一、作业内容

根据 16 本金庸武侠小说来计算中文的平均信息熵。

二、实验原理

1. 熵和信息熵

一、熵

熵，泛指某些物质系统状态的一种量度，某些物质系统状态可能出现的程度。亦被社会科学用以借喻人类社会某些状态的程度。熵的概念是由德国物理学家克劳修斯于 1865 年所提出。最初是用来描述“能量退化”的物质状态参数之一，在热力学中有广泛的应用。但那时熵仅仅是一个可以通过热量改变来测定的物理量，其本质仍没有很好的解释，直到统计物理、信息论等一系列科学理论发展，熵的本质才逐渐被解释清楚，即，熵的本质是一个系统“内在的混乱程度”。它在控制论、概率论、数论、天体物理、生命科学等领域都有重要应用，在不同的学科中也有引申出的更为具体的定义，按照数理思维从本质上说，这些具体的引申定义都是相互统一的，熵在这些领域都是十分重要的参量。

二、信息熵

信息熵的定义公式：

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x)$$

并且规定：0 log（0） = 0。

信息熵的三个性质：

信息论之父克劳德·香农给出的信息熵的三个性质：

1. 单调性，发生概率越高的事件，其携带的信息量越低；

2. 非负性，信息熵可以看作为一种广度量，非负性是一种合理的必然；

3. 累加性，即多随机事件同时发生存在的总不确定性的量度是可以表示为各事件不确定性的量度的和，这也是广度量的一种体现。

香农从数学上严格证明了满足上述三个条件的随机变量不确定性度量函数具有唯一形式：

$$H(X) = -C \sum_{x \in \chi} p(x) \log p(x)$$

三、实验过程及结果

首先将 16 本小说合并为一本 txt，并以 ANSI 形式保存，文件名为 data，以便后续计算处理。读入 data 数据，并进一步读入停词 cn_stopwords.txt 文件，获取停词内容。去除文本中的除中文以外的所有字符，进一步删去文中的停词内容。

```
corpus=[]
with open('data.txt', "r", encoding="ANSI") as file:
    text = [line.strip("\n").replace("\u3000", "").replace("\t", "") for line in file][3:]
    corpus += text

with open('cn_stopwords.txt',"r", encoding="ANSI") as file:
    words= [line.strip() for line in file.readlines()]


sub= u'[a-zA-Z0-9’!"#$%&\'()*+,-./:: ;「<=>?@, 。?★、…【】《》？“”‘’！[\\]^_`{|}~]+'
for j in range(len(corpus)):
    corpus[j] = re.sub(sub, "", corpus[j])
```

```
new_corpus = []
character_count = 0
for text in corpus:
    new_words = []
    split_words = list(jieba.cut(text))
    for word in split_words:
        if word not in words:
            new_words.append(word)
    character_count += len(''.join(map(str, new_words)))
    new_corpus.append(''.join(map(str, new_words)))
```

字的信息熵计算及结果：

```
token = []
for para in corpus:
    token += jieba.lcut(para)
token_num = len(token)
ct = Counter(token)
vocab1 = ct.most_common()
entropy_1gram = sum([-(eve[1]/token_num)*math.log((eve[1]/token_num),2) for eve in vocab1])

print("词库总词数：", token_num, " ", "不同词的个数：", len(vocab1))
print("出现频率前10的1-gram词语：", vocab1[:10])
print("entropy_1gram:", entropy_1gram)
```

词库总词数： 4263939
不同词的个数： 160454
出现频率前 10 的词语：
[(’的’, 115926), (’了’, 105066), (’他’, 64457), (’是’, 63941), (’道’, 58517), (’我’, 57030), (’你’, 55814), (’在’, 43917), (’也’, 32170), (’这’, 31542)]
信息熵: 12.172445692705105

词的信息熵计算及结果：

```
token_2gram = []
for para in new_corpus:
    cutword_list = jieba.lcut(para)
    token_2gram += combine2gram(cutword_list)
# 2-gram的频率统计
token_2gram_num = len(token_2gram)
ct2 = Counter(token_2gram)
vocab2 = ct2.most_common()
# print(vocab2[:20])
# 2-gram相同句首的频率统计
same_1st_word = [eve.split("s")[0] for eve in token_2gram]
assert token_2gram_num == len(same_1st_word)
ct_1st = Counter(same_1st_word)
vocab_1st = dict(ct_1st.most_common())
vocab1=ct_1st.most_common()
entropy_2gram = 0
for eve in vocab2:
    p_xy = eve[1]/token_2gram_num
    first_word = eve[0].split("s")[0]
    # p_y = eve[1]/vocab_1st[first_word]
    entropy_2gram += -p_xy*math.log(eve[1]/vocab_1st[first_word], 2)
print("词库总词数：", token_2gram_num, " ", "不同词的个数：", len(vocab1))
print("出现频率前10的2-gram词语：", vocab1[:10])
print("entropy_2gram:", entropy_2gram)
```

词库总词数： 2734861
不同词的个数： 174120
出现频率前 10 的词语：
[('道', 54530), ('说', 17883), ('便', 16017), ('说道', 12967), ('中', 12555), ('听', 10542), ('见', 9944), ('韦小宝', 9768), ('一个', 9560), ('一声', 6406)]
词的信息熵：6.240190512472567