Directory **20news-18828** contains a list of subdirectories. Each one is a newsgroup. We can check the names of these subdirectories by using *os.listdir*. We can see there are 20 newsgroups and their names:

In [ ]:

import os

# Prefix is a variable that contains the directory where each newsgroup is located

# Notice that a double \\ has to be used when \ is desired within a string

prefix = "20news-18828\\"

directories = os.listdir(prefix)

print(len(directories))

print(directories)

Inside each subdirectory, there are many files. Each file constains a message that was posted long time ago in that newsgroup. Let's see the names of the files that contain the first 5 messages, by using again *os.listdir*:

In [ ]:

fileNames = os.listdir(prefix+"alt.atheism")

print(fileNames[:<FILL IN>])

Now, let's see the contents of the first of those messages. This can be done by reading the first file and displaying it, like this:

In [ ]:

fileName = prefix+"alt.atheism\\"+"49960"

print "The name of the file that contains the first message is: {}".format(fileName)

print "================================"

# Now, we open the file for reading

myFile = open(fileName, "r")

# and it's read whole.

firstMessage = myFile.read()

myFile.close() # The file is no longer needed, so it is closed

print(firstMessage)

Now it's your turn: read and print the **second** message of newsgroup *comp.graphics* by filling in the gaps.

In [ ]:

fileNames = os.listdir(prefix+<FILL IN>)

fileName = prefix+"comp.graphics\\"+<FILL IN>

print "The name of the file that contains the first message is: {}".format(fileName)

print "================================"

# Now, we open the file for reading

myFile = open(fileName, "r")

# and it's read whole.

firstMessage = myFile.read()

myFile.close() # The file is no longer needed, so it is closed

print(firstMessage)

Now, we are going to read the first two messages from newsgroup *alt.atheism* and the first two messages from *comp.graphics*. But first, let's define a function that reads messages, in order to save ourselves some typing.

In [ ]:

def readMessage(fileName):

# Now, we open the file for reading

myFile = open(fileName, "r")

# and it's read whole.

message = myFile.read()

myFile.close() # The file is no longer needed, so it is closed

return(message)

Let's use function *readMessage* to read two messages from *alt.atheism*

In [ ]:

group = "alt.atheism\\"

# Files contains the list of files (messages) of newsgroup alt.atheism

files = os.listdir(prefix+group)

# Let's read the first two messages

message1\_aa = readMessage(prefix+group+files[<FILL IN>])

message2\_aa = readMessage(prefix+group+files[1])

# Let's put them into a list:

messages\_aa = [message1\_aa, message2\_aa]

print "We have read {} messages".format(len(messages\_aa))

print "=================================================="

print "The first message starts with these words:\n{}".format(messages\_aa[0][0:200])

print "=================================================="

print "And the second message starts with these words:\n{}".format(messages\_aa[1][0:200])

Now, read a third message and *append* it into list *messages\_aa*. Use the *apend* function

In [ ]:

message3\_aa = readMessage(prefix+group+files[<FILL IN>])

messages\_aa.<FILL IN>(message3\_aa)

print(len(messages\_aa))

We can do even better by using a loop to read the n first messages

In [ ]:

group = "alt.atheism\\"

files = os.listdir(prefix+group)

messages\_aa = [] # Initially, we have no read any message (empty list)

# Let's read 5 messages (but it could be any number)

for myFile in files[:5]:

messages\_aa.append(readMessage(prefix+group+<FILL IN>))

print "We have read {} messages".format(len(messages\_aa))

Now, use the previous code in order to read 4 messages from newsgroup *comp.graphics*

In [ ]:

group = "comp.graphics\\"

files = os.listdir(prefix+group)

<FILL IN> = [] # Initially, we have no read any message (empty list)

# Let's read 5 messages (but it could be any number)

for myFile in files[:<FILL IN>]:

messages\_cg.append(readMessage(prefix+group+<FILL IN>))

print "We have read {} messages from {}".format(len(messages\_cg), group)

Next step: let's decompose each message into words. We can do this by means of *split*. Let's see an example. Also, there are a couple of problems. What happens with words **remember** and **racing**?

In [ ]:

myMessage = "Somewhere in la Mancha, in a place whose name I do not care to remember, a gentleman lived not long ago, one of those who has a lance and ancient shield on a shelf and keeps a skinny nag and a greyhound for racing."

print(myMessage)

print("")

words = myMessage.split()

print "The list of words is:\n{}".format(words)

A way of fixing the problem of separators like "," and "." is to substitute them by spaces (" "), so that there is only one delimiter (space). We can use *replace* for this. Try to understand the following code. Do we get the correct list of words now (i.e. do **remember** and **racing** look right now?)?

In [ ]:

myMessage = "Somewhere in la Mancha, in a place whose name I do not care to remember, a gentleman lived not long ago, one of those who has a lance and ancient shield on a shelf and keeps a skinny nag and a greyhound for racing."

print(myMessage)

print("")

myMessage = myMessage.replace(",", " ")

myMessage = myMessage.replace(".", " ")

words = myMessage.split()

print "The list of words is:\n{}".format(words)

Now, we could do the same for the three messages of *alt.atheism* which are contained in list *messages\_aa*. Try to understand this code.

In [ ]:

# First, let's make a copy of the original list "messages\_aa",

# so that messages\_aa is not modified and we can use it later unchanged

import copy

messages\_aa\_copy = copy.deepcopy(messages\_aa)

# Now, we will modify messages\_aa\_copy and remove , and . from them

# Let's remove "," and "." from the first message

messages\_aa\_copy[0] = messages\_aa\_copy[0].replace(",", <FILL IN>)

messages\_aa\_copy[0] = messages\_aa\_copy[0].replace(<FILL IN>, " ")

messages\_aa\_copy[0] = messages\_aa\_copy[0].split()

# Same for the second message

messages\_aa\_copy[1] = messages\_aa\_copy[1].replace(<FILL IN>, " ")

messages\_aa\_copy[1] = messages\_aa\_copy[1].replace(".", <FILL IN>)

messages\_aa\_copy[1] = messages\_aa\_copy[1].split()

# Same for the third message

messages\_aa\_copy[2] = messages\_aa\_copy[2].replace(<FILL IN>, <FILL IN>)

messages\_aa\_copy[2] = messages\_aa\_copy[2].replace(<FILL IN>, <FILL IN>)

messages\_aa\_copy[2] = messages\_aa\_copy[2].split()

print "=================================================="

print "We can check what happenned to the first 20 words of the first message:"

print(messages\_aa\_copy[0][0:50])

print "=================================================="

But using the above code for every message is rather tedious, specially if there are many messages. Please, try to improve the above code by using a **for** loop. Reminder: range(3) == [0, 1, 2]. Please, use variable **messages\_aa** that contains a list with the three messages.

In [ ]:

# len(messages\_aa) is the number of elements in the list (i.e. the number of messages,

# 3 in this case)

for messageNumber in <FILL IN>(len(messages\_aa)):

messages\_aa[messageNumber] = messages\_aa[messageNumber].replace(",", " ")

messages\_aa[messageNumber] = messages\_aa[messageNumber].replace(".", " ")

messages\_aa[messageNumber] = messages\_aa[messageNumber].<FILL IN>

print "=================================================="

print "We can check what happened to the first 50 words of the first message:"

print(messages\_aa[0][0:50])

print "=================================================="

Finally, we are going to compute the frequency of each word for every message. We will first count the words, and then divide by the total number of words in order to get the frequency. In order to save you some work, the function *createDictionary* is provided to you.

In [ ]:

def createDictionary(words):

d = {}

for word in words:

if word in d:

d[word] = d[word] + 1

else:

d[word] = 1

return(d)

Let's see how *createDictionary* works. Please, notice that the dictionary is a list of pairs (key, count), where count is the number of times each word (key) appears in the sentence.

In [ ]:

myMessage = "Somewhere in la Mancha, in a place whose name I do not care to remember, a gentleman lived not long ago, one of those who has a lance and ancient shield on a shelf and keeps a skinny nag and a greyhound for racing."

print(myMessage)

print("")

myMessage = myMessage.replace(",", " ")

myMessage = myMessage.replace(".", " ")

words = myMessage.split()

# Here, we create a dictionary with the count of words

myDictionary = createDictionary(words)

print(myDictionary)

In order to compute the frequency, we need to divide each count by the total number of words in the message. Reminder: function *len* computes the number of elements in a list. We will multiply by 100 in order to compute percentages. Something like this:

In [ ]:

for <FILL IN> in myDictionary:

myDictionary[word] = 100\*myDictionary[word]/<FILL IN>

print(myDictionary)

Finally, compute a similar dictionary, but for the **first** message of messages\_aa.

In [ ]:

<FILL IN>

What follows is a solution with a loop, in order to process the three messages of messages\_aa. Just try to understand it, and you are done.

In [ ]:

list\_of\_dictionaries = [] # Initially, an empty list

for messageNumber in range(len(messages\_aa)):

words = messages\_aa[messageNumber]

myDictionary = createDictionary(words)

for word in myDictionary:

myDictionary[word] = 100\*myDictionary[word]/len(words)

list\_of\_dictionaries.append(myDictionary)

print(list\_of\_dictionaries)