南京邮电大学

实 验 报 告

（ 2023 / 2024 学年第 1 学期）

课程名称 自然语言处理

实验名称 语料预处理和语料库构建与应用

实验日期 2023/12/1

指导教师 朱博

学生姓名 单家俊 学号 B21080526

专 业 人工智能 年级 大三

实 验 报 告

一、实验目的和要求

1、掌握语料库的构建方法。
2、掌握语料库的预处理技术。
3、掌握语料库的简单分析方法

二、实验环境(实验设备)

PC、PyCharm

三、实验原理及内容

1、构建金庸武侠小说集语料库

（1）收集金庸武侠小说的 txt 文本文件，完成数据采集和预处理，获取保存的文件列表。

使用的核心函数有：nltk.corpus 中的 PlaintextCorpusReader

数据采集和预处理代码：

```python
# -*- coding: gbk -*-

import jieba
import nltk
import re
from nltk.book import *
from nltk.corpus import PlaintextCorpusReader
import matplotlib.pyplot as plt


corpus_root = 'D:/QQ/1413679561/FileRecv/data'
# file_pattern = r'.*\.txt'
corpus_reader = PlaintextCorpusReader(corpus_root, fileids: '.*')
file_list = corpus_reader.fileids()
print(file_list)

# 加载停用词列表
stop_words = []
path = 'E:/NLP/stopword.txt'
for line in open(path, encoding='utf8'):
    # print(line)
    line = line.strip()
    stop_words.append(line)


# 去除停用词
result = []
with open(file=r'D:\QQ\1413679561\FileRecv\data\金庸-神雕侠侣.txt', mode='r', encoding="gbk") as f:
    content = f.read()
    cleaned_data = ''.join(re.findall( pattern: '[\u4e00-\u9fa5]', content))
    content_list = jieba.lcut(cleaned_data, cut_all=False)
    for word in content_list:
        if word not in stop_words:
            result.append(word)
text = ''.join(result)
# print(text)

with open(file='金庸-神雕侠侣1.txt', mode='w') as f:
    f.write(text)

with open('金庸-神雕侠侣1.txt', 'r', encoding='gbk') as f:
    str = f.read()
    # len(set(str))
    # len(str)/len(set(str))

# cleaned_data = ''.join(re.findall('[\u4e00-\u9fa5]', str))
wordlist = jieba.lcut(str)
text = nltk.Text(wordlist)
print(text)
```

结果：

```
['金庸-书剑恩仇录.txt', '金庸-侠客行.txt', '金庸-倚天屠龙记.txt', '金庸-天龙八部.txt', '金庸-射雕英雄传.txt', '金庸-白马啸西风.txt', '金庸-碧血剑.txt', '金庸-神雕侠侣.txt', '金庸-笑傲江湖.txt', '金庸-越女剑.txt', '金庸-连城诀.txt', '金庸-雪山飞狐.txt', '金庸-飞狐外传.txt', '金庸-鸳鸯刀.txt', '金庸-鹿鼎记.txt']
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ADan\AppData\Local\Temp\jieba.cache
Loading model cost 0.573 seconds.
Prefix dict has been built successfully.
<Text: 全本 全集 精校 小说 更 资源 下载 声明...>
杨过 小龙女 赵志敬 周伯通 还 笑 武三通 武敦儒 洪七公 耶律齐
```

（2）选其中一本小说，利用 NLTK 中的基本函数，在你感兴趣的小说中，搜索相似词语、指定词语、搭配词语、查询文本词汇频数分配等，画出词汇分布离散图。（参考教材 21 页-22 页（1）-（11）内容，每条都要做）

内容 1：

```
# (1)
text.similar(word='李莫愁',num=10)
print('\n')
```

```
杨过 小龙女 赵志敬 周伯通 还 笑 武三通 武敦儒 洪七公 耶律齐
```

内容 2：

```
# (2)
text.concordance(word='侠',width=30,lines=3)
print('\n')
```

```
Displaying 3 of 54 matches:
 江湖 上 赫赫有名 神雕 侠 四川人 问道 叫作 神雕
 四川人 问道 叫作 神雕 侠 汉子 道 这位 大侠 行侠
小 功劳 那天 晚上 神雕 侠 突然 来到 临安 带领 伙
```

内容 3：

```
# (3)
text.collocations()
print('\n')
```

```
潇湘子 尹克西; 丘处机 王处一; 潇湘子 尼摩星; 甄志丙 赵志敬; 七十二路 空明拳; 鞠躬尽瘁 死而后已; 王志坦 宋德方; 武敦儒
武修文; 全真教 第三代; 老顽童 周伯通; 李志常 王志坦; 大力神 史季强; 马真人 丘真人; 生死相许 天南地北; 小龙女 微微一笑;
郝大通 孙不二; 广宁子 郝大通; 为国为民 侠之大者; 马道长 丘道长; 倒大出 意料之外
```

内容 4：

```
# (4)
text.common_contexts(['杨过','小龙女'])
print('\n')
```

```
去_道 罢_道 听_说 问_道 是不是_道 听_道 师父_道 干什么_道 说_道 没有_道 知道_道 好_道 人_道 出来_道 道_道
此时_已 走_道 死_道 只_一人 见_脸上
```

内容 5：

```
# (5)
print(len(str))
print('\n')
```

```
700054
```

内容 6：

```
# (6)
print(set(str))
print('\n')
```

```
{'根', '皿', '迹', '猫', '埋', '貂', '澈', '满', '牢', '临', '淘', '邀', '拳', '抑', '忸', '桌', '盟', '仄',
 '传', '服', '险', '扰', '淋', '絮', '廿', '揖', '虬', '前', '屡', '嚼', '焉', '族', '内', '砸', '渺', '宇
', '话', '髭', '稔', '机', '是', '除', '寒', '池', '程', '呛', '蜈', '介', '瑶', '伍', '袋', '式', '拿', '
嗓', '晌', '目', '尔', '漉', '却', '稻', '脾', '仗', '骄', '诫', '絮', '健', '咋', '旋', '王', '吱', '典',
'然', '要', '居', '肋', '孽', '跄', '缪', '玖', '预', '赎', '软', '炆', '傍', '靠', '萦', '测', '纶', '闹',
 '竖', '套', '咽', '单', '疏', '镀', '筋', '减', '污', '岩', '搭', '迥', '翔', '冒', '凰', '渖', '憔', '产
', '驹', '喝', '脐', '陪', '堆', '绊', '斗', '潼', '轸', '霞', '剖', '咫', '倜', '蕾', '梅', '角', '凸', '
捷', '著', '骨', '姨', '够', '控', '赶', '越', '眶', '胎', '繁', '兆', '疤', '贝', '逢', '责', '伯', '家',
'镂', '搓', '溅', '敞', '蜮', '豸', '聋', '洵', '灶', '膝', '溪', '搌', '宦', '汤', '珑', '议', '射', '床',
 '孳', '敞', '敖', '佳', '川', '抖', '症', '跃', '磊', '绌', '灵', '茧', '薛', '括', '巷', '翌', '予', '哀
', '樊', '馈', '缴', '蟋', '餮', '功', '们', '雨', '涕', '芥', '缠', '霁', '体', '沿', '良', '酷', '虚', '
勒', '郡', '佬', '掌', '篁', '鸣', '聘', '寡', '座', '蟹', '渗', '圃', '艺', '约', '嗣', '朦', '审', '扼',
'色', '折', '淦', '硬', '芽', '喙', '恸', '响', '叶', '允', '烧', '岬', '赛', '忡', '头', '职', '刎', '仔',
```

内容 7：

```
# (7)
sorted(set(str))
print(f'词汇表大小:{len(set(str))}')
print(f'每个词平均使用次数:{len(str)/len(set(str))}')
print('\n')
```

```
词汇表大小:4014
每个词平均使用次数:174.40308918784254
```

内容 8：

```
# (8)
print('杨过一词出现的次数: ', end='')
print(str.count('杨过'))
fdist = FreqDist(str)
print('出现频率最高的: ')
print(fdist.most_common(30))
```

```
杨过一词出现的次数：6205
出现频率最高的：
[('不', 14979), ('一', 13150), ('道', 12076), ('过', 8379), ('人', 8286), ('杨', 6743), ('大', 6525),
 ('中', 5803), ('上', 5462), ('说', 5098), ('手', 5036), ('来', 5014), ('心', 4987), ('之', 4948), ('下
', 4791), ('小', 4765), ('见', 4612), ('子', 4571), ('出', 4360), ('女', 4161), ('师', 3927), ('去',
 3854), ('郭', 3790), ('身', 3620), ('声', 3501), ('无', 3160), ('只', 3109), ('得', 3021), ('时',
 3005), ('便', 2970)]
```

内容 9：

```python
# (9)
print(sorted(set(str)))
```

```
['一', '丁', '七', '万', '丈', '三', '上', '下', '不', '与', '丐', '丑', '专', '且', '世', '丘', '丙', '业',
 '丛', '东', '丝', '丞', '丢', '两', '严', '丧', '个', '丫', '丬', '中', '丰', '串', '临', '丸', '丹', '为
', '主', '丽', '举', '乃', '久', '么', '义', '之', '乌', '乍', '乎', '乏', '乐', '乒', '乓', '乔', '乖', '
乘', '乙', '九', '乞', '也', '习', '乡', '书', '买', '乱', '乳', '乾', '了', '予', '争', '事', '二', '于',
 '亏', '云', '互', '五', '井', '亚', '些', '亟', '亡', '亢', '交', '亥', '亦', '产', '享', '亩', '亭', '京',
 '亭', '亮', '亲', '亵', '人', '什', '仁', '仃', '仄', '仅', '仆', '仇', '今', '介', '仍', '从', '仑', '仓
', '仔', '他', '仗', '付', '仙', '仞', '代', '令', '以', '仪', '们', '仰', '仲', '件', '价', '任', '份', '
仿', '企', '伊', '伍', '伎', '伏', '伐', '休', '众', '优', '伙', '会', '伞', '伟', '传', '伤', '伥', '伦',
 '伧', '伪', '伫', '伯', '估', '伴', '伶', '伸', '伺', '似', '伽', '佃', '但', '位', '低', '住', '佐', '佑',
 '体', '何', '佗', '余', '佛', '作', '佚', '你', '佩', '佬', '佯', '佳', '佻', '使', '侃', '侄', '例
', '侍', '侏', '供', '依', '侠', '侣', '侥', '侦', '侧', '侨', '侩', '侮', '侯', '侵', '便', '促', '俟', '
俏', '俐', '俗', '俘', '俚', '保', '俣', '信', '俦', '俨', '俩', '俪', '俯', '俱', '俸', '倍', '候', '倒',
 '倔', '倘', '候', '倚', '倜', '借', '倡', '倦', '倨', '倩', '倪', '债', '值', '倾', '假', '偈', '偌', '偎',
 '偏', '偕', '做', '停', '健', '偬', '偶', '偷', '偿', '傀', '傅', '傍', '傥', '储', '催', '傲', '傻', '像
', '僧', '僮', '僵', '僻', '儒', '儡', '儿', '兀', '允', '元', '兄', '充', '兆', '先', '光', '克', '免', '
```

内容 10：

```python
# (10)
plt.rcParams['font.sans-serif'] = 'SimHei'
words = ['小龙女', '杨过', '郭靖', '黄蓉']
nltk.draw.dispersion.dispersion_plot(text, words, title='词汇离散图')
```

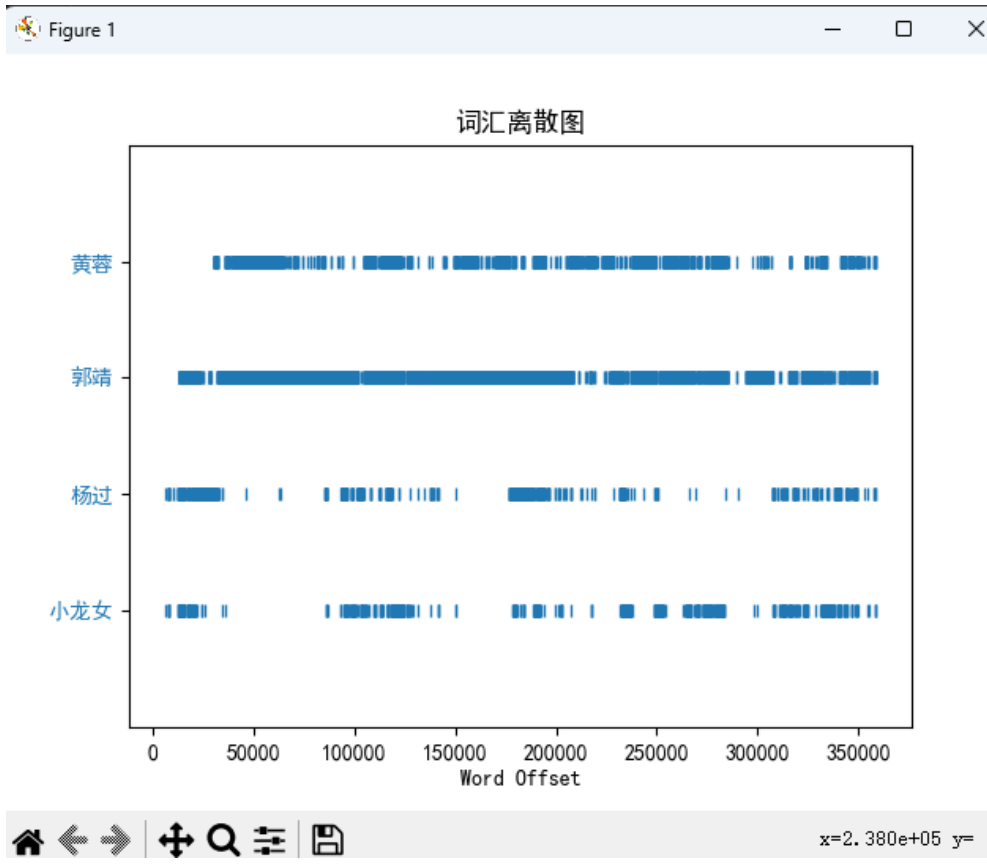
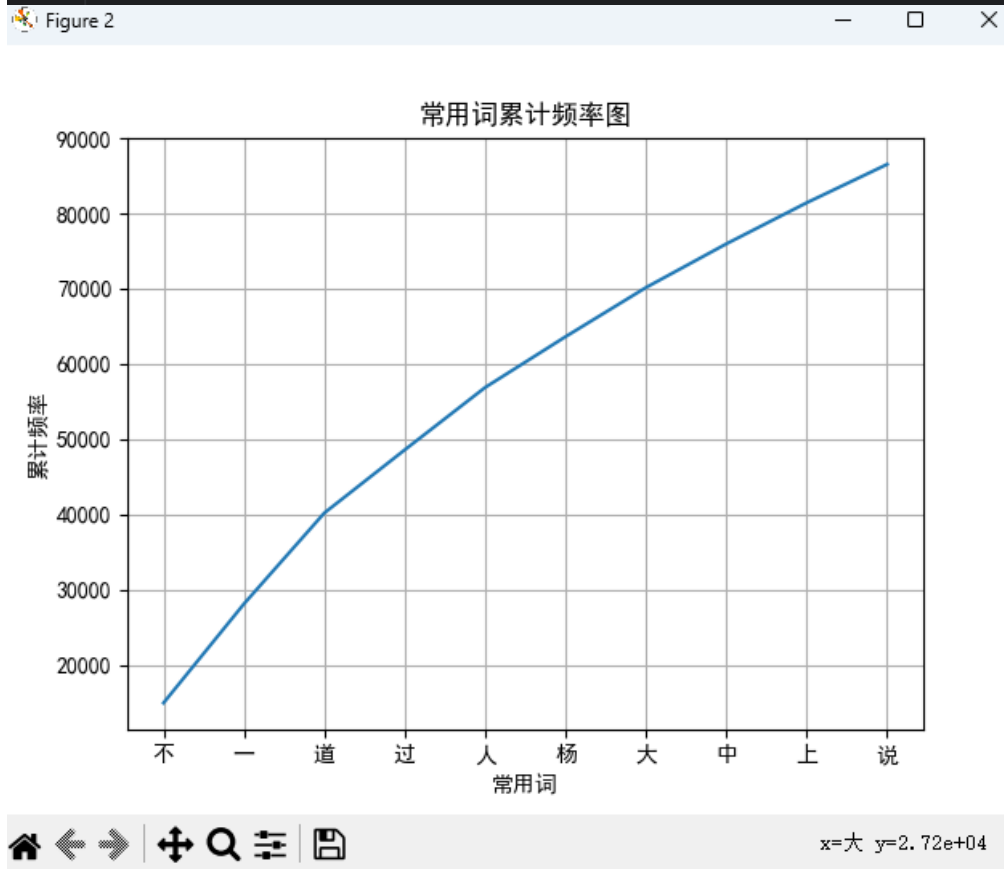


Figure 1
词汇离散图
黄蓉
郭靖
杨过
小龙女
0
50000
100000
150000
200000
250000
300000
350000
Word Offset
x=2.380e+05 y=

内容 11：

```python
# (10)
plt.rcParams['font.sans-serif'] = 'SimHei'
words = ['小龙女', '杨过', '郭靖', '黄蓉']
nltk.draw.dispersion.dispersion_plot(text, words, title='词汇离散图')
# (11)
fig = plt.figure()
plt.grid()
fdist1 = dict(fdist)
fdist1 = sorted(fdist.items(), key= lambda x: x[1], reverse=True)
x = []
y = []
for i in range(10):
    x.append(fdist1[i][0])
    y.append(fdist1[i][1])
t = 0
for i in range(len(y)):
    y[i] = y[i] + t
    t = y[i]
plt.plot( *args: x, y)
plt.title('常用词累计频率图')
plt.ylabel('累计频率')
plt.xlabel('常用词')
plt.show()
```



2、金庸武侠小说集语料库分析

（1）选其中一本小说进行分析。

预处理代码（去除停用词、去除非正文段落、增添特殊停用词）：

```
import jieba
import nltk
import re
from nltk.book import *
# 加载停用词列表
stop_words = []
path = 'E:/NLP/stopword.txt'
for line in open(path, encoding='utf8'):
    # print(line)
    line = line.strip()
    stop_words.append(line)
# 增添特殊停用词
added_stopword = ['具体来说','再其次','分期分批','切莫','到目前为止','猛然间','$']
stop_words.extend(added_stopword)
# 去除非正文段落
with open(file=r'D:\QQ\1413679561\FileRecv\data\金庸-天龙八部.txt', mode='r', encoding="gbk") as f:
    content_list = f.readlines()
    content_list = content_list[52:]
    content_list = content_list[:-40]
with open(file='金庸-天龙八部1.txt',mode='w') as f:
    content_str = ''.join(content_list)
    f.write(content_str)
# 去除停用词
result = []
chinese_result = []
with open(file='金庸-天龙八部1.txt', mode='r', encoding="gbk") as f:
    content = f.read()
    cleaned_data = ''.join(re.findall( pattern: '[\u4e00-\u9fa5]', content))
    content_list = jieba.lcut(content, cut_all=False)
    chinese_content_list = jieba.lcut(cleaned_data, cut_all=False)
    for word in content_list:
        if word not in stop_words:
            result.append(word)
    for word in chinese_content_list:
        if word not in stop_words:
            chinese_result.append(word)
str = ''.join(result)
chinese_str = ''.join(chinese_result)

```

（2）统计小说中总用词量和平均每个词的使用次数

```
# (2)
print(f'总用词量：{len(set(str))}')
print(f'平均每个词的使用次数{len(str)/len(set(str))}')
```

```
总用词量：4192
平均每个词的使用次数211.27862595419847
```

（3）自行指定 3 个词语，查看小说中的使用次数

```
# (3)
print(f'乔峰一次的使用次数：{str.count("乔峰")}')
print(f'段誉一次的使用次数：{str.count("段誉")}')
print(f'虚竹一次的使用次数：{str.count("虚竹")}')
```

```
乔峰一次的使用次数：1243
段誉一次的使用次数：3534
虚竹一次的使用次数：1671
```

（4）统计并输出前 30 个高频字符和对应出现次数

```python
# (4)
fdist =FreqDist(str)
print(fdist.most_common(30))
```

```
[('\u3000', 22150), ('不', 19690), ('道', 15684), ('一', 14460), ('人', 12617), ('\n', 11222), ('大', 9519), ('说', 7124), ('中', 6936), ('子', 6612), ('上', 6529), ('来', 6528), ('下', 6263), ('之', 6240), ('段', 5733), ('手', 5465), ('去', 5151), ('心', 5021), ('出', 5019), ('便', 5016), ('…', 4968), ('见', 4615), ('声', 4409), ('是', 4366), ('身', 4228), ('到', 3753), ('只', 3714), ('得', 3680), ('誉', 3658), ('好', 3578)]
```

（5）使用 jieba 对小说正文前十句话进行分词

```python
# (5)
# print(content_str.split('。')[0:10])
tensentences = content_str.split('。')[0:10]
for sentence in tensentences:
    splited_sentence = jieba.lcut(sentence)
    print(splited_sentence)
```

```
['一', ' ', '青衫', '磊落', '险峰', '行', '\n', '\n', '\u3000', '\u3000', '青光闪', '动', '，', '一柄', '青钢剑', '候地', '刺', '出', '，', '指向', '中年', '汉子', '左肩', '，', '使剑', '少年', '不待', '剑', '招用', '老', '，', '腕抖', '剑', '斜', '，', '剑锋', '已削', '向', '那', '汉子', '右颈']
['那', '中年', '汉子', '竖剑', '挡格', '，', '铮', '的', '一', '声响', '，', '双剑相', '击', '，', '嗡嗡', '做声', '，', '震声', '未绝', '，', '双刃剑', '光', '霍霍', '，', '已', '拆', '了', '三招']
['中年', '汉子', '长剑', '猛地', '击落', '，', '直', '斩', '少年', '顶门']
['那', '少年', '避', '向', '右侧', '，', '左手', '剑诀', '斜引', '，', '青钢剑', '疾刺', '那', '汉子', '大腿']
['\n', '\u3000', '\u3000', '两人', '剑法', '迅捷', '，', '全力', '相搏']
['\n', '\u3000', '\u3000', '练武', '厅', '东边', '坐', '着', '二人']
['上首', '是', '个', '四十左右', '的', '中年', '道姑', '，', '铁青', '着', '脸', '，', '嘴唇', '紧闭']
['下首', '是', '个', '五十余岁', '的', '老者', '，', '右手', '捻', '着', '长须', '，', '神情', '甚', '是', '得意']
['两人', '的', '座位', '相距', '一丈', '有余', '，', '身后', '各站', '着', '二十余名', '男女', '弟子']
['西边', '一排', '椅子', '上', '坐', '着', '十余位', '宾客']
```

（6）统计并输出前 30 个高频词和对应出现次数

```python
# (6)
fdist = FreqDist(chinese_str)
print(fdist.most_common(30))
```

```
[('不', 19692), ('道', 15684), ('一', 14412), ('人', 12617), ('大', 9519), ('说', 7124), ('中', 6936), ('来', 6656), ('子', 6612), ('之', 6566), ('上', 6529), ('下', 6263), ('段', 5733), ('手', 5465), ('去', 5151), ('心', 5021), ('出', 5019), ('便', 5016), ('见', 4612), ('声', 4409), ('是', 4303), ('身', 4228), ('到', 3742), ('只', 3714), ('誉', 3658), ('得', 3605), ('好', 3577), ('无', 3546), ('头', 3499), ('老', 3457)]
```

（7）与同你选择同样一本小说的同学，对比前述统计结果，说明结果相

同或者不同的原因。

不同的原因：

1.增添的特殊停用词不同

2.处理数据的时候选择的是否是只含有中文词的文本

3.jieba 分词的时候选用的模式不同

五、指导教师评语

成绩：　　批阅人：　　日期：