Report of Deep learning for Natural language Processing

唐宗润 ZY2303211

Abstract

介绍了 Zipf's Law，并基于金庸的武侠著作进行验证

Introduction

Zipf's Law，即齐夫定律，是描述自然语言中单词出现频率分布的一种经验定律。它由美国语言学家乔治·金斯利·齐夫（George Kingsley Zipf）在 20 世纪初提出，并被广泛应用于语言学、信息论和其他领域的研究中。

根据齐夫定律，如果将一个语言中所有单词按照它们在文本中出现的频率排序，然后将它们的排名（即出现频率从高到低的顺序）乘以它们的出现频率，得到的结果将近似为一个常数。换句话说，排名越高的单词，其出现频率越低，但它们的乘积近似相等

齐夫定律表明，自然语言中的单词出现频率呈现出一种明显的幂律分布特征，即排名与频率之间存在着负相关关系。这意味着在任何给定的文本中，少数单词出现频率非常高，而大多数单词出现频率非常低。这个规律不仅适用于英语，还适用于许多其他自然语言，以及其他领域中的一些现象，如城市人口分布、互联网流量分布等。

Methodology

首先对数据进行预处理，将文件中的空格，换行符等无意义且高频字符进行去除。

```
for novel_file in novel_files:
    novel_file_name = 'D:/zongruntang/nlp/jyxstxtqj_downcc.com/' + novel_file + '.txt'
    with open(novel_file_name, 'r', encoding='ANSI') as f:
        novel_text = f.read()
        novel_text = novel_text.replace( _old: '本书来自www.cr173.com免费txt小说下载站\n更多更新免费电子书请关注www.cr173.com', _new: '')
        novel_text = novel_text.replace( _old: u'\u3000', _new: u'').replace( _old: '\n', _new: '').replace( _old: '\r', _new: '').replace( _old:
        novel_text = novel_text.replace( _old: '[', _new: '').replace( _old: ']', _new: '').replace( _old: ', ', _new: '').replace( _old: '。', _new
        novels.append(novel_text)
```

其次，使用 jieba 库进行分词

```
# 使用jieba进行分词
novel_text_combined = ''.join(novels) #novels 中的所有
seg_lists = ' '.join(jieba.cut(novel_text_combined))
```

需要注意的是，#jieba.cut() 返回的结果是一个迭代器对象，而不是字符串，因此无法直接进行解码操作.将分词结果转换为空格分隔的字符串来实现迭代器对象到字符串的转换。

接下来，统计词频后进行排序。

```
# 统计词频

word_freq = {}
for word in seg_lists:
    if  word.strip(): #去除空字符串
        if word not in word_freq:
            word_freq[word] = 1
        else:
            word_freq[word] += 1
```

最后进行绘图

```python
# 绘制Zipf's Law
plt.figure(figsize=(10, 6))
plt.plot( *args: ranks, frequencies,  linestyle='-')
plt.xlabel('Rank')
plt.xscale('log')
plt.yscale('log')
plt.ylabel('Frequency')
plt.title("Zipf's Law")
plt.xlim( *args: 1, 10000)  # 设置横坐标范围为1到100
plt.grid(True)
plt.show()
```

Conclusion

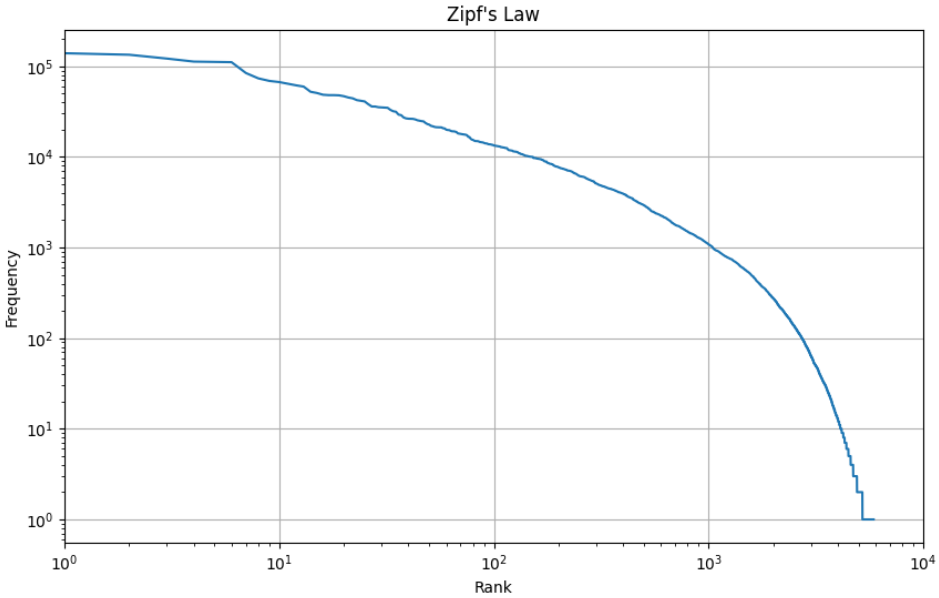

图 1 排名和频率的关系

如图 1，排名和频率呈现负相关。