数据的读取和分析

1. 数据的读取

用 pandas 来读取大赛给的 csv 文件

第一步是库的导入

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

第二步读取数据

```
In [2]: df_train=pd.read_csv("train_set.csv", sep='\t', nrows=50000)
        # df_train=pd.read_csv("train_set.csv", sep='|t')#(200000,2)200000条新闻，一个为label，一个为新闻（不过都用数字代替了）
        # df_test=pd.read_csv("test_a.csv", sep='|t')#(50000,1)测试集有50000条新闻组成

        df_train.head()
```

用 head（n）显示前 n 行可得（也可用 tail(n)显示最后几行）

Out[2]:

	label	text
0	2	2967 6758 339 2021 1854 3731 4109 3792 4149 15...
1	11	4464 486 6352 5619 2465 4802 1452 3137 5778 54...
2	3	7346 4068 5074 3747 5681 6093 1777 2226 7354 6...
3	2	7159 948 4866 2109 5520 2490 211 3956 5520 549...
4	3	3646 3055 3055 2490 4659 6065 3370 5814 2465 5...

2. 数据分析

从上述显示的图片我们可以知道我们得到的 dataframe 主要由以下两部分组成，一个是 label，另一部分是 text（词或符号都用数字做了代替），我们可通 dataframe["text"]获得其中的文本内容，dataframe["label"]获得其中的标签内容，下图为截取为前 100 个文本内容/标签内容

```
In [5]: df_train["text"][:100]
```

```
Out[5]: 0     2967 6758 339 2021 1854 3731 4109 3792 4149 15...
        1     4464 486 6352 5619 2465 4802 1452 3137 5778 54...
        2     7346 4068 5074 3747 5681 6093 1777 2226 7354 6...
        3     7159 948 4866 2109 5520 2490 211 3956 5520 549...
        4     3646 3055 3055 2490 4659 6065 3370 5814 2465 5...
                                    ...
        95    6065 3370 1519 499 7157 5620 3317 1679 3270 12...
        96    7256 134 7539 7543 3137 3335 2695 669 3068 333...
        97    7160 5087 2400 4411 7044 1519 7039 2265 408 67...
        98    507 6981 2999 62 3080 6704 5310 2400 4411 1099...
        99    3870 3641 6248 913 1866 7495 3648 5370 4333 45...
        Name: text, Length: 100, dtype: object
```

```
In [6]: df_train["label"][:100]
```

```
Out[6]: 0      2
        1     11
        2      3
        3      2
        4      3
              ..
        95     7
        96     1
        97     2
        98     2
        99     4
        Name: label, Length: 100, dtype: int64
```

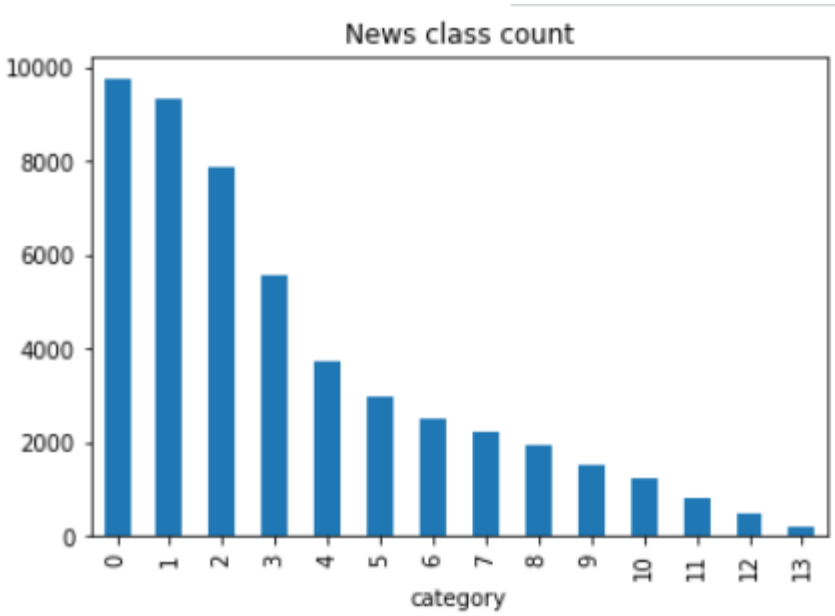
接下来我们对标签进行一个解析，用 unique 函数我们可得有多少个分类

```
In [25]: #可分为14个类
         df_train["label"].unique()
```

```
Out[25]: array([ 2, 11,  3,  9, 10, 12,  0,  7,  4,  1,  6,  5,  8, 13],
               dtype=int64)
```

然后我们可以观看每一类的个数，画出一个直方图

```
In [6]: #样本个数，样本不均衡
        df_train['label'].value_counts().plot(kind='bar')
        plt.title('News class count')
        plt.xlabel("category")
```



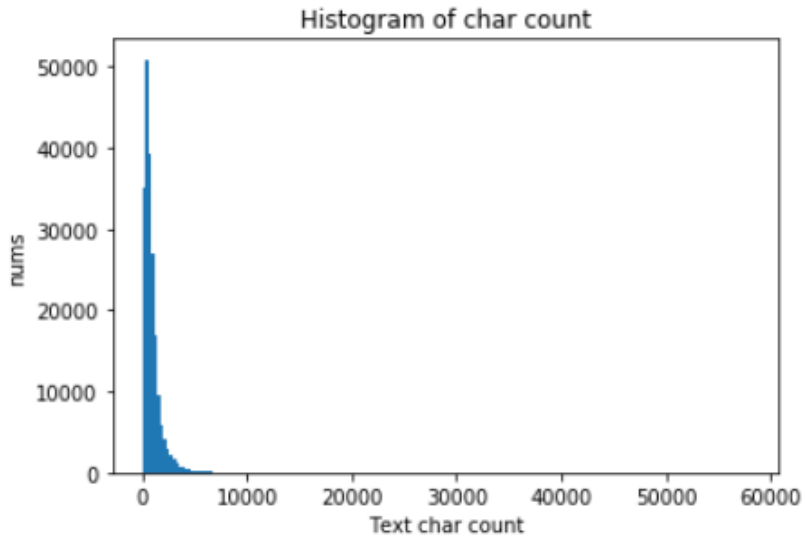
我们可以观察到各个类别非常不均匀，接下来我们可以对文本进行解析，首先我们可以计算下文本的长度。

In [5]:
```
#统计长度数据（count指代统计多少个，mean指均值，std指标准差，min，max分别指最大最小值）
%pylab inline
df_train['text_len'] = df_train['text'].apply(lambda x: len(x.split(' ')))
print(df_train['text_len'].describe())
```

```
Populating the interactive namespace from numpy and matplotlib
count    50000.000000
mean       904.589900
std        961.345267
min          2.000000
25%        373.000000
50%        670.500000
75%       1123.000000
max      44665.000000
Name: text_len, dtype: float64
```

我们可以发现，文本最长可达 44665，最短只有 2。紧接着我们可以再对每个文本的词长度画个直方图来统计一下

In [21]:
```
#画出一个直方图，统计长度（bins控制区间长度，0-200在一条柱里面，200-400在另一条柱里，其中横坐标代表）
plt.hist(df_train['text_len'], bins=200)
# plt.hist(df_train['text_len'])
plt.xlabel('Text char count')
plt.ylabel("nums")
plt.title("Histogram of char count")
```



我们还可统计所有文本加起来一下总共有多少个词

```
#统计各个词出现的次数
from collections import Counter
all_lines = ' '.join(list(df_train['text']))
# all_lines2 = ' '.join(list(df_train['text'][30000:]))
# all_lines=all_lines1+" "+all_lines2
word_count = Counter(all_lines.split(" "))
word_count = sorted(word_count.items(), key=lambda d:d[1], reverse = True)

print(len(word_count))

print(word_count[0])
#('3750', 1863795)前50000个文档出现的个数
print(word_count[1])
#('648', 1225648)
print(word_count[2])
#('900', 810253)
```

```
6180
('3750', 1863795)
('648', 1225648)
('900', 810253)
```

今天打卡到此结束，谢谢（2020/07/22）。