Used Google Colab, need this extra step to mount the drive. Other Jupyter notebooks can skip this step

In [1]:
```
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [2]:
```
folder_path = '/content/drive/MyDrive/NLP_Data/data/'
```

In [4]:
```
import sys
import nltk
import pickle
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.util import bigrams
from collections import defaultdict
```

In [5]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```
Out[5]:
```
True
```

1a. Create a function with a filename as argument; the function will:

- read in the file text and remove newlines
- tokenize the text
- use ntlk to create a bigrams list
- use ntlk to create a unigrams list
- use the bigram list to create a bigram dictionary of bigrams and counts

  > 'token1 token2' -> count note that ('token1', 'token2') also works as a key

- use the unigram list to create a unigram dictionary of unigrams and counts,

  > 'token' -> count

- return both the unigram dictionary and bigram dictionary from the function

In [6]:
```
def bigram_and_unigram_dict(filename):
  file_path = folder_path + filename

  # Attempt to open the file
  try:
    with open(file_path, 'r') as file:
      raw_text = file.read()

  # Catch if file could not be found
  except FileNotFoundError:
    print("The file, ", filename, ", cannot be found")
    sys.exit(1)

  # Catch all other errors
  except Exception as e:
    print(e)
    sys.exit(1)

  # print(raw_text[:100])

  # Remove newlines
  remove_newline_text = raw_text.replace('\n', '')
  # print("\n", remove_newline_text[:100])

  # Adapted from: https://github.com/kjmazidi/NLP/blob/master/Part_2-Words/Chapter_08_ngrams/8_ngrams_1.ipynb
  # Tokenize words
  words = word_tokenize(remove_newline_text)
  # print(words[:100])

  # Create Bigrams
  bigram_list = list(bigrams(words))

  # Create Unigrams (exact same thing as the result of word_tokenize)
  unigram_list = words

  # Create Bigram Dictionary Bigram:count
  bigram_dict = {b:bigram_list.count(b) for b in set(bigram_list)}
  # print(bigram_dict)

  # Create Unigram Dictionary Unigram:count
  unigram_dict = {t:unigram_list.count(t) for t in set(unigram_list)}
  # print(unigram_dict)

  return unigram_dict, bigram_dict
```

Calling the function 3 times, once for each training file (hard code the test names)

In [7]:
```
english_test_name = "LangId.train.English.txt"
french_test_name = "LangId.train.French.txt"
italian_test_name = "LangId.train.Italian.txt"
```

In [8]:
```
english_unigram, english_bigram = bigram_and_unigram_dict(english_test_name)
```

In [9]:
```
french_unigram, french_bigram = bigram_and_unigram_dict(french_test_name)
```

In [10]:
```
italian_unigram, italian_bigram = bigram_and_unigram_dict(italian_test_name)
```

Sorting the unigrams and bigrams by descending order. Printing the first 10 of each ordered dictionary. For demonstation only.

In [11]:
```
# Sorting the dict and printing the first 10 options
def sort_dict_and_print(dict):
  sorted_items = sorted(dict.items(), key=lambda x: x[1], reverse=True)
  print("Top 10 key-value pairs:")
  for i in range(10):
    print(sorted_items[i])
```

In [13]:
```
sort_dict_and_print(english_unigram)
```

```
Top 10 key-value pairs:
('the', 5310)
(',', 3853)
('.', 2832)
('of', 2754)
('to', 2480)
('and', 2027)
('in', 1506)
('is', 1296)
('a', 1274)
('that', 1154)
```

In [14]:
```
sort_dict_and_print(english_bigram)
```

```
Top 10 key-value pairs:
(('of', 'the'), 903)
(('in', 'the'), 418)
(('.', 'The'), 341)
(('to', 'the'), 330)
((',', 'the'), 311)
(('on', 'the'), 289)
(('.', 'I'), 260)
(("'", '.'), 248)
(('and', 'the'), 238)
(('.', 'We'), 235)
```

In [15]:
```
sort_dict_and_print(french_unigram)
```

```
Top 10 key-value pairs:
("'", 4548)
(',', 4286)
('de', 3985)
('.', 2825)
('la', 2437)
('-', 2359)
('et', 1862)
('l', 1786)
('le', 1669)
('à', 1577)
```

In [16]:
```
sort_dict_and_print(french_bigram)
```

```
Top 10 key-value pairs:
(('l', "'"), 1786)
(('d', "'"), 1158)
(("'", '-'), 1040)
(('de', 'la'), 723)
(('de', 'l'), 541)
(('qu', "'"), 440)
(('n', "'"), 267)
(("'", 'est'), 263)
(('à', 'l'), 242)
(('.', 'Il'), 240)
```

In [17]:
```
sort_dict_and_print(italian_unigram)
```

```
Top 10 key-value pairs:
(',', 4013)
('.', 2845)
('di', 2676)
("'", 2164)
('che', 1949)
('e', 1714)
('la', 1459)
('il', 1224)
('in', 1133)
('per', 966)
```

In [18]:
```
sort_dict_and_print(italian_bigram)
```

```
Top 10 key-value pairs:
(('l', "'"), 715)
(('dell', "'"), 432)
(('Presidente', ','), 227)
((',', 'che'), 216)
(('.', 'La'), 215)
(('Signor', 'Presidente'), 179)
(('all', "'"), 160)
((',', 'ma'), 158)
((',', 'in'), 157)
(("'", 'Unione'), 152)
```

Pickling all dictionaries

In [19]:
```
# Hard Coded this step as well
pickle_path = '/content/drive/MyDrive/NLP_Data'

with open(pickle_path + 'english_unigram.pickle', 'wb') as handle:
  pickle.dump(english_unigram, handle)

with open(pickle_path + 'english_bigram.pickle', 'wb') as handle:
  pickle.dump(english_bigram, handle)

with open(pickle_path + 'french_unigram.pickle', 'wb') as handle:
  pickle.dump(french_unigram, handle)

with open(pickle_path + 'french_bigram.pickle', 'wb') as handle:
  pickle.dump(french_bigram, handle)

with open(pickle_path + 'italian_unigram.pickle', 'wb') as handle:
  pickle.dump(italian_unigram, handle)

with open(pickle_path + 'italian_bigram.pickle', 'wb') as handle:
  pickle.dump(italian_bigram, handle)
```

In [20]:
```
def show_pickle_comparison(pickle_name):
  with open(pickle_path + pickle_name, 'rb') as handle:
    new_dict = pickle.load(handle)

  sort_dict_and_print(new_dict)
```

In [21]:
```
show_pickle_comparison('english_unigram.pickle')
```

```
Top 10 key-value pairs:
('the', 5310)
(',', 3853)
('.', 2832)
('of', 2754)
('to', 2480)
('and', 2027)
('in', 1506)
('is', 1296)
('a', 1274)
('that', 1154)
```

In [23]:
```
show_pickle_comparison('english_bigram.pickle')
```

```
Top 10 key-value pairs:
(('of', 'the'), 903)
(('in', 'the'), 418)
(('.', 'The'), 341)
(('to', 'the'), 330)
((',', 'the'), 311)
(('on', 'the'), 289)
(('.', 'I'), 260)
(("'", '.'), 248)
(('and', 'the'), 238)
(('.', 'We'), 235)
```

In [24]:
```
show_pickle_comparison('french_unigram.pickle')
```

```
Top 10 key-value pairs:
("'", 4548)
(',', 4286)
('de', 3985)
('.', 2825)
('la', 2437)
('-', 2359)
('et', 1862)
('l', 1786)
('le', 1669)
('à', 1577)
```

In [30]:
```
show_pickle_comparison('french_bigram.pickle')
```

```
Top 10 key-value pairs:
(('l', "'"), 1786)
(('d', "'"), 1158)
(("'", '-'), 1040)
(('de', 'la'), 723)
(('de', 'l'), 541)
(('qu', "'"), 440)
(('n', "'"), 267)
(("'", 'est'), 263)
(('à', 'l'), 242)
(('.', 'Il'), 240)
```

In [27]:
```
show_pickle_comparison('italian_unigram.pickle')
```

```
Top 10 key-value pairs:
(',', 4013)
('.', 2845)
('di', 2676)
("'", 2164)
('che', 1949)
('e', 1714)
('la', 1459)
('il', 1224)
('in', 1133)
('per', 966)
```

In [29]:
```
show_pickle_comparison('italian_bigram.pickle')
```

```
Top 10 key-value pairs:
(('l', "'"), 715)
(('dell', "'"), 432)
(('Presidente', ','), 227)
((',', 'che'), 216)
(('.', 'La'), 215)
(('Signor', 'Presidente'), 179)
(('all', "'"), 160)
((',', 'ma'), 158)
((',', 'in'), 157)
(("'", 'Unione'), 152)
```