Amazon Review - Wordcloud Creation

May 21, 2017

Extracting keywords is one of the most important tasks when working with text. Readers benefit from keywords because they can judge more quickly whether the text is worth reading. Website creators benefit from keywords because they can group similar content by its topics. Algorithm programmers benefit from keywords because they reduce the dimensionality of text to the most important features. And these are just some examples;

By definition, keywords describe the main topics expressed in a document.

1 Import Libraries

```
In [1]: import pandas as pd
         import numpy as np
         from nltk.tokenize import RegexpTokenizer
         from nltk.corpus import stopwords
         from stop_words import get_stop_words
         from nltk.stem.snowball import SnowballStemmer
         from gensim import corpora, models
         import gensim
```

2 Import Reviews Data

```
In [2]: df = pd.read_csv('reviews.csv')
         df.columns = ['ID','numDate','prod','overall','helpful','votes',
                       'date','asin','summary','reviewText']

In [3]: toShow=df[['numDate','overall','helpful','date','summary','reviewText']]
         toShow.head()

Out[3]:     numDate  overall  helpful                  date  \
         0  20161030        2        0  on October 30, 2016
         1  20161030        4        0  on October 30, 2016
         2  20161029        5        0  on October 29, 2016
         3  20161028        5        0  on October 28, 2016
         4  20161027        5        0  on October 27, 2016

                                                summary  \
         0  but product arrived in excellent condition.
         1                        Not fun to put together
         2                           Get This Cool Trike!
         3                                       Love it!
         4                                    Sweet ride.
```

```
                                          reviewText
0  whobbly to ride.  going around corners VERY ca...
1  Not fun to put together! You have to do a lot ...
2  All Senior Citizens need to own this trike!  V...
3  Needed metric tools to put it together, but on...
4  We bought this, added a boat seat, 80cc gas mo...
```

3 Keyword extraction with Python using RAKE

A typical keyword extraction algorithm has three main components:

1. Candidate selection: extract all possible words, phrases, terms or concepts (depending on the task) that can potentially be keywords.

2. Properties calculation: For each candidate, calculate properties that indicate that it may be a keyword.

3. Scoring and selecting keywords: All candidates can be scored by either combining the properties into a formula, or using a machine learning technique to determine probability of a candidate being a keyword. A score or probability threshold, or a limit on the number of keywords is then used to select the final set of keywords.

Finally, parameters such as the minimum frequency of a candidate, its minimum and maximum length in words, or the stemmer used to normalize the candidates help tweak the algorithm's performance to a specific dataset.

The following algorithm is described in the Text Mining Applications and Theory book by Michael W. Berry:

```
In [4]: import rake
         import operator

         rake_object = rake.Rake("SmartStoplist.txt", 8, 2, 10) #528
         # "SmartStoplist.txt" // "FoxStoplist.txt" // "FrenchStoplist.txt"
         # Now, we have a RAKE object that extracts keywords where:
         # Each word has at least A characters
         # Each phrase has at most B words
         # Each keyword appears in the text at least C times

         text = df["reviewText"].str.cat(sep='\n')
         keywords = rake_object.run(text)
         print("Keywords:", [x[0] for x in keywords])
```

```
Keywords: ['great bike', 'missing parts', 'front fender', 'balance issues',
        'handle bars', 'front tire', 'front brake','fairly easy','nice bike',
        'bike shop', 'wheel bike', 'rear fenders','regular bike','back wheels',
        'highly recommend', 'assembly instructions', 'shipping', 'delivery',
        'excellent', 'assembly', 'comfortable', 'steering', 'christmas',
         'assembled', 'problems', 'recommend', 'tricycle', 'assembling',
         'instructions', 'delivered', 'stability', 'neighborhood', 'daughter',
         'transport', 'scratches', 'difficult', 'exercise','purchase',
         'shopping','directions', 'received', 'replaced','purchased',
         'handlebars','surprised', 'returned', 'assemble','expected',...]
```

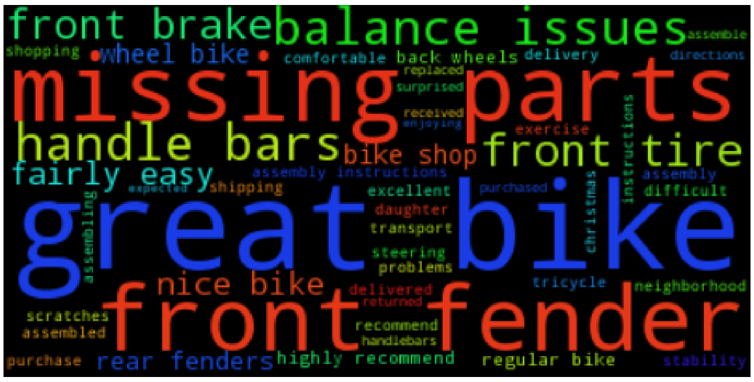
4 Wordcloud Creation - v1

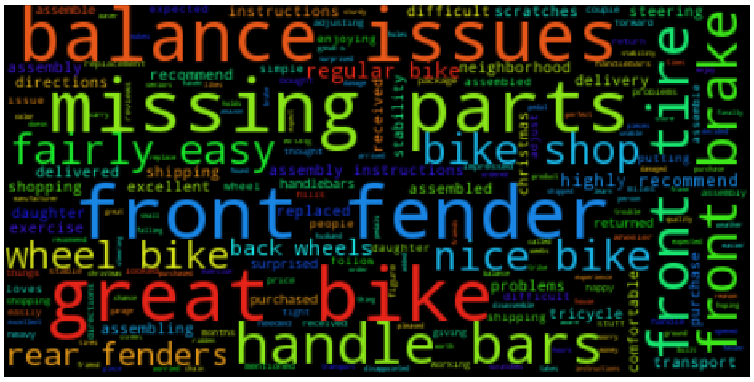
```
In [5]: from os import path
         from wordcloud import WordCloud
         # Generate a word cloud image
         wordcloud = WordCloud().generate_from_frequencies(keywords)

         # Display the generated image:
         # the matplotlib way:
         import matplotlib.pyplot as plt
         plt.imshow(wordcloud)
         plt.axis("off")

         # lower max_font_size
         wordcloud = WordCloud(max_font_size=40).
         generate_from_frequencies(keywords)

         plt.figure()
         plt.imshow(wordcloud)
         plt.axis("off")
         plt.show()
```





5 Wordcloud Creation - v2

```
In [6]: import numpy as np
         from PIL import Image
         from os import path
         import matplotlib.pyplot as plt
         import random
```

```
from wordcloud import WordCloud, STOPWORDS


def grey_color_func(word, font_size, position, orientation,
random_state=None,
                    **kwargs):
    return "hsl(0, 0%%, %d%%)" % random.randint(60, 100)

import os
d=os.getcwd()
dd=path.join(d, "mask1.png")

# read the mask image
mask = np.array(Image.open(dd))

# adding movie script specific stopwords
stopwords = set(STOPWORDS)
stopwords=open("SmartStoplist.txt","r")
#stopwords.add("int")
#stopwords.add("ext")

wc = WordCloud(width=1920, height=1080,background_color="black",
max_words=2000, mask=mask, stopwords=stopwords, margin=9,
               random_state=3).
               generate_from_frequencies(keywords) #sortedKeywords

default_colors = wc.to_array()
plt.figure(figsize=(20,10))
plt.imshow(wc.recolor(color_func=grey_color_func,
 random_state=3),interpolation="bilinear")
wc.to_file("wordcloud.png")
plt.axis("off")
plt.show()
```

handle bars
great bike
front brake
balance issues
front fender
front tire
wheel bike
missing parts
fairly easy
nice bike
regular bike
back wheels
bike shop
rear fenders
highly recommend
stability
instructions
assembling
comfortable
tricycle
excellent
difficult
delivery
problems
transport
purchased
christmas
steering
shopping
expected
scratches
recommend
received
daughter
enjoy
assembly instructions
neighborhood
delivered
perfect
replace
replaced
package
paint
reason
directions
special
things
ordered
shipping
surprised
returned
color
tools
exercise
trikes
handlebars
purchase
ridden
wheels
expect
riding
ground
basket
assembled
trouble
seniors
enjoying
miles
front
stuff
small
follow
bolts
fenders
makes
easily
holes
weeks
water
person
ended
house
problem
hills
tight
easier
fixed
carry