Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Text Visualization with Wordcloud

Name: Chong Mun Chen

IC Number: 960327-07-5097

Date : 4/7/2023

Introduction : Learning about visualizing texts with Word Cloud and using Mask to change the layout of the Word Cloud.

Conclusion : I understand more about Word Cloud and Mask and how to use them together.

P17 - Visualizing Text with Word Cloud

Word Cloud

What is a word cloud?

Data visualizations (like charts, graphs, infographics, and more) one of the many ways to communicate important information at a glance, but what if the raw data is text-based?

Word clouds (also known as text clouds or tag clouds): the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.

Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases. They can also help business users compare and contrast two different pieces of text to find the wording similarities between the two.

Useful for quick summary of common customer feedback, text documents, identifying new SEO terms to target.

https://pypi.org/project/wordcloud/ (https://pypi.org/project/wordcloud/)

Know how to search for packages?

https://en.wikipedia.org/wiki/Tag_cloud (https://en.wikipedia.org/wiki/Tag_cloud)

References:

https://amueller.github.io/word_cloud/ (https://amueller.github.io/word_cloud/)

https://github.com/amueller/word_cloud (https://github.com/amueller/word_cloud)

https://www.kaggle.com/agisga/word-clouds (https://www.kaggle.com/agisga/word-clouds)

https://www.wordclouds.com/ (https://www.wordclouds.com/)

Installation

conda install -c conda-forge wordcloud

In [3]:
```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

text = "This is my first Word Cloud, Word Cloud is cool. Whatever this is"

wc = WordCloud()
wc = WordCloud(background_color="white", repeat=True)

wc.generate(text)

plt.axis("off")
plt.imshow(wc, interpolation="bilinear")
plt.show()
```

In [4]:
```
from wordcloud import WordCloud, STOPWORDS

STOPWORDS
```

Out[4]:
```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'all',
 'also',
 'am',
 'an',
 'and',
 'any',
 'are',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
```

Let's get real world data

From Wikipedia

conda install -c conda-forge wikipedia

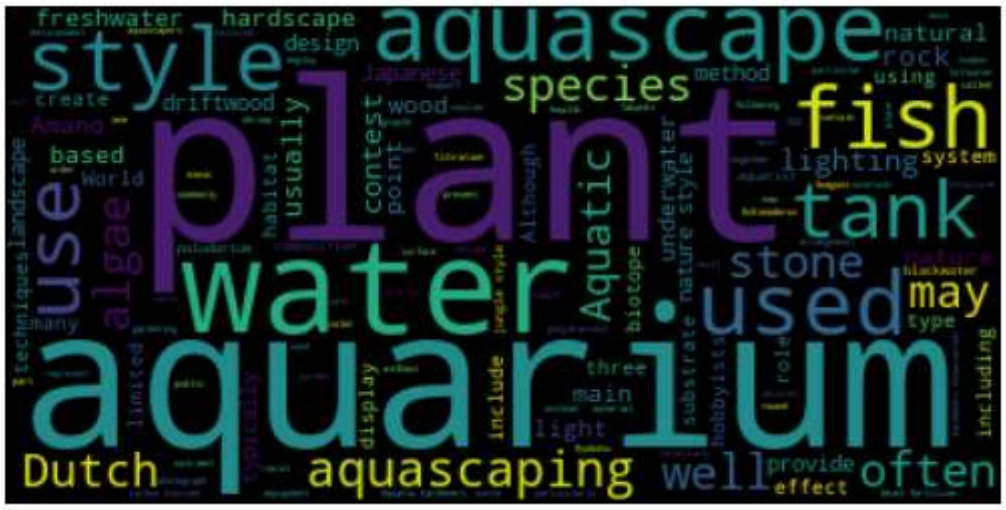
In [1]:
```
import sys
import wikipedia
from wordcloud import WordCloud, STOPWORDS

inputstring = str(input('Enter the title; '))

title = wikipedia.search(inputstring)[0]

page = wikipedia.page(title)

text = page.content
```

```
Enter the title; Aquascaping
```

In [2]:
```
print(text)
```

```
Aquascaping is the craft of arranging aquatic plants, as well as rocks,
stones, cavework, or driftwood, in an aesthetically pleasing manner wit
hin an aquarium—in effect, gardening under water. Aquascape designs inc
lude a number of distinct styles, including the garden-like Dutch style
and the Japanese-inspired nature style. Typically, an aquascape houses
fish as well as plants, although it is possible to create an aquascape
with plants only, or with rockwork or other hardscape and no plants.
Aquascaping appears to have begun to be a popular hobby in the 1930s in
the Netherlands, following the introduction of the Dutch style aquascap
ing techniques. With the increasing availability of mass-produced fresh
water fishkeeping products and popularity of fishkeeping following the
First World War, hobbyists began exploring the new possibilities of cre
ating an aquarium that did not have fish as the main attraction.Althoug
h the primary aim of aquascaping is to create an artful underwater land
scape, the technical aspects of tank maintenance and the growth require
ments of aquatic plants are also taken into consideration. Many factors
must be balanced in the closed system of an aquarium tank to ensure the
success of an aquascape. These factors include filtration, maintaining
carbon dioxide at levels sufficient to support photosynthesis underwate
```

In [5]:
```
import matplotlib.pyplot as plt

wordcloud  = WordCloud(background_color='black', max_words=200, stopwords=S

wordcloud.generate(text)

plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```



2. From PDF File

In [6]:
```
import requests

url = 'https://www.agc.gov.my/agcportal/common//uploads/publication/391/202

# Download the PDF
myfile = requests.get(url, allow_redirects=True, verify=False)

open('./IT_Security_Policy_for_AGC.pdf', 'wb').write(myfile.content)
```

```
C:\Users\ACER\anaconda3\envs\python-dscourse\lib\site-packages\urllib3\co
nnectionpool.py:1056: InsecureRequestWarning: Unverified HTTPS request is
being made to host 'www.agc.gov.my'. Adding certificate verification is s
trongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-u
sage.html#ssl-warnings (https://urllib3.readthedocs.io/en/1.26.x/advanced
-usage.html#ssl-warnings)
  warnings.warn(
```

Out[6]: 1485266

In [ ]:
```
!conda install PyPDF2
```

In [1]:
```
# Convert PDF to Text
import PyPDF2

with open('IT_Security_Policy_for_AGC.pdf','rb') as pdf_file, open('IT_Secu
    read_pdf = PyPDF2.PdfFileReader(pdf_file)
    number_of_pages = read_pdf.getNumPages()
    for page_number in range(number_of_pages):
        page = read_pdf.getPage(page_number)
        page_content = page.extractText()
        text_file.write(page_content)
```

In [2]:
```
page_content
```

Out[2]:
```
'DASAR KESELAMATAN TE KNOLOGI MAKLUMAT JAB ATAN PEGUAM NEGARA   \nTARIKH
: 15 FEBRUARI  2018 MUKA SURAT  74 DARI 75 \n o) Akta Rahsia Rasmi 1972;
\n \np) Akta Jenayah Komputer 1997;  \n \nq) Akta Hak Cipta (Pindaan) Tah
un 1997;  \n \nr) Akta Komunikasi dan Multimedia 1998;  \n \ns) Perintah
-Perintah Am;  \n \nt)  Arahan Perbendaharaan;  \n \nu) Arahan Teknologi
Maklumat 2007;  \n \nv)  Garis Panduan Keselamatan AGC 2004;  \n \nw) Sta
ndard Operating Procedure (SOP) ICT AGC;  \n \nx) Surat Pekeliling Am Bil
angan 3 Tahun 2009 – Garis Panduan Penilaian Tahap \nKeselamatan Rangkaia
n dan Sistem ICT Sektor Awam yang bertarikh 17 \nNovember 2009;  \n \ny)
Surat Arahan Peguam Negara AGC – Pengurusan Kesinambungan \nPerkhidmatan
Ag ensi Sektor Awam yang bertarikh 22 Januari 2010.  \n '
```

Alternative PDF libraries

https://anaconda.org/anaconda/repo (https://anaconda.org/anaconda/repo)

http://mstamy2.github.io/PyPDF2/ (http://mstamy2.github.io/PyPDF2/)

https://pypi.org/project/pdftotext/ (https://pypi.org/project/pdftotext/)

https://realpython.com/pdf-python/ (https://realpython.com/pdf-python/)

Downloading Files

https://dzone.com/articles/simple-examples-of-downloading-files-using-python
(https://dzone.com/articles/simple-examples-of-downloading-files-using-python)

In [ ]:
```
# %matplotlib inline
```

In [3]:

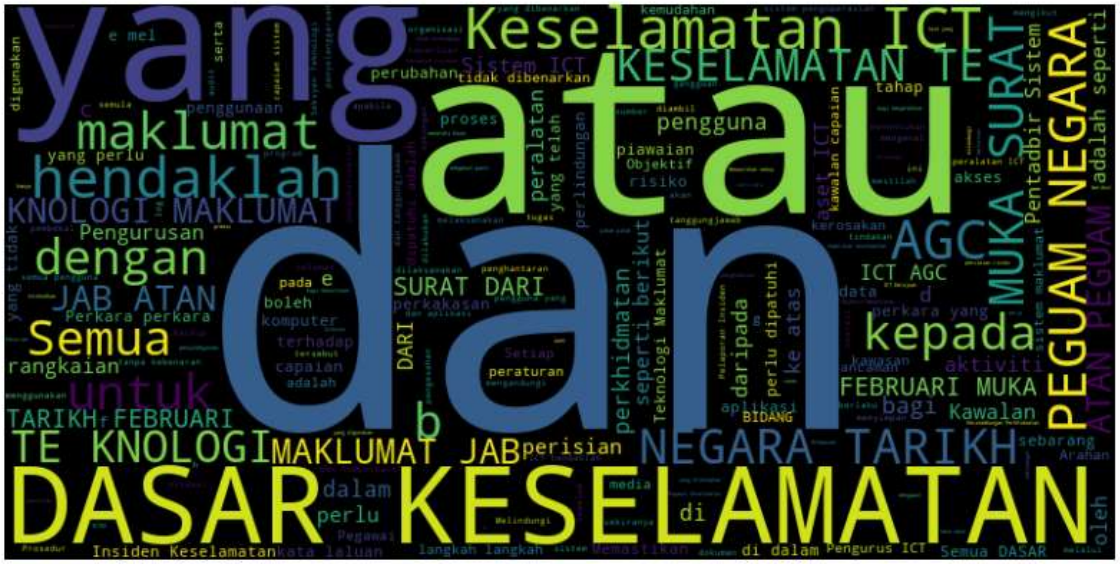
```
from wordcloud import WordCloud, STOPWORDS

# Read the whole text.
text = open('./IT_Security_Policy_for_AGC.txt', encoding='utf-8').read()
text = text.replace ('  ',' ')

# Generate a word cloud image
wordcloud = WordCloud().generate(text)

# Display the generated image:

# the matplotlib way:
import matplotlib.pyplot as plt
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()

# The pil way (if you don't have matplotlib)
# from IPython.display import Image
# pil_img = wordcloud.to_image()
# display(pil_img)
```

In [4]:

```
# Generate a word cloud image
wordcloud = WordCloud().generate(text)

# Display the generated image:
plt.figure(figsize=(10,10)) #inches
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')

plt.show()

# note image size generated and the canvas size of plot
# https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.figure.html
```

In [5]:

```
# Generate a word cloud image
wordcloud = WordCloud(width=800, height=400).generate(text)
#wordcloud = WordCloud(width=3600, height=1600).generate(text)

# Display the generated image:
plt.figure(figsize=(10,10)) # inches
plt.axis("off")
plt.imshow(wordcloud, interpolation='bilinear')
plt.show()

# note image size generated and the canvas size of plot
```



In [6]:
```
# Lower max_font_size
wordcloud = WordCloud(max_font_size=20).generate(text)

# Display the generated image:
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[6]: <function matplotlib.pyplot.show(close=None, block=None)>

In [7]:
```
# Change font size, Background Color
wordcloud = WordCloud(max_font_size=50, background_color='white').generate(
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[7]: <function matplotlib.pyplot.show(close=None, block=None)>

In [8]:
```
# Lower font size, maximum words, Background Color
wordcloud = WordCloud(max_font_size=50, max_words=10,background_color='whit
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[8]: <function matplotlib.pyplot.show(close=None, block=None)>

In [9]:
```
from wordcloud import STOPWORDS

# Create stopword list:
stopwords = set(STOPWORDS)

wordcloud = WordCloud(stopwords=stopwords, max_font_size=50, max_words=10,b
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[9]: <function matplotlib.pyplot.show(close=None, block=None)>

In [10]:
```
from wordcloud import STOPWORDS

# Create stopword list:
stopwords = set(STOPWORDS)
stopwords.update(["yang","di","sabah","sarawak", "section","force","clause'
# stop_words = list(stopwords)+["yang", "di","sabah sarawak", "section forc

wordcloud = WordCloud(stopwords=stopwords, max_font_size=50, max_words=10,b
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[10]: <function matplotlib.pyplot.show(close=None, block=None)>

In [11]:
```
stopwords
```

```
 "couldn't",
 'di',
 'did',
 "didn't",
 'do',
 'does',
 "doesn't",
 'doing',
 "don't",
 'down',
 'during',
 'each',
 'else',
 'ever',
 'federal',
 'few',
 'for',
 'force',
 'from',
 'further',
```

In [13]:
```
from wordcloud import WordCloud, STOPWORDS

testtext = 'yang di is'

# Create stopword list:
stopwords = STOPWORDS
stop_words = ['yang'] + list(stopwords)

wordcloud = WordCloud(stopwords=stop_words, max_font_size=50, max_words=10,

plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
```

Out[13]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

Mask

Change the layout

Generate a Numpy grid

In [17]:

```
# Add Mask
import numpy as np

x, y = np.ogrid[:300, :300]

mask = (x - 150) ** 2 + (y - 150) ** 2 > 130 ** 2
mask = 255 * mask.astype(int)

wordcloud = WordCloud(max_font_size=50, background_color='white', mask=mask
plt.figure()
plt.axis("off")
plt.imshow(wordcloud, interpolation="bilinear")
plt.show
plt.savefig('mask.png')
```

Mask from another Image

First find or create an Image

Eg.

1. Use Paint and save it as mask.png

In [18]:
```
from IPython.display import display, Image
display(Image(filename='./mask.png'))
```

In [20]:
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

mask = np.array(Image.open('./yellow-house-hi.png'))

color= ImageColorGenerator(mask)

wordcloud = WordCloud(width=50,
                      height=50,
                      max_words=50,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white',
                      random_state=42).generate(text)

plt.figure(figsize=(20,20)) # inches
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color), interpolation='bilinear')
plt.show()
```

Read up

https://matplotlib.org/gallery/images_contours_and_fields/interpolation_methods.html (https://matplotlib.org/gallery/images_contours_and_fields/interpolation_methods.html)

2. Or download an Image

Flag of Malaysia

User Google Search

Find Images with larger sizes

Eg. https://en.wikipedia.org/wiki/Flag_of_Malaysia#/media/File:Flag_of_Malaysia.svg (https://en.wikipedia.org/wiki/Flag_of_Malaysia#/media/File:Flag_of_Malaysia.svg)

In [21]:

```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

mask = np.array(Image.open('./1920px-Flag_of_Malaysia.svg.png'))

color= ImageColorGenerator(mask)

wordcloud = WordCloud(width=1920,
                      height=1080,
                      max_words=400,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white',
                      random_state=42).generate(text)

plt.figure(figsize=(10,10)) # inches
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color),interpolation='bilinear')
plt.show()
```

In [22]:
```
# Save to File
wordcloud.to_file('MalaysiaWordCloud.png')
```

Out[22]:
```
<wordcloud.wordcloud.WordCloud at 0x1f937b8d780>
```

Try all the examples below

Python script to search google and produce a word cloud from the abstracts of the first page of results

https://github.com/charlie9578/googleWordCloud (https://github.com/charlie9578/googleWordCloud)

In [23]:

```
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
import urllib
import requests
import numpy as np
import matplotlib.pyplot as plt

words = 'access guest guest apartment area area bathroom bed bed bed bed be
mask = np.array(Image.open(requests.get('http://www.clker.com/cliparts/O/i/

# This function takes in your text and your mask and generates a wordcloud.
def generate_wordcloud(words, mask):
    word_cloud = WordCloud(width = 512, height = 512, background_color='whi
    plt.figure(figsize=(10,8),facecolor = 'white', edgecolor='blue')
    plt.imshow(word_cloud)
    plt.axis('off')
    plt.tight_layout(pad=0)
    plt.show()

#Run the following to generate your wordcloud
generate_wordcloud(words, mask)
```

Download from the source

The source code of word_cloud https://github.com/amueller/word_cloud (https://github.com/amueller/word_cloud)

The Jupyter notebooks https://amueller.github.io/word_cloud/ (https://amueller.github.io/word_cloud/)

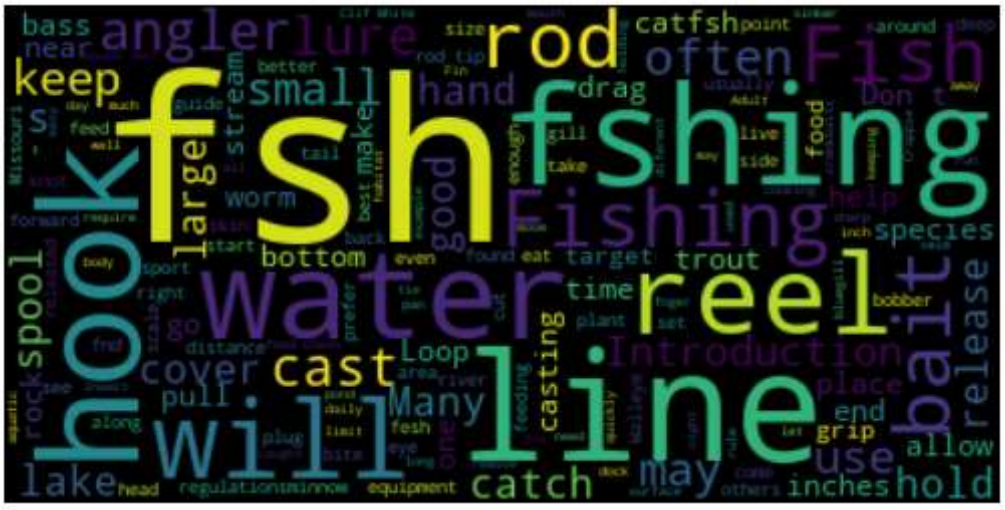
Quiz

1. Download pdf from this link: https://huntfish.mdc.mo.gov/sites/default/files/downloads/page/IntroToFishing_2017_v2.pdf (https://huntfish.mdc.mo.gov/sites/default/files/downloads/page/IntroToFishing_2017_v2.pdf)
2. Text Visualization without mask for this text (using WordCloud)(Black and White)
3. Text Visualization with a mask (you can choose your prefered mask)

- Put in the url link of your mask

In [26]:

```
import PyPDF2
import matplotlib.pyplot as plt
import numpy as np
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from PIL import Image

with open('IntroToFishing_2017_v2.pdf','rb') as pdf_file, open('IntroToFish
    read_pdf = PyPDF2.PdfFileReader(pdf_file)
    number_of_pages = read_pdf.getNumPages()
    for page_number in range(number_of_pages):
        page = read_pdf.getPage(page_number)
        page_content = page.extractText()
        text_file.write(page_content)

text = open('./IntroToFishing_2017_v2.txt', encoding='utf-8').read()
text = text.replace ('  ',' ')

wordcloudBlack = WordCloud().generate(text)
plt.axis("off")
plt.imshow(wordcloudBlack, interpolation='bilinear')
plt.show()

wordcloudWhite = WordCloud(background_color='white').generate(text)
plt.axis("off")
plt.imshow(wordcloudWhite, interpolation='bilinear')
plt.show()

mask = np.array(Image.open('./kyogre.jpg'))
color= ImageColorGenerator(mask)
wordcloud = WordCloud(width=500,
                      height=500,
                      mask=mask,
                      stopwords=STOPWORDS,
                      background_color='white').generate(text)
plt.figure(figsize=(8,8), facecolor = 'blue', edgecolor='blue')
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=color), interpolation='bilinear')
plt.tight_layout(pad=0)
plt.show()
```




fsh
fshing
hook
reel
water
line
will
Fishing
Fish
rod
bait
angler
lure
cast
casting
catch
small
may
Introduction
keep
often
use
release
catfsh
large
make
hold
bottom
good
cover
place
species
worm
food
rock
spool
stream
bass
end
hand
go
lake
trout
near
pull
Don t
Many
inches
target
drag
grip
help
one
allow


fsh
fshing
hook
line
water
reel
Will
bait
rod
angler
Introduction
Fishing
Fish
cast
lure
keep
may
often
trout
lake
large
use
catch
end
good
Many
hold
stream
hand
bass
species
casting
spool
release
catfsh
cover
small
help
make
pull
target
bottom
inches
time