Cofer Assignment

Daniel Ventura - 318875085

First of all, the assignment took a lot of thinking and i still i couldnt find a 100% working solution.

the assignment:
we have to build a program that identify an encrypted text file.

i encountered a few problems that could breach any solution i could think of:

1) if the attacker decides to encrypt a file and then convert the encryption to ACSII i have to find a way to defer between normal text and english text.
2) i could check for words that are not in english, but then if a normal edit of the file happens and a word like “Atuda” is added to the file it would be considered as encryption and that is a case we want to avoid.
3) the last problem is that there is an option that only a part of a text file would be encrypted. so we have to be able to identify it.

My solution:
i chose to make a 2 part solution, it is not the fastest way of solving but it gets most cases of encryption in relatively short period of time.

first i look for a non ascii text in the folder we chose and if i find something like that it is indeed encrypted and we are done.
in addition every time i scan a file i save its last editing time stemp.

then i go over all the files that have ascii only text and i take each word in the text without repetitions and find out if its an English word or not.

then, because there are words, names and shortcuts that just will not be considered as English words we cannot just say a file is encrypted for having non English words, so what i did is dividing the number of nonEnglish words by the total amount of non repeting words and if it passes the 35 percent thresh hold it’s encrypted and if not then its not.

i know working with percents is problematic but i couldn’t find any other way of soving this problem.

this is my code:
note: at start up of the program you have to enter the path of the folder with ‘ at the beginning and ‘ at the end. for example: ‘`/root/Desktop/python/ex4/textenc/`’

```
import os,time
import enchant

def isEnglishWord(text):  #This function checks whether a String is a word in
English or not.
 try:
   dictionary = enchant.Dict("en_US")
   b = dictionary.check(text)
   if(b == False):
     return False
   return True
 except:
   return True


def is_ascii(text):   #this function checks whether a String is fully in ASCII or
not.
 if isinstance(text, unicode):
     try:
         text.encode('ascii')
     except UnicodeEncodeError:
         return False
 else:
     try:
         text.decode('ascii')
     except UnicodeDecodeError:
         return False
 return True

def main():
 firstTimeRun = True #a boolean that says whether it's the first time the loop runs
or not.
 numOfGenuineWords = {} #this dictionary contains the number of times each word
apears in each text file.
 dictionary = {} #this dictionary contains the time stemp of last change of each
file.
 tfd = {} #this dictionary contains the file names and a boolean that says whether
the file is encrypted or not.
 path = input("enter path:  ")
 while(True):
   isEnc = True #represnt a check if we shell check for encrypted file or not.
   for i in os.listdir(path):
     numOfGenuineWords.clear() #we want to check each file seperately for unique
words so we have to clear it every iteration.
     b = True #b is used as the first method of incryption detection.
```

```
    if i.endswith('.txt'):  #on every file with thy ending shell pass on to the
encryption detection code below.
      with open(path + i) as fp:
        if(firstTimeRun == False): #to speed up thy code we check:
                                   #1. is it the first time we iterate? if so lets
move on the code below.
                                   #2. did the time stemp change? if so lets check
if the text fiel is encrypted now.
          if(dictionary[i] != time.ctime(os.path.getmtime(path + i))):
            isEnc = True
          else:
            isEnc = False
        else:
          isEnc = True
        if(isEnc == True): #incase we need to run the detection:
            dictionary[i] = time.ctime(os.path.getmtime(path + i)) #update the
time stemp in the dictionary.
            totalwords=0 #total unique words in each text file
            for line in fp:
              words = line.split() #we split the line to words by the default " "
(space)
              for word in words: #lets iterate on each word...
                try: #gere we count each word and how many times it apears and the
total amount of unique words.
                  numOfGenuineWords[word] = numOfGenuineWords[word] + 1
                except:
                  numOfGenuineWords[word] = 1
                  totalwords = totalwords + 1
              if(is_ascii(line) is False):
                #if the line contains non ascii chars then the file is encrypted.
(because text files only contain ascii charcters)
                b = False
                break
            if(b is False):
              if(firstTimeRun == True):
                print("The file: "+ str(i) + " is encrypted")
              try:
                if(tfd[i] == True):
                  print("The file: "+ str(i) + " is encrypted")
              except:
                True #do nothing.
              tfd[i] = False
            else: #if all the chars are ascii, we still need to check for a cheap
encryption that only changes the order of the ABC..
              tfd[i] = True
```

```
                #the best way i found of finding encryption is filter by the present
of unique non english words. (considering there are so words
                #like "Atuda" that have no translate + human spelling mistakes I
gave a 35% thresh hold).
                notenglishwords = 0
                for eachword in numOfGenuineWords:
                  a = eachword.split(",") #incase the word had a "." added to it I
am taking it off the word so my function can detect the word.
                  if(a[0] == ","):
                    eachword = a[1]
                  else:
                    eachword = a[0]
                  boolean = isEnglishWord(eachword) #send the word to the function
above
                  if(boolean == False):
                    notenglishwords = notenglishwords + 1
                if(notenglishwords * 100 / totalwords > 35):
                  print("The file: "+ str(i) + " is encrypted")
                  tfd[i] = False



   firstTimeRun = False
if __name__ == '__main__':
 main()
```