目录

关于作者

廖雪峰 北京 朝阳区
+加关注

chardet

阅读: 44890

字符串编码一直是令人非常头疼的问题，尤其是我们在处理一些不规范的第三方网页的时候。虽然Python提供了Unicode表示的`str`和`bytes`两种数据类型，并且可以通过`encode()`和`decode()`方法转换，但是，在不知道编码的情况下，对`bytes`做`decode()`不好做。

对于未知编码的`bytes`，要把它转换成`str`，需要先"猜测"编码。猜测的方式是先收集各种编码的特征字符，根据特征字符判断，就能有很大概率"猜对"。

当然，我们肯定不能从头自己写这个检测编码的功能，这样做费时费力。chardet这个第三方库正好就派上了用场。用它来检测编码，简单易用。

安装chardet

如果安装了Anaconda，chardet就已经可用了。否则，需要在命令行下通过pip安装：

```
$ pip install chardet
```

如果遇到Permission denied安装失败，请加上sudo重试。

使用chardet

当我们拿到一个`bytes`时，就可以对其检测编码。用chardet检测编码，只需要一行代码：

```
>>> chardet.detect(b'Hello, world!')
{'encoding': 'ascii', 'confidence': 1.0, 'language': ''}
```

检测出的编码是`ascii`，注意到还有个`confidence`字段，表示检测的概率是1.0（即100%）。

我们来试试检测GBK编码的中文：

```
>>> data = '离离原上草，一岁一枯荣'.encode('gbk')
>>> chardet.detect(data)
{'encoding': 'GB2312', 'confidence': 0.7407407407407407, 'language': 'Chinese'}
```

检测的编码是`GB2312`，注意到GBK是GB2312的超集，两者是同一种编码，检测正确的概率是74%，`language`字段指出的语言是`'Chinese'`。

对UTF-8编码进行检测：

```
>>> data = '离离原上草，一岁一枯荣'.encode('utf-8')
>>> chardet.detect(data)
{'encoding': 'utf-8', 'confidence': 0.99, 'language': ''}
```

我们再试试对日文进行检测：

```
>>> data = '最新の主要ニュース'.encode('euc-jp')
>>> chardet.detect(data)
{'encoding': 'EUC-JP', 'confidence': 0.99, 'language': 'Japanese'}
```

可见，用chardet检测编码，使用简单。获取到编码后，再转换为`str`，就可以方便后续处理。

chardet支持检测的编码列表请参考官方文档Supported encodings。

小结

使用chardet检测编码非常容易，chardet支持检测中文、日文、韩文等多种语言。

读后有收获可以请作者喝咖啡，读后有疑问请加群讨论：

还可以分享给朋友：

分享到微博

上一页 下一页

评论

试试

不会TC的猫 created at October 16, 2018 11:36 AM, Last updated at May 24, 2019 2:53 PM

> data = '灰烬之灵'.encode('gbk') chardet.detect(data) {'encoding': None, 'confidence': 0.0, 'language': None}

又下眉头
Created at May 24, 2019 2:53 PM

哈哈哈，火猫哭晕在厕所

全部讨论 回复

不明白

我很有素质 created at May 4, 2019 7:13 PM, Last updated at May 13, 2019 7:19 PM

都

```
encode的 了
```

为何还检测？ 多此一举？

快没时间啦
Created at May 13, 2019 12:18 PM

举例只是为了证明chardet的准确性。。。

廖雪峰
Created at May 13, 2019 7:19 PM

换个写法：

```
>>> data = b'\xc0\xeb\xc0\xeb\xd4\xad\xc9\xcf\xb2\xdd\xa3\xac\xd2\xbb\xcb\xea\xd2\xbb\xbf\xdd\xc8\xd9'
>>> chardet.detect(data)
{'encoding': 'GB2312', 'confidence': 0.7407407407407407, 'language': 'Chinese'}
```

全部讨论 回复

这是啥情况？？？？？

-文_少- created at January 17, 2018 10:27 AM, Last updated at December 1, 2018 2:13 PM

data='天王盖地虎'.encode('utf-8')

chardet.detect(data) Out[40]: {'confidence': 0.9690625, 'encoding': 'utf-8', 'language': ''}

data='天王盖地虎'.encode('gbk')

chardet.detect(data) Out[42]: {'confidence': 0.0, 'encoding': None, 'language': None}

data='天王盖地虎'.encode('gb2312')

chardet.detect(data) Out[44]: {'confidence': 0.0, 'encoding': None, 'language': None}

氓不氓
Created at January 20, 2018 2:11 AM

我的猜测是：

1. 因为中文的古诗词与人们平时说的语言区别比较大，所以导致它的猜测算法出错。随便试了几个随机中文的GBK编码，都猜测不出来。
2. 样本太短

```
>>> data = '淆始贪级耻擅哑鸡铁狮力贵拓碌抽憎贯坚处议税往蛙躲卓完银范盒屑私计瀑诉下备图拜多蕴瞻胶弄袋跑港钻滥拧贼决项挟堵尿尸枣旋策分蜘待侵荠鸳驮竣茧谐掺扣驳殖辈卖套纱耽洁滨晰监纸宽柄啥寓榕砸博渔舶翁叶碾记奶草媚语'.encode('gbk')
>>> chardet.detect(data)
{'encoding': 'GB2312', 'confidence': 0.2606310013717421, 'language': 'Chinese'}
>>> data = '淆始贪级耻擅哑鸡铁狮力贵拓碌'.encode('gbk')
>>> chardet.detect(data)
{'encoding': None, 'confidence': 0.0, 'language': None}
```

廖雪峰
Created at January 20, 2018 4:57 PM

先注意你的py文件的编码对不对

我的天啊又要取名
Created at March 8, 2018 4:25 PM

加个小鸡炖蘑菇就可以了

颜成子由
Created at March 26, 2018 11:02 AM

```
data='天王盖地虎,小鸡炖蘑菇'.encode('GBK')
chardet.detect(data)
{'language': 'Chinese', 'confidence': 0.7407407407407407, 'encoding': 'GB2312'}
data='天王盖地虎'.encode('GBK')
chardet.detect(data)
{'language': None, 'confidence': 0.0, 'encoding': None}
```

还真是啊！

当垆人似月
Created at April 10, 2018 8:38 PM

经试验，和字数有关系，字数越多越容易识别出来

孙子文02463
Created at December 1, 2018 2:13 PM

我在用requests.get爬取一个pdf网页时，返回的内容却不知道是什么东西。我用chardet.detect()去检查，结果却是{'encoding': None, 'confidence': 0.0, 'language': None}。请问这是怎么回事啊？（附部分开头内容： b'%PDF-1.7\r%\x80\x84\x88\x8c\x90\x94\x98\x9c\xa0\xa4\xa8\xac\xb0\xb4\xb8\xbc\xc0\xc4\xc8\xcc\xd0\xd4\xd8\xdc\xe0\xe4\xe8\xec\xf0\xf4\xf8\xfc\r\r912 0 obj\r<< /T 1021131 /L 1039524 /Linearized 1 /E 220286 /O 916 /H\r[3537 941\r] /N 25\r>>\rendobj xref\r912 159\r0000000044 00000 n\r\n0000004478 00000 n\r\n0000004834 00000 n\r\n0000004863 00000 n\r\n0000004961 00000 n\r\n0000005347 00000 n\r\n0000008107 00000 n\r\n0000010187 00000 n\r\n0000012031 00000 n\r\n0000014019 00000 n\r\n0000016241 00000 n\r\n0000018381 00000 ）

全部讨论 回复

打卡

果味lll0 created at September 20, 2018 3:11 PM, Last updated at September 20, 2018 3:11 PM

data = '出现'.encode('gbk') chardet.detect(data) {'encoding': 'KOI8-R', 'confidence': 0.38398486178080915, 'language': 'Russian'}

data = '出现出现出现出现出现出现出现出现出现出现出现出现出现出现出现出现出现出现出现'.encode('gbk') chardet.detect(data) {'encoding': 'GB2312', 'confidence': 0.99, 'language': 'Chinese'}

全部讨论 回复

总结

我是世外大帝 created at December 19, 2017 2:19 PM, Last updated at March 26, 2018 11:01 AM

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Created by TaoYuan on 2017/12/19 0019.
# @Link    : http://blog.csdn.net/lftaoyuan
# Github   : https://github.com/seeways
# @Remark  : Python学习群: 315857408
"""
检测字符编码库
chardet.detect(content)
    :param byte_str:     The byte sequence to examine.
    :type byte_str:      ``bytes`` or ``bytearray``

https://chardet.readthedocs.io/en/latest/supported-encodings.html
"""
import chardet as chardet


def check_char(content):
    return chardet.detect(content)

# 默认只接受byte_str，否则返回TypeError
```

Read More

颜成子由
Created at March 26, 2018 11:01 AM

> # utf-8编码: 英文还是ascii，中文是utf-8了，但是language没有指出，是因为utf-8适用的太多了

貌似utf-8是不分语言的吧

全部讨论 回复

注意细节

锁不住梦想的盒子 created at March 7, 2018 11:19 AM, Last updated at March 7, 2018 11:19 AM

import chardet

全部讨论 回复

解码

Leonardo_6666 created at November 27, 2017 7:41 PM, Last updated at November 27, 2017 7:41 PM

data = '最新の主要ニュース'.encode('euc-jp') data.decode((chardet.detect(data))['encoding']) '最新の主要ニュース'

全部讨论 回复

发表评论

登录后发表评论