Week-11

September 26, 2024

[2]:
```
from nltk.tokenize import sent_tokenize
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[2]: True

[8]:
```
file=open('nlp.txt','r')
text=file.read()
print(text)
```

```
If you have ever worked at a FAANG or even technology-driven start-up like
Instacart, then you have probably realized that data drives everything.

To the point that analysts, PMs, and product managers are starting to understand
SQL out of necessity. SQL is the language of data and if you want to interact
with data, you need to know it.

Do you want to easily figure out the average amount of time a user spends on
your product, but donâ€ t want to wait for an analyst? You better figure out how
to run a query.

This ability to run queries easily is also driven by the fact that SQL editors
no longer need to be installed. With cloud-based data, warehouses come SaaS SQL
editors. We will talk about a SaaS SQL editor more in the next section.

However, the importance here is you donâ€ t have to wait 30 minutes to install
and editor and deal with all the hassle of managing it.

Now you can just go to a URL and access your teamâ€ s data warehouse. This has
allowed anyone in the company easy access to their data.

We know both from anecdotal experience as well as the fact that indeed.comâ€ s
tracking in 2019 has shown a steady requirement for SQL skill sets for the past
5 years.
```

[9]:
```
sentences=sent_tokenize(text)
```

[10]:
```
print("number of sentences:",len(sentences))
for i in  range(len(sentences)):
    print("\nSentence",i+1,":\n",sentences[i])
```

```
number of sentences: 12

Sentence 1 :
 If you have ever worked at a FAANG or even technology-driven start-up like
Instacart, then you have probably realized that data drives everything.

Sentence 2 :
 To the point that analysts, PMs, and product managers are starting to
understand SQL out of necessity.

Sentence 3 :
 SQL is the language of data and if you want to interact with data, you need to
know it.

Sentence 4 :
 Do you want to easily figure out the average amount of time a user spends on
your product, but donâ€ t want to wait for an analyst?

Sentence 5 :
 You better figure out how to run a query.

Sentence 6 :
 This ability to run queries easily is also driven by the fact that SQL editors
no longer need to be installed.

Sentence 7 :
 With cloud-based data, warehouses come SaaS SQL editors.

Sentence 8 :
 We will talk about a SaaS SQL editor more in the next section.

Sentence 9 :
 However, the importance here is you donâ€ t have to wait 30 minutes to install
and editor and deal with all the hassle of managing it.

Sentence 10 :
 Now you can just go to a URL and access your teamâ€ s data warehouse.

Sentence 11 :
 This has allowed anyone in the company easy access to their data.

Sentence 12 :
```

```
 We know both from anecdotal experience as well as the fact that indeed.comâ€ s
tracking in 2019 has shown a steady requirement for SQL skill sets for the past
5 years.
```

[11]:
```
from nltk.tokenize import word_tokenize
```

[12]:
```
words=word_tokenize(text)
```

[13]:
```
print("total number of words:",len(words))
print(words)
```

```
total number of words: 236
['If', 'you', 'have', 'ever', 'worked', 'at', 'a', 'FAANG', 'or', 'even',
'technology-driven', 'start-up', 'like', 'Instacart', ',', 'then', 'you',
'have', 'probably', 'realized', 'that', 'data', 'drives', 'everything', '.',
'To', 'the', 'point', 'that', 'analysts', ',', 'PMs', ',', 'and', 'product',
'managers', 'are', 'starting', 'to', 'understand', 'SQL', 'out', 'of',
'necessity', '.', 'SQL', 'is', 'the', 'language', 'of', 'data', 'and', 'if',
'you', 'want', 'to', 'interact', 'with', 'data', ',', 'you', 'need', 'to',
'know', 'it', '.', 'Do', 'you', 'want', 'to', 'easily', 'figure', 'out', 'the',
'average', 'amount', 'of', 'time', 'a', 'user', 'spends', 'on', 'your',
'product', ',', 'but', 'donâ€ t', 'want', 'to', 'wait', 'for', 'an', 'analyst',
'?', 'You', 'better', 'figure', 'out', 'how', 'to', 'run', 'a', 'query', '.',
'This', 'ability', 'to', 'run', 'queries', 'easily', 'is', 'also', 'driven',
'by', 'the', 'fact', 'that', 'SQL', 'editors', 'no', 'longer', 'need', 'to',
'be', 'installed', '.', 'With', 'cloud-based', 'data', ',', 'warehouses',
'come', 'SaaS', 'SQL', 'editors', '.', 'We', 'will', 'talk', 'about', 'a',
'SaaS', 'SQL', 'editor', 'more', 'in', 'the', 'next', 'section', '.', 'However',
',', 'the', 'importance', 'here', 'is', 'you', 'donâ€ t', 'have', 'to', 'wait',
'30', 'minutes', 'to', 'install', 'and', 'editor', 'and', 'deal', 'with', 'all',
'the', 'hassle', 'of', 'managing', 'it', '.', 'Now', 'you', 'can', 'just', 'go',
'to', 'a', 'URL', 'and', 'access', 'your', 'teamâ€ s', 'data', 'warehouse', '.',
'This', 'has', 'allowed', 'anyone', 'in', 'the', 'company', 'easy', 'access',
'to', 'their', 'data', '.', 'We', 'know', 'both', 'from', 'anecdotal',
'experience', 'as', 'well', 'as', 'the', 'fact', 'that', 'indeed.comâ€ s',
'tracking', 'in', '2019', 'has', 'shown', 'a', 'steady', 'requirement', 'for',
'SQL', 'skill', 'sets', 'for', 'the', 'past', '5', 'years', '.']
```

[14]:
```
words=word_tokenize(text,preserve_line=True)
len(words)
```

[14]: 226

[15]:
```
from nltk.tokenize import word_tokenize
```

[16]:
```
file=open('nlp.txt','r')
text=file.read()
```

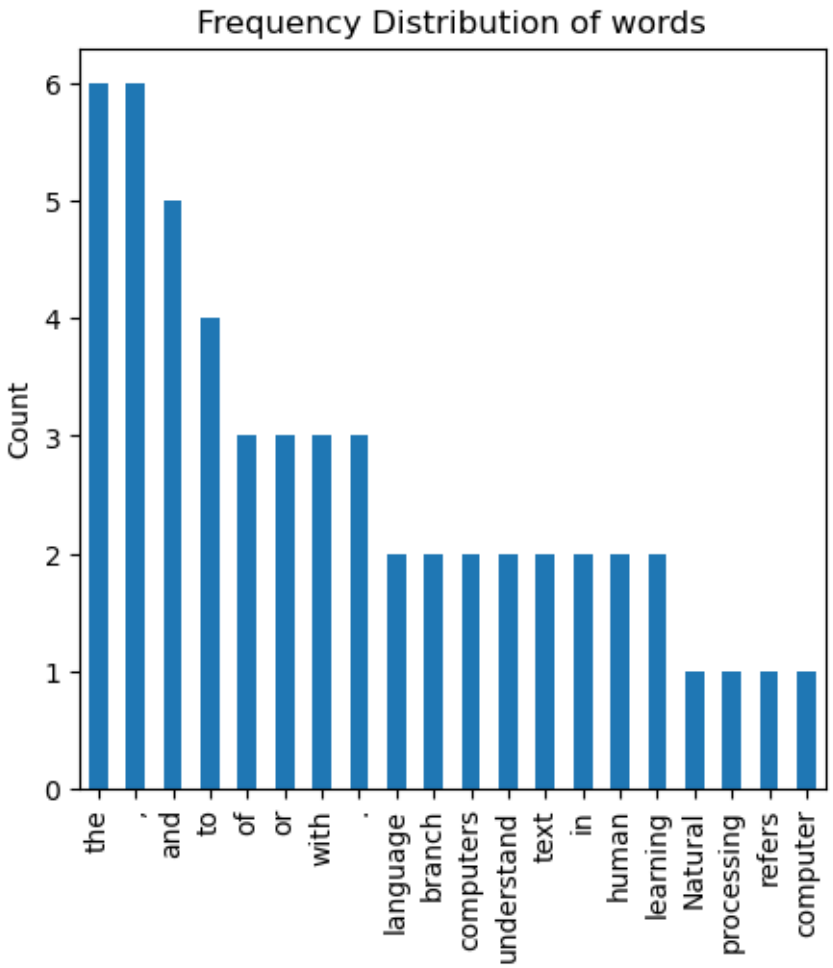
[17]:
```
words=word_tokenize(text)
len(words)
```

[17]:
```
236
```

[18]:
```
from nltk.probability import FreqDist
all_fdist=FreqDist(words).most_common(20)
print(all_fdist)
```

```
[('to', 12), ('.', 11), ('the', 10), ('you', 7), (',', 7), ('a', 6), ('data',
6), ('SQL', 6), ('and', 5), ('that', 4), ('of', 4), ('have', 3), ('out', 3),
('is', 3), ('want', 3), ('for', 3), ('in', 3), ('product', 2), ('with', 2),
('need', 2)]
```

[15]:
```
import matplotlib.pyplot as plt
import pandas as pd
all_fdist=pd.Series(dict(all_fdist))
fig,ax=plt.subplots(figsize=(5,5))
all_fdist.plot(kind='bar')
plt.title('Frequency Distribution of words')
plt.ylabel('Count')
plt.savefig('a.jpg')
```



[19]:
```
text=text.lower()
```

[20]:
```
import re
text=re.sub('[^A-Za-z0-9]+',' ',text)
```

[18]:
```
text=re.sub('\S*\d\S*','',text).strip()
```

[21]:
```
from wordcloud import WordCloud,STOPWORDS
import matplotlib.pyplot as plt
stopwords=set(STOPWORDS)
wordcloud=WordCloud(width=800,height=800, background_color='white',␣
 ↪stopwords=stopwords,min_font_size=10).generate(text)
plt.figure(figsize=(5,5),facecolor=None)
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
```

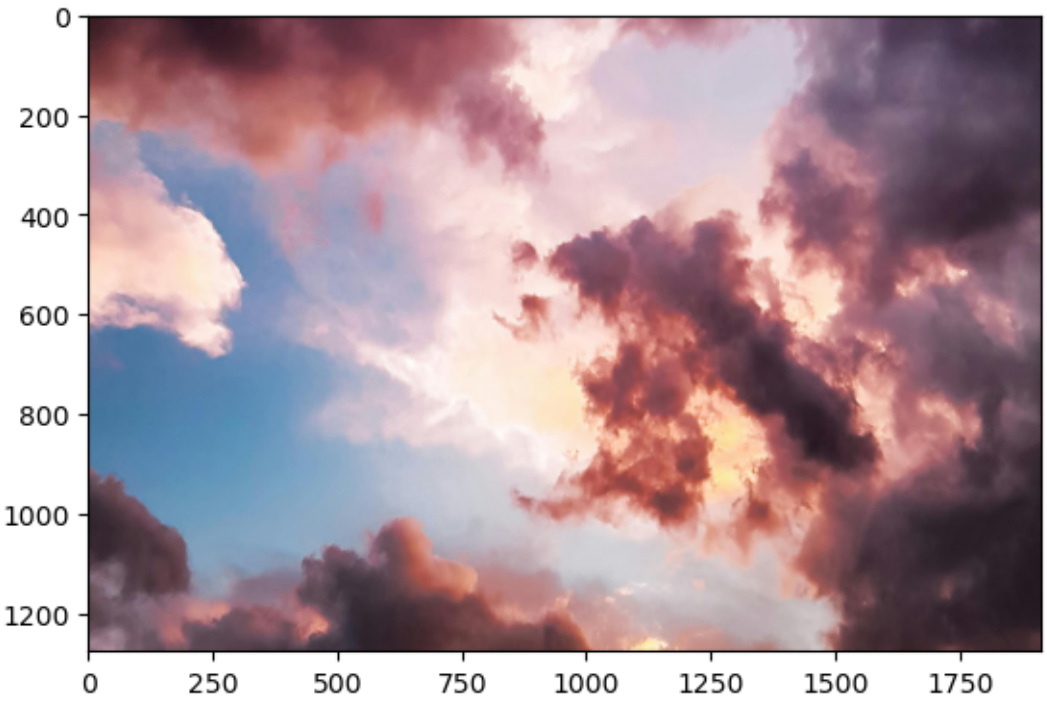
```
plt.show()
```

[21]:
```
from skimage.io import imread
tree=imread('Tree.jpg')
plt.imshow(tree)
```

[21]: <matplotlib.image.AxesImage at 0x2a3aeb8a6d0>

[22]:
```
from skimage.io import imread
cloud1=imread('Cloud1.jpg')
plt.imshow(cloud1)
```

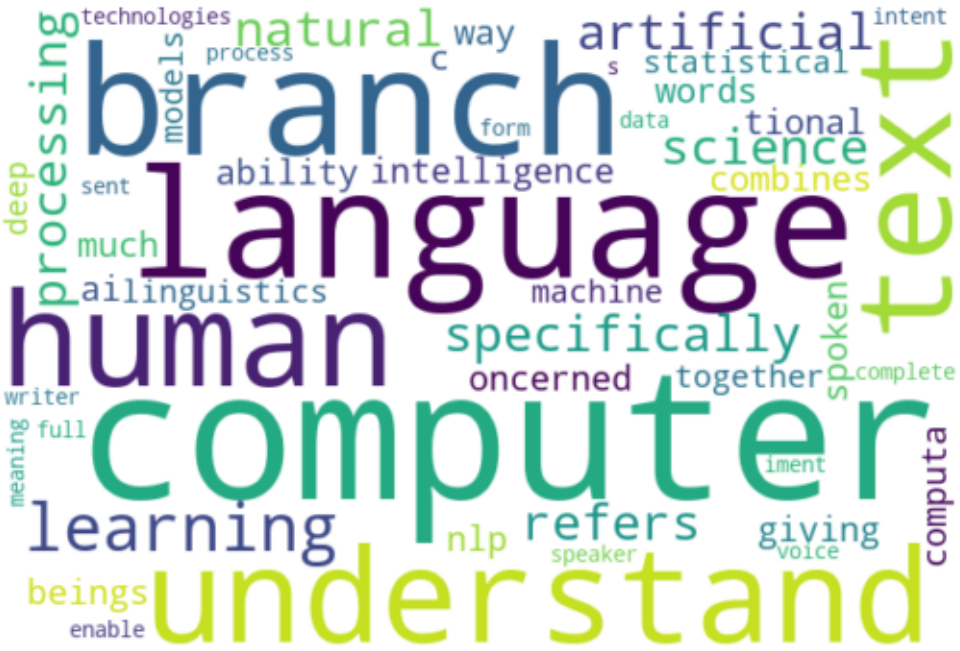
[22]: <matplotlib.image.AxesImage at 0x2a3aebc4590>



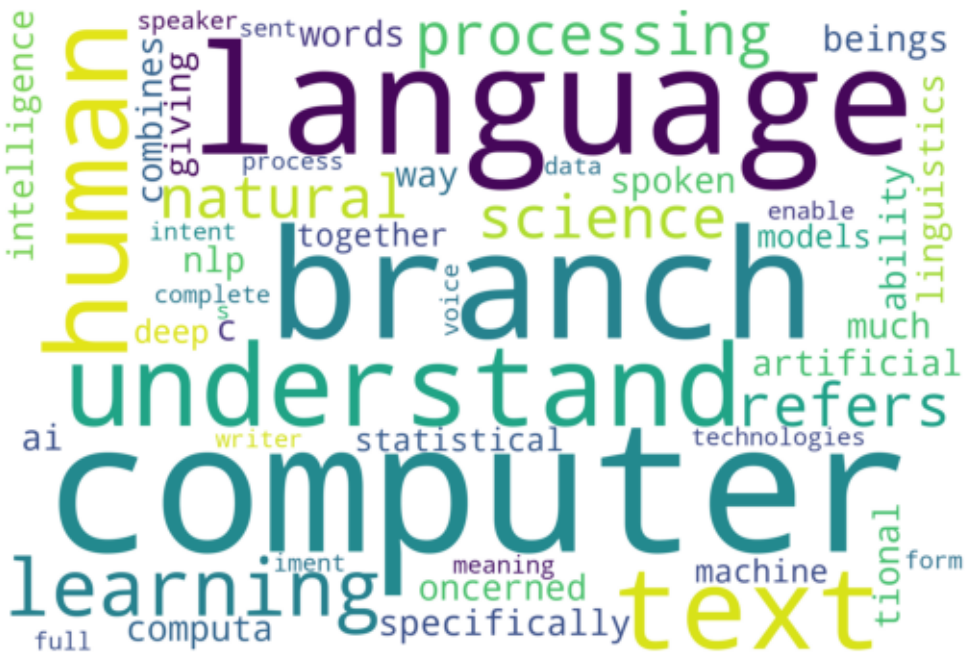
[23]:
```
from skimage.io import imread
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
```

[27]:
```
try:
    cloud = imread('cobra.jpg')
except FileNotFoundError:
    print("File 'cloud.png' not found. Please check the path.")
    cloud = None
if cloud is not None:
    # Define stopwords
    stopwords = set(STOPWORDS)

    # Create the word cloud
    wordcloud = WordCloud(
        width=800,
        height=800,
        background_color='white',
        stopwords=stopwords,
        min_font_size=10,
        mask=cloud
    ).generate(text)

    # Display the word cloud
```

```
    plt.figure(figsize=(5,5), facecolor=None)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.tight_layout(pad=0)
    plt.show()

    # Save the word cloud to a file
    wordcloud.to_file('abc.png')
else:
    print("Cannot generate word cloud as 'cloud.png' is not defined.")
```



[28]:
```
from wordcloud import WordCloud,STOPWORDS
import matplotlib.pyplot as plt
stopwords=set(STOPWORDS)
wordcloud=WordCloud(width=800,height=800,
background_color='white',
stopwords=stopwords,min_font_size=10,mask=cloud1).generate(text)
plt.figure(figsize=(5,5),facecolor=None)
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



[29]:
```
from wordcloud import WordCloud,STOPWORDS
import matplotlib.pyplot as plt
stopwords=set(STOPWORDS)
wordcloud=WordCloud(width=800,height=800, background_color='white',␣
 ↪stopwords=stopwords,min_font_size=10,mask=tree).generate(text)
plt.figure(figsize=(5,5),facecolor=None)
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

[22]:
```
import nltk
from nltk.metrics.distance import edit_distance
nltk.download('words')
from nltk.corpus import words
cw=words.words()
iw=['happpy','amzzzzzing','intelliegent']
for word in iw:
    temp=[(edit_distance(word,w),w)for w in cw if w[0]==word[0]]
print(sorted(temp,key=lambda val:val[0])[0][1])
from nltk.tokenize import word_tokenize
file=open('nlp.txt','r')
text=file.read()
text=text.lower()
import re
text=re.sub('[^A-Za-z0-9]',' ',text)
text=re.sub('\s*\d\s*','',text).strip()
print(text)
words=word_tokenize(text,preserve_line=True)
print(words)
from nltk.stem import PorterStemmer
ps=PorterStemmer()
ps_stem_sent=[ps.stem(words_sent)for words_sent in words]
print(ps_stem_sent)
print(words)
```

```
[nltk_data] Downloading package words to
```

```
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data…
[nltk_data]   Package words is already up-to-date!
```

```
intelligent
if you have ever worked at a faang or even technology driven start up like
instacart  then you have probably realized that data drives everything   to the
point that analysts  pms  and product managers are starting to understand sql
out of necessity  sql is the language of data and if you want to interact with
data  you need to know it   do you want to easily figure out the average amount
of time a user spends on your product  but don   t want to wait for an analyst
you better figure out how to run a query   this ability to run queries easily is
also driven by the fact that sql editors no longer need to be installed  with
cloud based data  warehouses come saas sql editors  we will talk about a saas
sql editor more in the next section   however  the importance here is you don
t have to waitminutes to install and editor and deal with all the hassle of
managing it   now you can just go to a url and access your team   s data
warehouse  this has allowed anyone in the company easy access to their data   we
know both from anecdotal experience as well as the fact that indeed com   s
tracking inhas shown a steady requirement for sql skill sets for the pastyears
['if', 'you', 'have', 'ever', 'worked', 'at', 'a', 'faang', 'or', 'even',
'technology', 'driven', 'start', 'up', 'like', 'instacart', 'then', 'you',
'have', 'probably', 'realized', 'that', 'data', 'drives', 'everything', 'to',
'the', 'point', 'that', 'analysts', 'pms', 'and', 'product', 'managers', 'are',
'starting', 'to', 'understand', 'sql', 'out', 'of', 'necessity', 'sql', 'is',
'the', 'language', 'of', 'data', 'and', 'if', 'you', 'want', 'to', 'interact',
'with', 'data', 'you', 'need', 'to', 'know', 'it', 'do', 'you', 'want', 'to',
'easily', 'figure', 'out', 'the', 'average', 'amount', 'of', 'time', 'a',
'user', 'spends', 'on', 'your', 'product', 'but', 'don', 't', 'want', 'to',
'wait', 'for', 'an', 'analyst', 'you', 'better', 'figure', 'out', 'how', 'to',
'run', 'a', 'query', 'this', 'ability', 'to', 'run', 'queries', 'easily', 'is',
'also', 'driven', 'by', 'the', 'fact', 'that', 'sql', 'editors', 'no', 'longer',
'need', 'to', 'be', 'installed', 'with', 'cloud', 'based', 'data', 'warehouses',
'come', 'saas', 'sql', 'editors', 'we', 'will', 'talk', 'about', 'a', 'saas',
'sql', 'editor', 'more', 'in', 'the', 'next', 'section', 'however', 'the',
'importance', 'here', 'is', 'you', 'don', 't', 'have', 'to', 'waitminutes',
'to', 'install', 'and', 'editor', 'and', 'deal', 'with', 'all', 'the', 'hassle',
'of', 'managing', 'it', 'now', 'you', 'can', 'just', 'go', 'to', 'a', 'url',
'and', 'access', 'your', 'team', 's', 'data', 'warehouse', 'this', 'has',
'allowed', 'anyone', 'in', 'the', 'company', 'easy', 'access', 'to', 'their',
'data', 'we', 'know', 'both', 'from', 'anecdotal', 'experience', 'as', 'well',
'as', 'the', 'fact', 'that', 'indeed', 'com', 's', 'tracking', 'inhas', 'shown',
'a', 'steady', 'requirement', 'for', 'sql', 'skill', 'sets', 'for', 'the',
'pastyears']
['if', 'you', 'have', 'ever', 'work', 'at', 'a', 'faang', 'or', 'even',
'technolog', 'driven', 'start', 'up', 'like', 'instacart', 'then', 'you',
'have', 'probabl', 'realiz', 'that', 'data', 'drive', 'everyth', 'to', 'the',
'point', 'that', 'analyst', 'pm', 'and', 'product', 'manag', 'are', 'start',
'to', 'understand', 'sql', 'out', 'of', 'necess', 'sql', 'is', 'the', 'languag',
```

```
'of', 'data', 'and', 'if', 'you', 'want', 'to', 'interact', 'with', 'data',
'you', 'need', 'to', 'know', 'it', 'do', 'you', 'want', 'to', 'easili', 'figur',
'out', 'the', 'averag', 'amount', 'of', 'time', 'a', 'user', 'spend', 'on',
'your', 'product', 'but', 'don', 't', 'want', 'to', 'wait', 'for', 'an',
'analyst', 'you', 'better', 'figur', 'out', 'how', 'to', 'run', 'a', 'queri',
'thi', 'abil', 'to', 'run', 'queri', 'easili', 'is', 'also', 'driven', 'by',
'the', 'fact', 'that', 'sql', 'editor', 'no', 'longer', 'need', 'to', 'be',
'instal', 'with', 'cloud', 'base', 'data', 'warehous', 'come', 'saa', 'sql',
'editor', 'we', 'will', 'talk', 'about', 'a', 'saa', 'sql', 'editor', 'more',
'in', 'the', 'next', 'section', 'howev', 'the', 'import', 'here', 'is', 'you',
'don', 't', 'have', 'to', 'waitminut', 'to', 'instal', 'and', 'editor', 'and',
'deal', 'with', 'all', 'the', 'hassl', 'of', 'manag', 'it', 'now', 'you', 'can',
'just', 'go', 'to', 'a', 'url', 'and', 'access', 'your', 'team', 's', 'data',
'warehous', 'thi', 'ha', 'allow', 'anyon', 'in', 'the', 'compani', 'easi',
'access', 'to', 'their', 'data', 'we', 'know', 'both', 'from', 'anecdot',
'experi', 'as', 'well', 'as', 'the', 'fact', 'that', 'inde', 'com', 's',
'track', 'inha', 'shown', 'a', 'steadi', 'requir', 'for', 'sql', 'skill', 'set',
'for', 'the', 'pastyear']
['if', 'you', 'have', 'ever', 'worked', 'at', 'a', 'faang', 'or', 'even',
'technology', 'driven', 'start', 'up', 'like', 'instacart', 'then', 'you',
'have', 'probably', 'realized', 'that', 'data', 'drives', 'everything', 'to',
'the', 'point', 'that', 'analysts', 'pms', 'and', 'product', 'managers', 'are',
'starting', 'to', 'understand', 'sql', 'out', 'of', 'necessity', 'sql', 'is',
'the', 'language', 'of', 'data', 'and', 'if', 'you', 'want', 'to', 'interact',
'with', 'data', 'you', 'need', 'to', 'know', 'it', 'do', 'you', 'want', 'to',
'easily', 'figure', 'out', 'the', 'average', 'amount', 'of', 'time', 'a',
'user', 'spends', 'on', 'your', 'product', 'but', 'don', 't', 'want', 'to',
'wait', 'for', 'an', 'analyst', 'you', 'better', 'figure', 'out', 'how', 'to',
'run', 'a', 'query', 'this', 'ability', 'to', 'run', 'queries', 'easily', 'is',
'also', 'driven', 'by', 'the', 'fact', 'that', 'sql', 'editors', 'no', 'longer',
'need', 'to', 'be', 'installed', 'with', 'cloud', 'based', 'data', 'warehouses',
'come', 'saas', 'sql', 'editors', 'we', 'will', 'talk', 'about', 'a', 'saas',
'sql', 'editor', 'more', 'in', 'the', 'next', 'section', 'however', 'the',
'importance', 'here', 'is', 'you', 'don', 't', 'have', 'to', 'waitminutes',
'to', 'install', 'and', 'editor', 'and', 'deal', 'with', 'all', 'the', 'hassle',
'of', 'managing', 'it', 'now', 'you', 'can', 'just', 'go', 'to', 'a', 'url',
'and', 'access', 'your', 'team', 's', 'data', 'warehouse', 'this', 'has',
'allowed', 'anyone', 'in', 'the', 'company', 'easy', 'access', 'to', 'their',
'data', 'we', 'know', 'both', 'from', 'anecdotal', 'experience', 'as', 'well',
'as', 'the', 'fact', 'that', 'indeed', 'com', 's', 'tracking', 'inhas', 'shown',
'a', 'steady', 'requirement', 'for', 'sql', 'skill', 'sets', 'for', 'the',
'pastyears']
```

[23]:
```
import nltk
nltk.download('wordnet')
from nltk.stem.wordnet import WordNetLemmatizer
l=WordNetLemmatizer()
```

```
ls=[l.lemmatize(words_sent)for words_sent in words]
print(ls)
from nltk.stem import WordNetLemmatizer
l=WordNetLemmatizer()
print('rocks:',l.lemmatize('rocks'))
print('corpora:',l.lemmatize('corpora'))
print('better:',l.lemmatize('better',pos='a'))
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

```
['if', 'you', 'have', 'ever', 'worked', 'at', 'a', 'faang', 'or', 'even',
'technology', 'driven', 'start', 'up', 'like', 'instacart', 'then', 'you',
'have', 'probably', 'realized', 'that', 'data', 'drive', 'everything', 'to',
'the', 'point', 'that', 'analyst', 'pm', 'and', 'product', 'manager', 'are',
'starting', 'to', 'understand', 'sql', 'out', 'of', 'necessity', 'sql', 'is',
'the', 'language', 'of', 'data', 'and', 'if', 'you', 'want', 'to', 'interact',
'with', 'data', 'you', 'need', 'to', 'know', 'it', 'do', 'you', 'want', 'to',
'easily', 'figure', 'out', 'the', 'average', 'amount', 'of', 'time', 'a',
'user', 'spends', 'on', 'your', 'product', 'but', 'don', 't', 'want', 'to',
'wait', 'for', 'an', 'analyst', 'you', 'better', 'figure', 'out', 'how', 'to',
'run', 'a', 'query', 'this', 'ability', 'to', 'run', 'query', 'easily', 'is',
'also', 'driven', 'by', 'the', 'fact', 'that', 'sql', 'editor', 'no', 'longer',
'need', 'to', 'be', 'installed', 'with', 'cloud', 'based', 'data', 'warehouse',
'come', 'saas', 'sql', 'editor', 'we', 'will', 'talk', 'about', 'a', 'saas',
'sql', 'editor', 'more', 'in', 'the', 'next', 'section', 'however', 'the',
'importance', 'here', 'is', 'you', 'don', 't', 'have', 'to', 'waitminutes',
'to', 'install', 'and', 'editor', 'and', 'deal', 'with', 'all', 'the', 'hassle',
'of', 'managing', 'it', 'now', 'you', 'can', 'just', 'go', 'to', 'a', 'url',
'and', 'access', 'your', 'team', 's', 'data', 'warehouse', 'this', 'ha',
'allowed', 'anyone', 'in', 'the', 'company', 'easy', 'access', 'to', 'their',
'data', 'we', 'know', 'both', 'from', 'anecdotal', 'experience', 'a', 'well',
'a', 'the', 'fact', 'that', 'indeed', 'com', 's', 'tracking', 'inhas', 'shown',
'a', 'steady', 'requirement', 'for', 'sql', 'skill', 'set', 'for', 'the',
'pastyears']
rocks: rock
corpora: corpus
better: good
```

[24]:
```
import nltk
from nltk import word_tokenize
import nltk
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

[24]: True

[25]:
```
text='I am very hungry but stomak is empty'
words=word_tokenize(text)
print('parts of speech:',nltk.pos_tag(words))
```

```
parts of speech: [('I', 'PRP'), ('am', 'VBP'), ('very', 'RB'), ('hungry', 'JJ'),
('but', 'CC'), ('stomak', 'JJ'), ('is', 'VBZ'), ('empty', 'JJ')]
```

[26]:
```
file=open('nlp.txt','r')
text=file.read()
print(text)
```

```
If you have ever worked at a FAANG or even technology-driven start-up like
Instacart, then you have probably realized that data drives everything.

To the point that analysts, PMs, and product managers are starting to understand
SQL out of necessity. SQL is the language of data and if you want to interact
with data, you need to know it.

Do you want to easily figure out the average amount of time a user spends on
your product, but donâ€ t want to wait for an analyst? You better figure out how
to run a query.

This ability to run queries easily is also driven by the fact that SQL editors
no longer need to be installed. With cloud-based data, warehouses come SaaS SQL
editors. We will talk about a SaaS SQL editor more in the next section.

However, the importance here is you donâ€ t have to wait 30 minutes to install
and editor and deal with all the hassle of managing it.

Now you can just go to a URL and access your teamâ€ s data warehouse. This has
allowed anyone in the company easy access to their data.

We know both from anecdotal experience as well as the fact that indeed.comâ€ s
tracking in 2019 has shown a steady requirement for SQL skill sets for the past
5 years.
```

[ ]:

[ ]: