In this homework you will create bigram and unigram dictionaries for English, French, and Italian using the provided training data where the key is the unigram or bigram text and the value is the count of that unigram or bigram in the data. Then for the test data, calculate probabilities for each language and compare against the true labels.

```
import nltk
nltk.download("all")
from nltk import word_tokenize
from nltk import sent_tokenize
import pickle
```

```
[nltk_data]    |   Package swadesh is already up-to-date!
[nltk_data]    | Downloading package switchboard to /root/nltk_data...
[nltk_data]    |   Package switchboard is already up-to-date!
[nltk_data]    | Downloading package tagsets to /root/nltk_data...
[nltk_data]    |   Package tagsets is already up-to-date!
[nltk_data]    | Downloading package timit to /root/nltk_data...
[nltk_data]    |   Package timit is already up-to-date!
[nltk_data]    | Downloading package toolbox to /root/nltk_data...
[nltk_data]    |   Package toolbox is already up-to-date!
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package twitter_samples is already up-to-date!
[nltk_data]    | Downloading package udhr to /root/nltk_data...
[nltk_data]    |   Package udhr is already up-to-date!
[nltk_data]    | Downloading package udhr2 to /root/nltk_data...
[nltk_data]    |   Package udhr2 is already up-to-date!
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package unicode_samples is already up-to-date!
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package universal_tagset is already up-to-date!
[nltk_data]    | Downloading package universal_treebanks_v20 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package universal_treebanks_v20 is already up-to-
[nltk_data]    |       date!
[nltk_data]    | Downloading package vader_lexicon to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package vader_lexicon is already up-to-date!
[nltk_data]    | Downloading package verbnet to /root/nltk_data...
[nltk_data]    |   Package verbnet is already up-to-date!
[nltk_data]    | Downloading package verbnet3 to /root/nltk_data...
[nltk_data]    |   Package verbnet3 is already up-to-date!
[nltk_data]    | Downloading package webtext to /root/nltk_data...
[nltk_data]    |   Package webtext is already up-to-date!
[nltk_data]    | Downloading package wmt15_eval to /root/nltk_data...
[nltk_data]    |   Package wmt15_eval is already up-to-date!
[nltk_data]    | Downloading package word2vec_sample to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package word2vec_sample is already up-to-date!
[nltk_data]    | Downloading package wordnet to /root/nltk_data...
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet2021 to /root/nltk_data...
[nltk_data]    |   Package wordnet2021 is already up-to-date!
[nltk_data]    | Downloading package wordnet2022 to /root/nltk_data...
[nltk_data]    |   Package wordnet2022 is already up-to-date!
[nltk_data]    | Downloading package wordnet31 to /root/nltk_data...
[nltk_data]    |   Package wordnet31 is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data...
[nltk_data]    |   Package wordnet_ic is already up-to-date!
[nltk_data]    | Downloading package words to /root/nltk_data...
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package ycoe to /root/nltk_data...
[nltk_data]    |   Package ycoe is already up-to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

```
def functionOne(fileObject):
    # Tokenize the file
    fileRead = fileObject.read() #read file into a string
    fileList = fileRead.splitlines() #remove new lines, trailing spaces. Returns a list
    file = ' '.join(fileList) #make list a string for tokenization
    text = word_tokenize(file)
    bigramsList = list(nltk.bigrams(text))
    unigramsList = text

    # Create the dicts
    #sometimes it's confusing but remember the word is itself the key! (DICT)
    bigramsDict = {} # (bigram, count)
    for i in range(1, len(bigramsList)):
        bigram = bigramsList[i]
        if bigram in bigramsDict:
            bigramsDict[bigram] += 1
        else:
            bigramsDict[bigram] = 1

    unigramsDict = {} # (bigram, count)
```

```
#saved paths for desktop path
#f1 = open("/Users/shaansekhon/Desktop/UTD/NLP/HW2/data/LangId.train.English.txt")
#f2 = open("/Users/shaansekhon/Desktop/UTD/NLP/HW2/data/LangId.train.French.txt")
#f3 = open("/Users/shaansekhon/Desktop/UTD/NLP/HW2/data/LangId.train.Italian.txt")

#working paths for local drive path
f1 = open("/LangId.train.English.txt", mode='r')
f2 = open("/LangId.train.French.txt", mode='r')
f3 = open("/LangId.train.Italian.txt", mode='r')


EnglishBigramsDict, EnglishUnigramsDict = functionOne(f1)
FrenchBigramsDict, FrenchUnigramsDict = functionOne(f2)
ItalianBigramsDict, ItalianUnigramsDict = functionOne(f3)

# error_checking print(EnglishBigramsDict)
# error_checking print(EnglishUnigramsDict)

pickle.dump(EnglishBigramsDict, open('EnglishBigramsDict.pkl', 'wb'))
pickle.dump(EnglishUnigramsDict, open('EnglishUnigramsDict.pkl', 'wb'))
pickle.dump(FrenchBigramsDict, open('FrenchBigramsDict.pkl', 'wb'))
pickle.dump(FrenchUnigramsDict, open('FrenchUnigramsDict.pkl', 'wb'))
pickle.dump(ItalianBigramsDict, open('ItalianBigramsDict.pkl', 'wb'))
pickle.dump(ItalianUnigramsDict, open('ItalianUnigramsDict.pkl', 'wb'))
```