0004

2019年4月2日 16:59

任一个英文的纯文本文件，统计其中的单词出现的个数。

来自 <https://github.com/Yixiaohan/show-me-the-code>

Regular Expression

After talking about how to count the words in the .txt, I have met another troublesome problem, which is that, I have to distinguish the punctuation and the letter from the .txt. Coincidentally, I just learned regular expression in Principal of Compilers, so, I want to use RE to solve this problem
I have to review the class on Coursera(py4e) to get the idea of how to use RE in python.(I would like to take some notes this time)
After review, I find a brand new world, in re library, we have a method which is called re.findall(pattern, string) with this method, we can easily use RE to find all words without punctuation.
BTW, the findall method return a list.

```
pieces=re.findall('[a-zA-Z]+',line)
```

whole program:

```
importre
#path=input("Enterfilename:")
#iflen(path)<1:
path="demo.txt"
xFile=open(path)
newDict=dict()
forlineinxFile:
iflen(line)<1:
continue
pieces=re.findall('[a-zA-Z]+',line)
forwordinpieces:
word=word.lower()
newDict[word]=newDict.get(word,1)+1

maxKey=None
maxVal=None
forkey,valinnewDict.items():
ifmaxVal==None:
maxVal=val
maxKey=key
elifval>maxVal:
maxVal=val
maxKey=key
print(maxKey,maxVal)
```

How to count the words

For dictionary in python:

at first, we could create a dictionary in this way:

```
newDict = dict()
```

and every dictionary has 2 element: key and value
every key coresspond to a value
and key can't be repeated in one dictionary

you can store elements to the dictionary like this:

```
newDict = { a:1, b:2, c:3}
```

actually, not exactly that key should be a string or char, it could be everything, in the same way, value are not supposed to be an integer.
But we usually use dictionary in this way, which key is string and value is integer.

In this case, we may use dictionary to count every words in a .txt

to count word, we have two ways,

a. every time we want to add a new key(which is a word in the passage) to the dictionary, we check if there is already had a key in the dictionary, if there is, we add 1 to the correspond value, if not, we set value to 1.
Compare to the method below, I don't want to realize this in detail, because this method didn't show the advantages of Python.

For now(2/Apr/19) I think the method below is *Pythonic*

b. there is also a very efficient way to do this, only 1 line of code is enough.(Amazing Python!)

```
newDict[word]=newDict.get(word,1)+1
```

I will talk about the method get() in detail:
let's see its parameters: dictionary.get(key, defualt = None) this means that, get will search through the dictionary, to find wether there is a "key", if the key already there, it will return the value of the key, if not, it will return the default value which is None in this case.

Take the code upon in blue as an example, it will search through the dictionary and if find "word", it will return its value and add 1, if not, it will return to 1.