作业五：实战NLP——讽刺检测

姓名	学号
张三	0000000000
李四	1111111111

要求：

完成以下notebook，Sarcasm Detection数据集的下载代码已经给出，请同学们自行完成数据处理和训练过程。
作业提交 jupyter notebook 文件。

近年来，以社交媒体为媒介的电子新闻已成为信息消费的主要来源之一。许多媒体公司正在使用创造性的方法来增加帖子的浏览量。其中一种方法是使用讽刺标题作为用户点击的诱饵。

一个能够预测一篇新闻的标题是否具有讽刺意味的模型对于媒体公司来说很有用，可以方便他们通过一些策略分析季度收益。此外，从读者的角度来看，搜索引擎可以利用这些讽刺的信息，并根据读者的偏好，向他们推荐类似的文章。

数据集

用于讽刺检测的新闻标题数据集，该数据集来自两个新闻网站，theonion.com和huffingtonpost.com。以往的研究大多使用基于标签监督收集的Twitter数据集，但这些数据集在标签和语言方面存在噪声。此外，许多tweet是对其他tweet的回复，检测其中的讽刺需要上下文tweet的信息。这个新的数据集与现有的Twitter数据集相比有以下优点: 由于新闻标题是由专业人士以正式的方式编写的，所以没有拼写错误和非正式用法。这减少了稀疏性。 此外，由于TheOnion的唯一目的是发布讽刺的新闻，与Twitter数据集相比，标签的质量要更高，噪音小得多。与回复其他推文的推文不同，新闻标题是独立的。这将有助于我们梳理出真正的讽刺元素

下载和缓存数据集

In [ ]:

```
!pip install wget
```

In [ ]:

```
import wget

url = 'https://storage.googleapis.com/kaggle-data-sets/1222487/2040984/compressed/Sarcasm_Headl
wget.download(url, '../data')
!zip ../data/Sarcasm_Headlines_Dataset.json ../data/Sarcasm_Headlines_Dataset.json.zip
```

读取并查看数据集

In [1]:

```
import json

data_raw = [json.loads(line) for
        line in open('../data/Sarcasm_Headlines_Dataset.json', 'r')]
```

In [2]:

```
print(len(data_raw))
print(data_raw[0])
```

```
26709
{'article_link': 'https://www.huffingtonpost.com/entry/versace-black-code_us_5861fbe
fe4b0de3a08f600d5', 'headline': "former versace store clerk sues over secret 'black
code' for minority shoppers", 'is_sarcastic': 0}
```

可以看到数据集一共包含了26709条新闻标题以及对应的标签。**这里先忽略数据集里的'article_link'属性**

数据集的 **'headline'** 给出的是新闻标题，而 **'is_sarcastic'** 给出的是该新闻标题是否是讽刺性的标签。

下面看一下数据集里所有 **'headline'** 的长度统计数据。

In [7]:

```
max_length, min_length = 0, 0x3f3f3f
sum_length = 0
length_distribute = [0] * 1000
for i in range(len(data_raw)):
    l = len(data_raw[i]['headline'])
    sum_length += l
    max_length = max(max_length, l)
    min_length = min(min_length, l)
    length_distribute[l] += 1

avg = sum_length / len(data_raw)
print(f'max length: {max_length} \nmin length: {min_length} \navg length: {avg}')

print(length_distribute[:max_length + 1])
```

```
max length: 254
min length: 7
avg length: 60.910591935302705
[0, 0, 0, 0, 0, 0, 0, 1, 4, 4, 2, 13, 10, 19, 38, 30, 33, 43, 52, 49, 63, 68, 74, 6
9, 91, 117, 93, 131, 152, 146, 164, 170, 202, 210, 216, 255, 228, 235, 250, 315, 33
1, 316, 371, 356, 364, 370, 420, 403, 435, 436, 475, 497, 488, 460, 490, 502, 553, 5
20, 543, 530, 550, 577, 645, 581, 600, 673, 575, 575, 532, 570, 532, 529, 457, 497,
431, 439, 423, 387, 353, 338, 312, 254, 292, 256, 250, 213, 210, 194, 185, 155, 146,
141, 122, 113, 106, 89, 91, 71, 67, 66, 67, 59, 50, 49, 43, 33, 39, 33, 21, 22, 29,
24, 23, 31, 12, 16, 17, 15, 8, 13, 7, 8, 7, 9, 9, 6, 3, 7, 5, 2, 2, 5, 1, 2, 1, 0,
1, 4, 1, 3, 0, 2, 1, 4, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1]
```

可以看到最长的标题到达了254个单词之多，而最短只有7个单词。平均长度为60。

数据处理（自行完成）

请同学自己完成数据处理过程。

要求：

26709条新闻的前20000个作为训练集，后6709条作为测试集，不设验证集。 最后在测试集上测试自己模型的最终结果。

训练（自行完成）

请同学自己完成从**定义网络、定义损失函数、定义优化器到进行训练等一系列深度学习流水线。**

提交方式

包含训练结果的Jupyter notebook文件请命名为 work2_<组长姓名>_<组长学号>.ipynb 发送到邮箱 archie98@qq.com (mailto:archie98@qq.com)