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

def convertTuple(tup):
    str = ''
    for item in tup:
        str = str + item + ' '
    str = str.strip()
    return str

#OPEN THE PICKLES
with open("EnglishBigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    EB = pickle.load(f)
with open("EnglishUnigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    EU = pickle.load(f)
with open("FrenchBigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    FB = pickle.load(f)
with open("FrenchUnigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    FU = pickle.load(f)
with open("ItalianBigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    IB = pickle.load(f)
with open("ItalianUnigramsDict.pkl", "rb") as f: # "rb" because we want to read in binary mode
    IU = pickle.load(f)

#OPEN THE FILES WE WILL USE
testFile = open("/content/LangId.test.txt") # to get file for assessing
f = open("LangId.txt", "w") # to write results to
Lines = testFile.readlines()

#VARIABLES
VocabularySize = len(EU) + len(FU) + len(IU)

index = 0

#MAIN PROGRAM, start with computation
for line in Lines:
  EBCount = 0
  FBCount = 0
  IBCount = 0
  EUCount = 0
  FUCount = 0
  IUCount = 0
  text = word_tokenize(line)
  lineList = list(nltk.bigrams(text))
  #print(lineList)
  index +=1
  for EBTuple in EB:
    if EBTuple in lineList:
      #print(EBTuple)
      #print(lineList)
      #print("TUPLE IN LINE(ENGLISH)")
      EBCount += 1
      for unigram in EU:
        if unigram == EBTuple[0]:
          #print(unigram)
          EUCount += 1
  #print(EBCount , "EBCount")
  #print(EUCount, "EUCount")
  for FBTuple in FB:
    if FBTuple in lineList:
      #print(FBTuple)
      #print(lineList)
      #print("TUPLE IN LINE(FRENCH)")
      FBCount += 1
      for unigram in FU:
        if unigram == FBTuple[0]:
          #print(unigram)
          FUCount += 1
  #print(FBCount , "FBCount")
  #print(FUCount, "FUCount")
  for IBTuple in IB:
    if IBTuple in lineList:
```

```
        #print(IBTuple)
        #print(lineList)
        #print("TUPLE IN LINE (ITALIAN)")
        IBCount += 1
        for unigram in IU:
          if unigram == IBTuple[0]:
           # print(unigram)
           # print("UNIGRAM IN LINE")
            IUCount += 1
    #print(EBCount , "EBCount")
    #print(FBCount, "FBCount")
    #print(IBCount , "IBCount")
    #print(line)

    EnglishProbability = ((EBCount + 1)/(EUCount + VocabularySize))
    #EnglishProbability = EnglishProbability * (EUCount/VocabularySize)
    #print("ENGLISH PROBABILITY")
    #print(EBCount, "ENGLISH")
    #print(FBCount, "FRENCH")
    #print(IBCount, "ITALIAN")
    #print(EnglishProbability)
    FrenchProbability = ((FBCount + 1)/(FUCount + VocabularySize))
    #FrenchProbability = FrenchProbability * (FUCount/VocabularySize)
    #print("FRENCH PROBABILITY")
   # print(FrenchProbability)
    ItalianProbability = ((IBCount + 1)/(IUCount + VocabularySize))
    #ItalianProbability = ItalianProbability * (IUCount/VocabularySize)
    #print("ITALIAN PROBABILITY")
    #print(ItalianProbability)

    LanguageScore = max(EnglishProbability, FrenchProbability, ItalianProbability)
    if LanguageScore == EnglishProbability:
      indexString = str(index)
      Language = indexString + " English"
    if LanguageScore == FrenchProbability:
      indexString = str(index)
      Language = indexString + " French"
    if LanguageScore == ItalianProbability:
      indexString = str(index)
      Language = indexString + " Italian"

    f.write(Language + '\n')


f.close()
f = open("/content/LangId.txt", "r")
key = open("/content/LangId.sol.txt", "r")
test_test = f.readlines()
line_key = key.readlines()
correct = 0
wrongLines = []


for x in range (0,len(test_test)):
  if test_test[x] == line_key[x]:
    correct += 1
  else:
    wrongLines += [x] # remember we start at 0 technically


accuracy = correct/len(test_test)
print(accuracy)
print(wrongLines)
```

```
[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
0.9933333333333333
[86, 91]
```