Text Pre-Processing

In [1]:

```
1 import pandas as pd
```

In [2]:

```
1 f = open("allen.txt")
```

In [3]:

```
1 # print(f.read())
2
```

In [4]:

```
1 data = f.read()
```

In [5]:

```
1 print(data)
```

```
Alan Mathison Turing OBE FRS (/ËˆtjÊŠÉ™rÉªÅ‹/; 23 June 1912 â€“ 7 June 195
4) was an English mathematician, computer scientist, logician, cryptanalys
t, philosopher, and theoretical biologist.[6] Turing was highly influentia
l in the development of theoretical computer science, providing a formalis
ation of the concepts of algorithm and computation with the Turing machin
e, which can be considered a model of a general-purpose computer.[7][8][9]
He is widely considered to be the father of theoretical computer science a
nd artificial intelligence.
```

Pre-Processing

1. LowerCase Conversion
2. Sentence Tokenization
3. Stemming

LowerCase

In [6]:

```
1 data.lower()
```

Out[6]:

```
'alan mathison turing obe frs (/ëˆtjêšé™réªå‹/; 23 june 1912 â€“ 7 june 19
54) was an english mathematician, computer scientist, logician, cryptanaly
st, philosopher, and theoretical biologist.[6] turing was highly influenti
al in the development of theoretical computer science, providing a formali
sation of the concepts of algorithm and computation with the turing machin
e, which can be considered a model of a general-purpose computer.[7][8][9]
he is widely considered to be the father of theoretical computer science a
nd artificial intelligence.'
```

Stemming

In [7]:

```
1 import re
2
3 def remove_punctuation(text):
4     return re.sub(r'[^\w\s]', '', text)
```

In [8]:

```
1 clean_text = remove_punctuation(data)
```

In [9]:

```
1 print(clean_text)
```

```
Alan Mathison Turing OBE FRS ËˆtjÊŠÉrÉªÅ 23 June 1912 â 7 June 1954 was an
English mathematician computer scientist logician cryptanalyst philosopher
and theoretical biologist6 Turing was highly influential in the developmen
t of theoretical computer science providing a formalisation of the concept
s of algorithm and computation with the Turing machine which can be consid
ered a model of a generalpurpose computer789 He is widely considered to be
the father of theoretical computer science and artificial intelligence
```

Tokenization

In [10]:

```
1 from nltk.tokenize import sent_tokenize
```

In [11]:

```
1 sent_tokenize(clean_text)
```

Out[11]:

```
['Alan Mathison Turing OBE FRS ËˆtjÊŠÉrÉªÅ 23 June 1912 â 7 June 1954 was
an English mathematician computer scientist logician cryptanalyst philosop
her and theoretical biologist6 Turing was highly influential in the develo
pment of theoretical computer science providing a formalisation of the con
cepts of algorithm and computation with the Turing machine which can be co
nsidered a model of a generalpurpose computer789 He is widely considered t
o be the father of theoretical computer science and artificial intelligenc
e']
```

In [12]:

```
1 from nltk.tokenize import word_tokenize
```

In [17]:

```
1 tokens = word_tokenize(clean_text)
```

In [18]:

```
1 tokens
```

Out[18]:

```
['Alan',
 'Mathison',
 'Turing',
 'OBE',
 'FRS',
 'Ë^tjÊŠÉrÉªÅ',
 '23',
 'June',
 '1912',
 'â',
 '7',
 'June',
 '1954',
 'was',
 'an',
 'English',
 'mathematician',
 'computer',
 'scientist',
 'logician',
 ',
 'cryptanalyst',
 'philosopher',
 'and',
 'theoretical',
 'biologist',
 'Turing',
 'was',
 'highly',
 'influential',
 'in',
 'the',
 'development',
 'of',
 'theoretical',
 'computer',
 'science',
 'providing',
 'a',
 'formalisation',
 'of',
 'the',
 'concepts',
 'of',
 'algorithm',
 'and',
 'computation',
 'with',
 'the',
 'Turing',
 'machine',
 'which',
 'can',
 'be',
 'considered',
 'a',
 'model',
 'of',
 'a',
 'generalpurpose',
 'computer789',
 'He',
```

In [19]:
```
print(len(tokens))
```

75

Removing Stop Words

In [14]:
```
import nltk
from nltk.corpus import stopwords
```

In [15]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\MSCIT\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[15]:
```
True
```

In [16]:
```
stop_words = set(stopwords.words('english'))
```

In [21]:
```
stop = list(stop_words)
```

In [22]:

```
stp
```

```
'is',
'widely',
'considered',
'to',
'be',
'one',
'of',
'the',
'most',
'important',
'and',
'influential',
'theoretical',
'computer',
'science',
'papers',
'ever',
'written',
'it',
'laid',
'out',
'the',
'mathematical',
'foundations',
'of',
'artificial',
'intelligence']
```

In [25]:

```
filtered_words = [word for word in tokens if word not in stp]
```

In [26]:

```
1 filtered_words
```

Out[26]:

```
['Alan',
 'Mathison',
 'Turing',
 'OBE',
 'FRS',
 'Ë^tjÊŠÉrÉªÅ',
 '23',
 'June',
 '1912',
 'â',
 '7',
 'June',
 '1954',
 'English',
 'mathematician',
 'computer',
 'scientist',
 'logician',
 'cryptanalyst',
 'philosopher',
 'theoretical',
 'biologist6',
 'Turing',
 'highly',
 'influential',
 'development',
 'theoretical',
 'computer',
 'science',
 'providing',
 'formalisation',
 'concepts',
 'algorithm',
 'computation',
 'Turing',
 'machine',
 'considered',
 'model',
 'generalpurpose',
 'computer789',
 'He',
 'widely',
 'considered',
 'father',
 'theoretical',
 'computer',
 'science',
 'artificial',
 'intelligence']
```

In [29]:

```
print("Before Removing Stop Words : ",len(tokens))


print("After Removing Stop Words : ",len(filtered_words))
```

```
Before Removing Stop Words :  75
After Removing Stop Words :  49
```

Stemmimng

In [28]:

```
l1 = ['walk', 'walking', 'walked']
```

In [30]:

```
from nltk.stem import PorterStemmer
```

In [31]:

```
ps = PorterStemmer()
```

In [32]:

```
for w in l1:
    print(w, " : ", ps.stem(w))
```

```
walk  :  walk
walking  :  walk
walked  :  walk
```

In [35]:

```
l2 = ['Probability', 'Probable', 'Probably']
```

In [36]:

```
for w in l2:
    print(w, " : ", ps.stem(w))
```

```
Probability  :  probabl
Probable  :  probabl
Probably  :  probabl
```

Lemmatization

In [37]:

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
```

In [44]:

```
l3 = ['rocks', 'apples','fruits']
```

In [45]:

```
for w in l3:
    print(w, " : ", lemmatizer.lemmatize(w))
```

```
rocks  :  rock
apples  :  apple
fruits  :  fruit
```

In [41]:

```
print("rocks :", lemmatizer.lemmatize("rocks"))
```

```
rocks : rock
```

In [ ]:

```

```