

WORD CLOUD OF ACQUIRER BANKS

*Fay
Luo*

- *wordcloud package*
- *Use case*

Word
Cloud
in
1 min

Southeast_Banking_Corp
InterFirst_Corp
First_Florida_Banks_Inc
First_City_Bancorp_of_Texas
Hartford_National_Corp
Mellon_National_Corp
Bank_of_New_England
Affiliated Bankshares of CO
United Virginia Bankshares Inc
First_Virginia_Banks_Inc
Norstar Bancorp
Bancorp_Hawaii_Inc
Zions_Utah_Bancorp
Sun_Banks_Inc
Bank_of_Virginia_Co
Marine_Corp
First_Commerce
First_Commerce
Key_Banks_Inc
United Jersey Bks
Midlantic Banks Inc
First_National_State_Bancorp
Comerica_Inc
Barnett_Banks_FL_Inc
Boatmen's Bancshares
Citizens_and_Southern_GA_Corp
CBT_Corp
Security_Bancorp
First_Union_Corp
SouthTrust_Corp
United Virginia Bankshares Inc
EquiMark Corp
BankAmerica_Corp



wordcloud.ImageColorGenerator

```
class wordcloud.ImageColorGenerator(image, default_color=None) \[source\]
```

Color generator based on a color image.

Generates colors based on an RGB image. A word will be colored using the mean color of the enclosing rectangle in the color image.

After construction, the object acts as a callable that can be passed as `color_func` to the word cloud constructor or to the `recolor` method.

Parameters:

- `image`:nd-array, shape (height, width, 3)
Image to use to generate word colors. Alpha channels are ignored. This should be the same size as the canvas. for the wordcloud.
- `default_color`:tuple or None, default=None
Fallback colour to use if the canvas is larger than the image, in the format (r, g, b). If None, raise `ValueError` instead.

2 Function

wordcloud.WordCloud

```
class wordcloud.WordCloud(font_path=None, width=400, height=200, margin=2, ranks_only=None, prefer_horizontal=0.9, mask=None, scale=1, color_func=None, max_words=200, min_font_size=4, stopwords=None, random_state=None, background_color='black', max_font_size=None, font_step=1, mode='RGB', relative_scaling='auto', regexp=None, collocations=True, colormap=None, normalize_plurals=True, contour_width=0, contour_color='black', repeat=False) \[source\]
```

Word cloud object for generating and drawing.

Parameters:

- `font_path`:string
Font path to the font that will be used (OTF or TTF). Defaults to DroidSansMono path on a Linux machine. If you are on another OS or don't have this font, you need to adjust this path.
- `width`:int (default=400)
Width of the canvas.
- `height`:int (default=200)
Height of the canvas.
- `prefer_horizontal`:float (default=0.90)
The ratio of times to try horizontal fitting as opposed to vertical. If `prefer_horizontal < 1`, the algorithm will try rotating the word if it doesn't fit. (There is currently no built-in way to get only vertical words.)
- `mask`:nd-array or None (default=None)
If not None, gives a binary mask on where to draw words. If mask is not None, width and height will be ignored and the shape of mask will be used instead. All white (#FF or #FFFFFF) entries will be considered "masked out" while other entries will be free to draw on. [This changed in the most recent version!]
- `contour_width`: float (default=0)
If mask is not None and `contour_width > 0`, draw the mask contour.
- `contour_color`: color value (default="black")
Mask contour color.

WordCloud of Acquirer Banks.py

```
1 #!/usr/bin/env python2
2 # -*- coding: utf-8 -*-
3 """
4 WordCloud of Acquirer Bank
5 Created on Wed Nov 13 15:03:13 2018
6
7 @author: fay
8 """
9 from wordcloud import WordCloud
10 from wordcloud import ImageColorGenerator
11 from PIL import Image
12 import numpy as np
13 import os
14
15 base_path='/Users/fay/Desktop/3-Tech/AFP Projects/WordCloud'
16
17 # 🌟Read string from txt
18 banknames = open(base_path + os.sep + 'bankname.txt','r').read()
19 ## encoding = UTF-8, a whole str with separator
20
21 # 🌟Shape: from a picture (transparent background png)
22 ## Default rectangle
23 mask = Image.open(base_path+os.sep+'icon.png')
24 mask = np.array(mask) # get the color of mask image
25
26 # 🌟Color: from the picture
27 color_func = ImageColorGenerator(mask) # Array based on RGB
28
29 # 🌟Font: for En & Ch
30 font = base_path + os.sep + 'SNSanafonGyou.ttf'
31
32 # ! Build a wordcloud
33 wc = WordCloud(font_path=font, background_color="black", max_words=4000, mask=mask, max_font_size=300, color_func=color_func)
34 ## width=400,height=200; max number of words; stopwords;
35 wc.generate_from_text(banknames) # seperate words from str
36
37 wc.to_file(base_path + os.sep + 'wordcloud.png')
```

Thank You

