness score to do further analysis.

Here I choose Airbnb homes with cleanliness score as 8 to do the "pliot test".

As in above table, there are 113 homes with 2813 reviews in total.

In [7]:
```
sample2_rh_df[sample2_rh_df.scores_cleanliness == 8.0].groupby(['home_id']).count().hea
```

Out[7]:

home_id	scores_cleanliness	review_id	comments
15108	42	42	42

```
66611                    20          20          20
82763                    12          12          12
233502                    1           1           1
258571                  278         278         278
```

In [8]:
```
# Functions used to do sentence extraction.
import spacy
nlp = spacy.load('en')

def parseSentence(doc):
    return [sent for sent in doc.sents]

def token_text(doc):
    for token in doc:
        print('"' + token.text + '"')

def see_entity(doc):
    for ent in doc.ents:
        print(ent.text, ent.label_)

# Uses stopwords for english from NLTK, and all puntuation characters by default
r = Rake()
r.get_ranked_phrases_with_scores

def keyword_extraction(txtContent):
    r.extract_keywords_from_text(txtContent)
    return r.get_word_degrees()
```

0.2.4 Try a small sample: home 258571 which has 278 number of comments.

In [10]:
```
comment258571 = ''.join(list(sample2_rh_df[sample2_rh_df.home_id == 258571].comments))
doc = nlp(comment258571)

print("-" * 80 +
      "\nThere are in total "
      + str(len(list(doc.sents)))
      + " sentences in these 278 comments.\n"
      + "-" * 80)
```

```
--------------------------------------------------------------------------------
There are in total 1478 sentences in these 278 comments.
--------------------------------------------------------------------------------
```

In [11]:
```
nlp = spacy.load('en_core_web_md')

aspect_keywords_dic = {
```

```
    'location': ['region', 'locality', 'neck_of_the_woods', 'location', 'vicinity',
                 'neighbourhood', 'neighborhood'],
    'cleanliness': ['tidy_up', 'straighten_out', 'cleanliness', 'clean',
                    'neaten', 'square_away', 'straighten', 'clean_house', 'make_clean
                    'tidy', 'houseclean', 'clean_up', 'scavenge',
                   'soiled', 'unclean', 'colly', 'bemire', 'uncleanliness', 'soil', 'l
                    'grime', 'untidy', 'dirty']
}

def similarity_score(doc, aspect):

    similarity_score_dic = {}
    aspect_keywords = nlp(' '.join(aspect_keywords_dic[aspect]))

    # I tried use the total scores of all vectors, however, the results are very stra
    # So, I tried to use the count of the words in a sentance
    # that with lager than 0.7 similarity score to determine the relevance.
    for which_sen in range(len(list(doc.sents))):

        new_doc = list(doc.sents)[which_sen].text
        similarity_score_dic.setdefault(which_sen,{})
        similarity_score_dic[which_sen].setdefault('sentence', new_doc)

        sen_keywords = nlp(new_doc)

        total_score = compute_score(sen_keywords, aspect_keywords)

#         similarity_score_dic[which_sen].setdefault(aspect, score)
        similarity_score_dic[which_sen].setdefault('total_score', total_score)

    return similarity_score_dic

def compute_score(sen_keywords, aspect_keywords):
    count = 0
    for token1 in sen_keywords:
        for token2 in aspect_keywords:
            if token1.similarity(token2) >= 0.6:
                count += 1
    return count
```

0.2.5 Here, the logic of determine the aspect that a sentence talking about is:

`aspect_keywords_dic` **is a dictionary contains aspects and their relevant keywords.**

Firstly, I use the Word2Vec similarity algorithm to count the similarity score between two words.

For each sentence, the aspect-similarity-score will +1 when one word in it has the word-similarity-score with any word in keyword list lager than 0.7. I tried use the total scores of all vectors(word-in-sentence to word-in-keywords), however, the results are very strange.

So, temporarily, I tried to use the count of the words in a sentance that with lager than 0.7 similarity score to determine the relevance. And it works well for now.

```
In [15]: home258571_sent_score = similarity_score(doc, 'cleanliness')
         home258571_cleanliness_text = [home258571_sent_score [k]['sentence'] for k in home258
```

0.2.6 Result summary

```
In [17]: print("-" * 80
               + "\nThere are "
               + str(len(home258571_cleanliness_text))
               + "/"
               + str(len(list(doc.sents)))
               + " sentences talking about the cleanliness aspect.\n"
               + "-" * 80)
```

```
--------------------------------------------------------------------------------
There are 91/1478 sentences talking about the cleanliness aspect.
--------------------------------------------------------------------------------
```

```
In [18]: home258571_cleanliness_text
```

```
Out[18]: ['The apartment was clean and quiet.  ',
          'Great host, comfy, clean room with all you need in the most buzzing part of Seattle
          'Overall, incredibly good value and highly recommended!!The apartment was clean and
          'Thanks!Nick does his best to provide the things you would need for a comfortable st
          'Even though its at the back of the building and the giant living room windows look
          'the kitchen needed a good scrubbing - it was dirty.  ',
          'The apartment was a little on the small side - so probably better for just 2 people
          'Super clean and convenient.',
          'Comfortable, clean and quiet the apartment had just about everything you could ask
          'The apartment was clean and comfortable, and you absolutely cannot beat the locatio
          'The apartment was clean, well furnished and more than enough for what we needed.',
          'His apartment was very cozy, tidy, and space efficient since it was only the two of
          'Apartment was tidy and had everything we needed.',
          'The apartment was clean, private, a perfect home away from home.',
          'It was very clean and comfortable, and located within easy walking distance of the
          'The bathroom was very dirty with dust and hair in the tub and on the floor.   ',
          'Thanks again!The apartment was clean, cozy, well-stocked, and most importantly, ver
          'Apartment is cozy and clean.',
          'The place was clean and comfortable, and the location is ideal for those wanting to
          'The place is quiet and clean.',
```

```
'The apartment was clean and exactly what I was looking for.',
'Clean and wonderful place to stay!',
'Apartment was clean and as advertised.',
'Everything was clean for the most part and the place was quiet/warm enough during ou
'The place was nice, clean, and well-kept, but the best thing about it was the locati
'The apartment was clean and cozy and in a great location.  ',
"Thanks guys!It's good value and a great location (we walked everywhere) and the host
'One downside, however, was that i found the unit not clean enough.  ',
'We have stayed in a few Airbnb accommodations in the last year and in general this p
"Some maintenance issues and a degree of run-downness likely due to the building age
'I stayed here for 2 nights and found it clean, comfortable and location fantastic.'
'great location, cozy, clean, and comfortable.',
'The complex and apartment were a little old and worn, but clean and in a great area
'The good bit is the flat is very clean and in a convenient location to walk everywhe
'Otherwise the place was clean, safe and conveniently located.  ',
'Apartment was clean and not cluttered.',
'The apartment was nice and roomy, clean and well supplied.',
"The apartment is not spotless clean but it's clean enough - for instance, there was
'Cozy bedroom, clean bathroom, good price and perfect location.',
'It was clean, neat and comfortable.',
'The place was really nice and clean..',
"pretty sure the sheets weren't clean which is the worst, we both slept in our clothe
'but I was at least hoping the bed would be clean and not rock hard...',
'The apartment was very clean and Nick was very accomodating!The',
'The apartment was clean and the bed and pillows were comfortable.',
'The apartment was clean and everything worked.  ',
'In all, good value and great location , but a bit dirty and the photos make it seem
'When we checked out we tried to leave it clean.',
'Clean space, on the darker and older side, and the bed is very noisy but overall it
"Apartment was clean and close to the Capital Hill scene!For the location, it's hard
'The clean rooms were ample for 2, the bed was comfortable and great coffee was just
'The apartment was clean, and there is also a well-equipped kitchen.',
'This apartment was clean, and in an awesome location.',
'The appartment was so clean, quiet and nicely fitted out. \n\n',
'Clean and simple place.',
'The apartment is very clean and neat, just as is described.',
'Great and very clean space in a perfect location!  ',
'Clean comfortable apartment.',
'The apartment was clean and had everything I needed.',
'The location is ideal for neat local spots like Melrose market or walking up the Pi
'Bed was extremely comfortable and the apartment very clean.',
'floor and bed cover clean condition is so so. \n',
'The neighborhood is perfectly located, and the apartment was clean and well-furnishe
'I am by no means a "clean freak," but the apartment felt dingy.  ',
'There was hair in the tub, the floor was dusty, and the comforter felt dirty as well
'The apartment was more than adequate for the stay and was clean and comfortable.\r\n
'The apartment was quiet and clean.',
'Older apartment but clean.  ',
```

```
 'Once again, it was clean and comfortable.',
 'The apartment is represented very accurately by the pictures and it is a clean and
 'The bedroom and bathroom were very clean and well taken care of.',
 'It was clean, close to downtown, and met my needs while I was in Seattle.  ',
 'Clean and quiet.',
 'The bathroom, and most importantly the bathtub, seemed clean.  ',
 'Only concern is that the bed was really squeaky, but overall the apartment was clea
 'The apartment was clean and checking in was a smooth process.',
 'As for the apartment, it was a clean and very cozy place.',
 "Also we couldn't find an outlet in the bathroom, so we had to dry/straighten/curl ou
 'The apartment is clean and convenient to the highway.',
 'A few downsides were the lack of an outlet in the restroom and while the couch was
 'The apartment itself was tidy and even though there was a heat wave and no A/C, it
 'Great location and clean.',
 'Place was neat',
 'Great location, clean apartment, simple entry process.',
 'This accommodation was easily accessible and we arrived to a spotlessly clean apartm
 'The property was clean and comfortable.',
 "It was clean overall, but dusty in areas--couch was old & worn, and I couldn't reach
 'The apartment is clean and presents as advertised.',
 'The apartment itself was very clean and nicely furnished.',
 'Clean linens and towels were provided.',
 'The space was very clean and it felt like my own apartment.']
```

```
In [19]: def score_detail(sen_keywords, aspect_keywords):
             score = 0
             for token1 in sen_keywords:
                 for token2 in aspect_keywords:
                     if token1.similarity(token2) >= 0.7:
                         print(token1.text, token2.text, token1.similarity(token2))
                         score += 1
             return score

In [20]: h = nlp(u"Nick was very communicative and gracious enough to transport me to and from
         aspect_keywords = nlp(' '.join(aspect_keywords_dic['cleanliness']))

         # print(len(sen_keywords), len(aspect_keywords))
         score_detail(h, aspect_keywords)

Out[20]: 0

In [21]: def word_similarity(w1, w2):
             token1 = nlp(w1)
             token2 = nlp(w2)
             return token1.similarity(token2)

         word_similarity('scrubbing', 'clean')
```

Out[21]: 0.648041676443703

In [22]:
```
# print("Brief pre-processing logic for one sentence.")
# print("*" * 93 + "\nAn example:\n" + "-" * 80 + "\nThe first sentence:")
# print(list(doc.sents)[0].text)

# print("-" * 93 + "\nSentence keywords got from keyword_extraction function:")
# sen_keywords = list(keyword_extraction(list(doc.sents)[0].text).keys())
# print(sen_keywords)

# print("-" * 93 + "\nJoin sentence keywords together for further similarity use:")
# sen_keywords = ' '.join(sen_keywords)
# print(sen_keywords)
# print("-" * 93)
```

In [ ]: