7-2-25class

February 7, 2025

[1]:
```
!pip install wordcloud
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting wordcloud
  Using cached wordcloud-1.9.4-cp312-cp312-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in
c:\programdata\anaconda3\lib\site-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-
packages (from wordcloud) (10.4.0)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-
packages (from wordcloud) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Using cached wordcloud-1.9.4-cp312-cp312-win_amd64.whl (301 kB)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.4

  WARNING: The script wordcloud_cli.exe is installed in
'C:\Users\abhig\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
```

[2]:
```
import wordcloud
```

[5]:
```
dir(wordcloud.WordCloud.generate)
```

[5]:
```
['__annotations__',
 '__builtins__',
 '__call__',
 '__class__',
 '__closure__',
 '__code__',
 '__defaults__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__get__',
 '__getattribute__',
 '__getstate__',
 '__globals__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__kwdefaults__',
 '__le__',
 '__lt__',
 '__module__',
 '__name__',
 '__ne__',
 '__new__',
 '__qualname__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__type_params__']
```

[ ]:
```
wordcloud.WordCloud.generate()
```

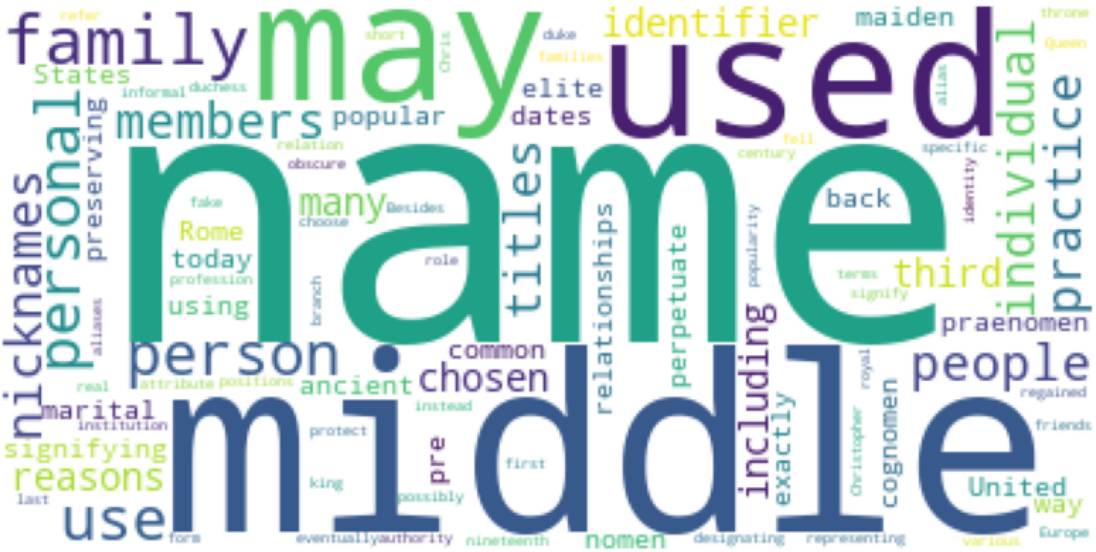
[7]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

[14]:
```
text = '''
Middle names are also used by many people as a third identifier, and can be␣
 ↪chosen for personal reasons including signifying relationships, preserving␣
 ↪pre-marital/maiden names (a popular practice in the United States), and to␣
 ↪perpetuate family names. The practice of using middle names dates back to␣
 ↪ancient Rome, where it was common for members of the elite to have a␣
 ↪praenomen (a personal name), a nomen (a family name, not exactly used the␣
 ↪way middle names are used today), and a cognomen (a name representing an␣
 ↪individual attribute or the specific branch of a person's family).[4] Middle␣
 ↪names eventually fell out of use, but regained popularity in Europe during␣
 ↪the nineteenth century.[4]

Besides first, middle, and last names, individuals may also have nicknames,␣
 ↪aliases, or titles. Nicknames are informal names used by friends or family␣
 ↪to refer to a person ("Chris" may be used as a short form of the personal␣
 ↪name "Christopher"). A person may choose to use an alias, or a fake name,␣
 ↪instead of their real name, possibly to protect or obscure their identity.␣
 ↪People may also have titles designating their role in an institution or␣
 ↪profession (members of royal families may use various terms such as king,␣
 ↪Queen, duke, or duchess to signify their positions of authority or their␣
 ↪relation to the throne).'''
```

[ ]:
```
WordCloud()
```

[15]:
```
plt.figure(figsize=(10,10))
wordcloud = WordCloud(background_color='white').generate(text)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



[32]:
```
import PIL.Image
import numpy as np
```

[45]:
```
mask = np.array(PIL.Image.open('lion.jpg'))
```

[51]:
```
mask.shape
```

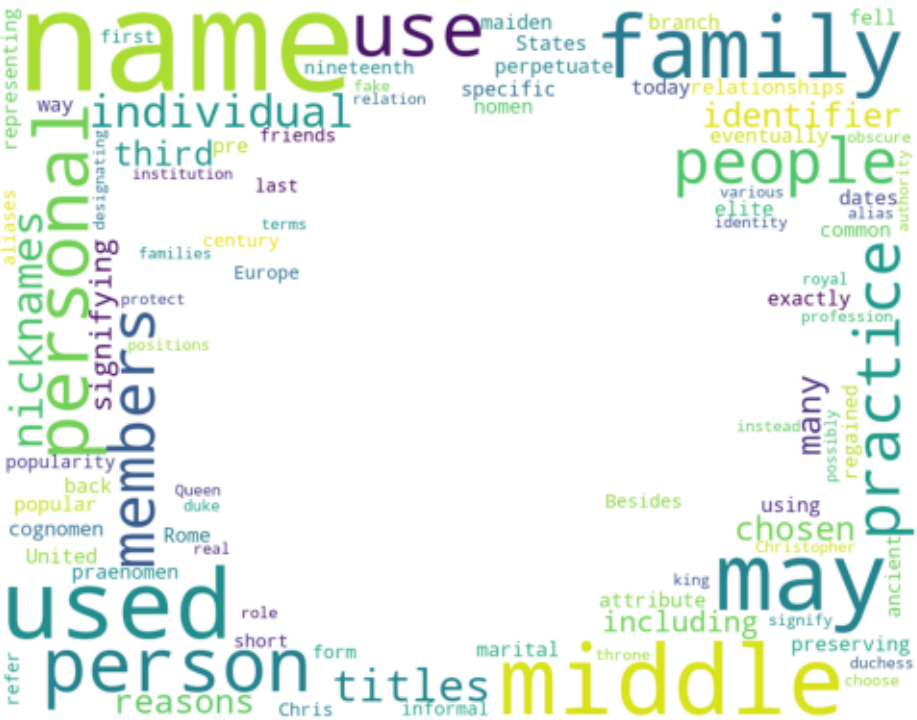
[51]:
```
(778, 1000)
```

[47]:
```
mask=255-mask
```

[48]:
```
wc = WordCloud(background_color='white', mask=mask)
wc.generate(text)
```

[48]:
```
<wordcloud.wordcloud.WordCloud at 0x1f25167c230>
```

[49]:
```
plt.imshow(wc)
plt.axis('off')
plt.show()
```



[53]:
```
import PIL.Image
import numpy as np
```

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Load image and convert to grayscale
mask = np.array(PIL.Image.open('lion.jpg').convert('L'))

# Convert mask to binary: White (255) for shape, Black (0) for background
mask = np.where(mask > 128, 255, 0).astype(np.uint8)

# Print shape to confirm
print("Mask shape:", mask.shape)

# Generate word cloud
wc = WordCloud(background_color="white", mask=mask, mode="RGB",␣
 ↪contour_width=2, contour_color="black")
wc.generate(text)

# Display the word cloud
plt.figure(figsize=(10, 10))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```

```
Mask shape: (778, 1000)
```

Adobe Stock | #

[ ]: