云南大学软件学院

实 验 报 告

课程： 大数据分析及应用实验 任课教师： 何婧

专业： 人工智能 学号： 20211060245 姓名： 陈俊宏 成绩：

专业： 人工智能 学号： 20211120171 姓名： 孔令高 成绩：

# 实验四 中文文本处理

## 一、 实验目的

1、掌握词性标注、去停用词等文本预处理方法

2、掌握中文文本分词的基本原理和方法。

3、掌握中文文本特征提取的基本原理和方法。

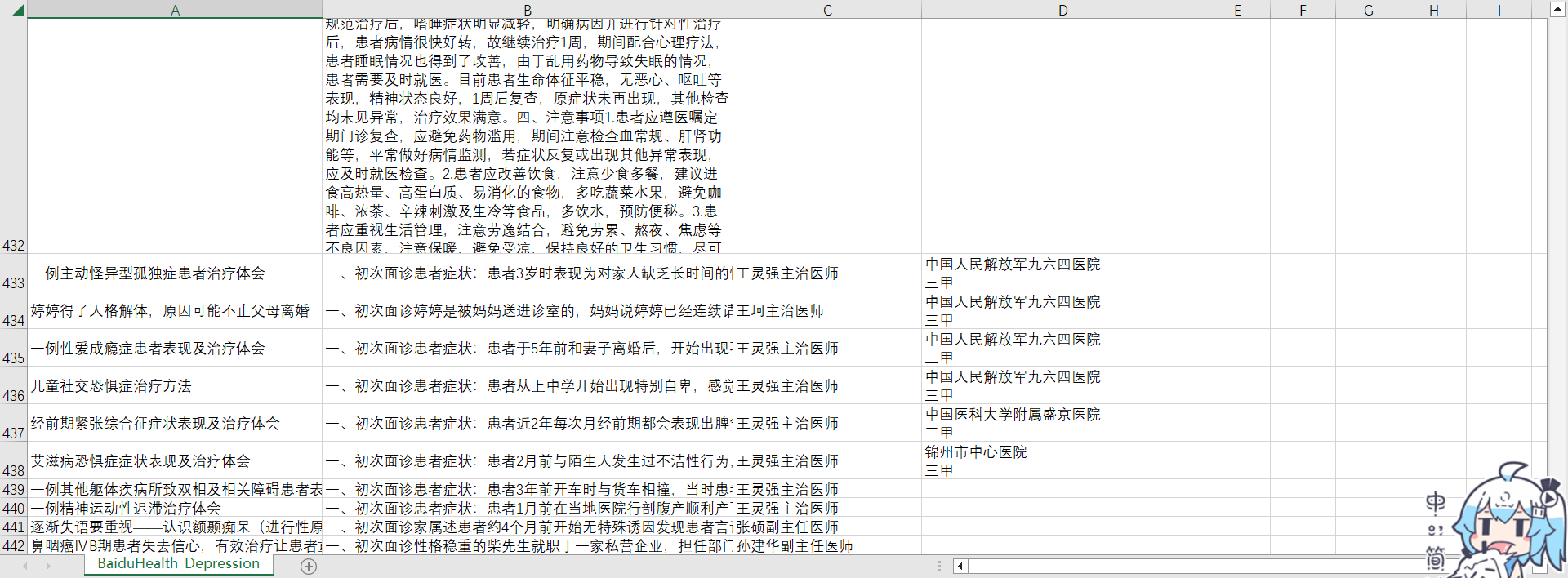
## 二、 实验内容及步骤

**1、基于正则表达式进行中文文本清洗**

（对已爬取的数据进行清洗，可以自己设计清洗样例）

1. 原始数据样例





分析数据可以看出，我们爬取的数据中，回答存在不是一句回答，而是病历本的形式，这对之后的数据分析流程是不友好的，所以需要进行清洗，且有部分数据的医院信息为空，为了维护数据的一致性，这些数据也需要进行清洗。

1. 实验流程和代码

*import* pandas *as* pd  
  
*# 清洗病历类型的数据*df = pd.read\_csv("BaiduHealth\_Depression.csv", encoding="utf-8")  
mask = df['回答'].str.match('^一、初次面诊')  
df = df[~mask]  
  
*# 清洗存在空值的数据*print("移除空值前")  
print(df[df.isna().any(axis=1)])  
df.dropna(axis=0, how='any', inplace=*True*)  
print("移除空值后")  
print(df[df.isna().any(axis=1)])  
df.to\_csv("Depression.csv", encoding="utf-8-sig", index=*False*)

通过正则表达式'^一、初次面诊'用以匹配病历本形式的数据，并调用df.dropna方法来清洗医院信息为空值的数据。

1. 实验结果

成功清洗原始数据，清洗前共441条数据，清洗后共363条数据。

**2、基于****jieba分词进行中文文本分词**

（选取自己爬取的文本进行实验，也可以使用较长的新闻文本进行实验）

1. 原始数据样例



对已爬取数据中的回答一列进行分词

1. jieba分词的基本原理和具体实现方法

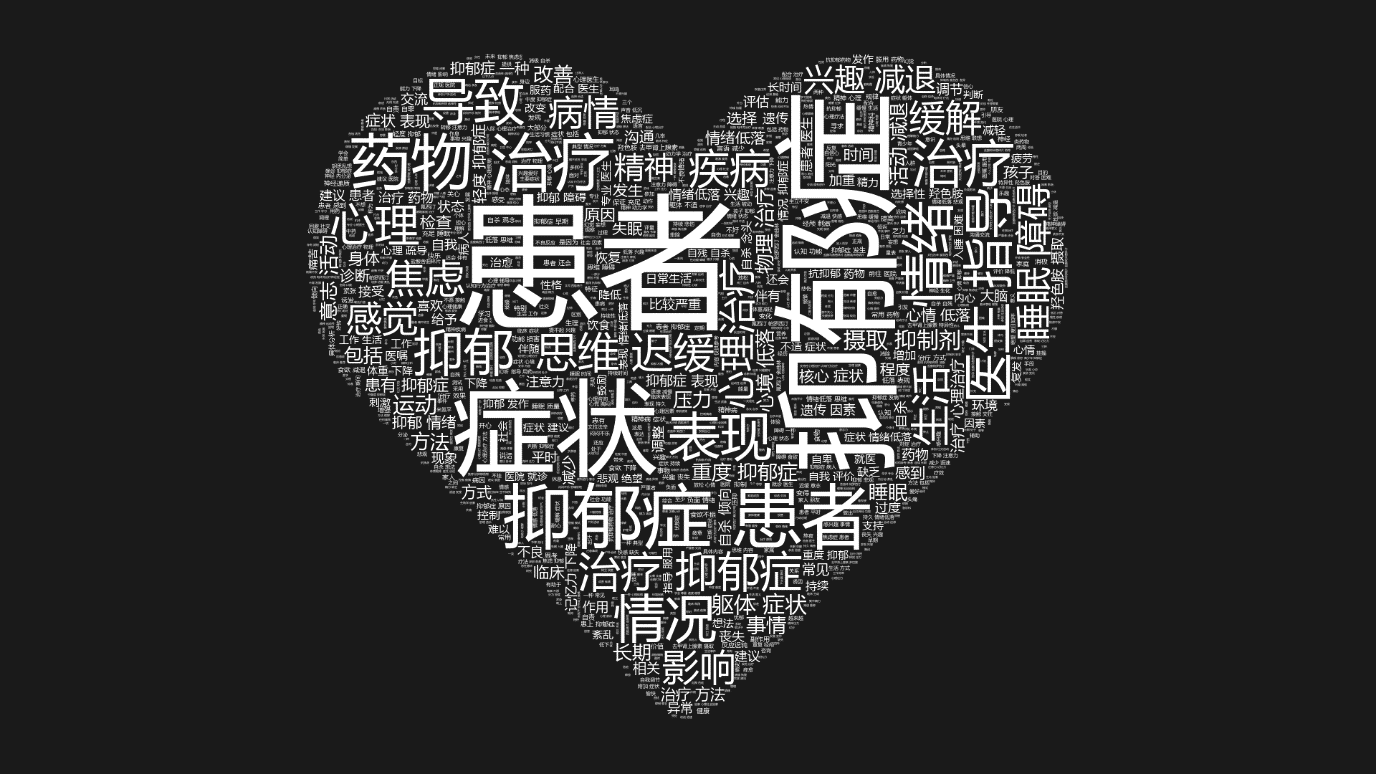
jieba分词的基本原理：通过字典匹配和统计信息，将中文文本切分成有意义的词语。它使用了一个包含常见词汇和短语的词典，采用全模式和精确模式来生成可能的词语组合。同时，jieba还利用隐马尔可夫模型处理未知词汇，以及使用TF-IDF算法进行关键词提取。用户可以通过自定义词典来调整分词效果。

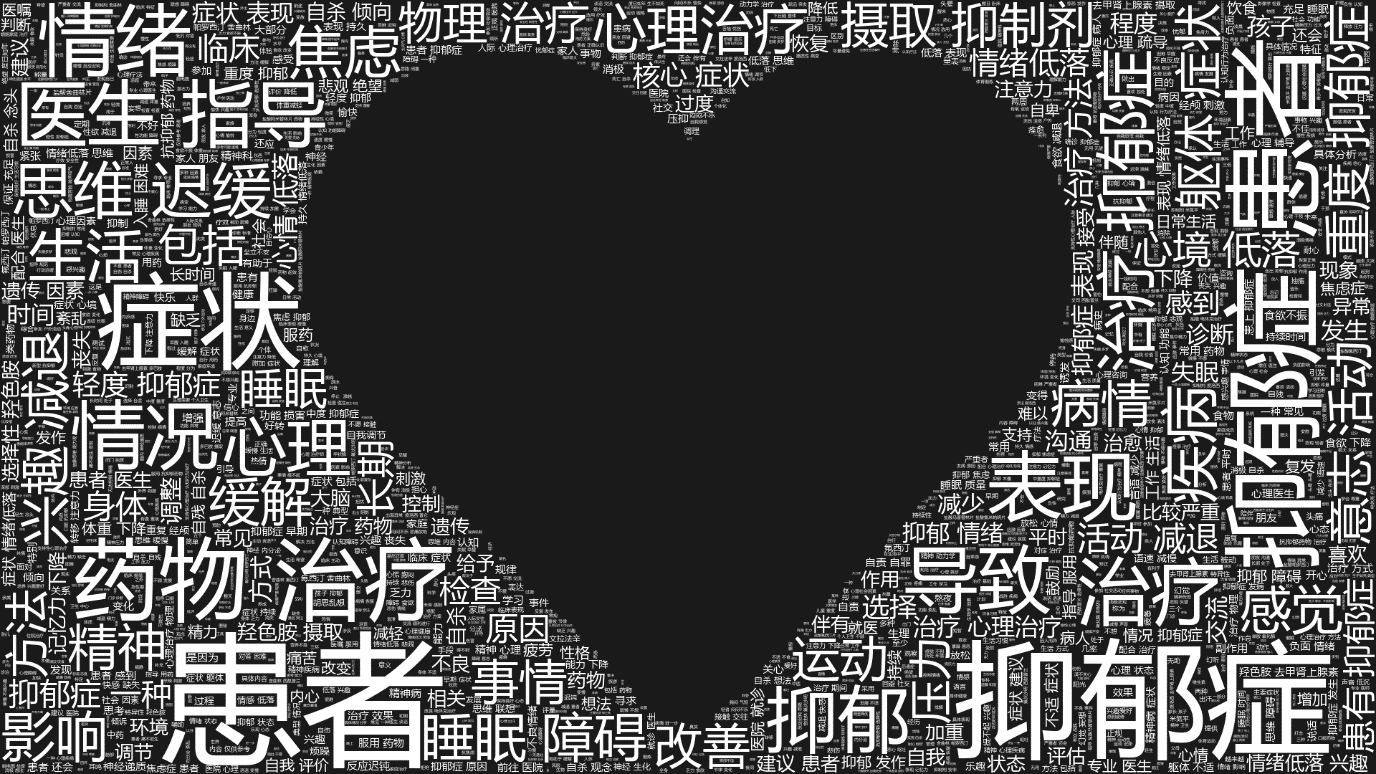
具体实现方法：利用python调取jieba模块来对回答内容进行分词，先读取爬取数据中的“回答”一列，加载停用词，加载药品名称词库以提高准确性，设置jieba分词模式为paddle模式(基于GRU循环神经网络)进行分词，以下是具体代码

*import* pandas *as* pd  
*import* jieba  
*import* paddle  
*import* stylecloud  
  
*# 启用paddle模式*paddle.enable\_static()  
jieba.enable\_paddle()  
  
*# 读取数据*df = pd.read\_csv('./Depression.csv', encoding='utf-8')  
answer = ''.join([i *for* i *in* df['回答']])  
  
*# 加载停用词*stopwords1 = [line.rstrip() *for* line *in* open(r'stopwords\baidu\_stopwords.txt', 'r',  
 encoding='utf-8')]  
stopwords2 = [line.rstrip() *for* line *in* open(r'stopwords\cn\_stopwords.txt', 'r',  
 encoding='utf-8')]  
stopwords3 = [line.rstrip() *for* line *in* open(r'stopwords\hit\_stopwords.txt', 'r',  
 encoding='utf-8')]  
stopwords4 = [line.rstrip() *for* line *in* open(r'stopwords\scu\_stopwords.txt', 'r',  
 encoding='utf-8')]  
stopwords5 = [line.rstrip() *for* line *in* open(r'stopwords\stopwords.txt', 'r',  
 encoding='utf-8')]  
  
stopwords = stopwords1 + stopwords2 + stopwords3 + stopwords4 + stopwords5  
meaningful\_words = []  
  
*# # 加载药品名称词库,数据来源于**https://github.com/xtea/chinese\_medical\_words*jieba.load\_userdict("medicine.txt")  
  
*# 分词*seg = list(jieba.cut(answer, use\_paddle=*True*))  
  
*# 删除停用词  
for* i *in* seg:  
 *if* (i *not in* stopwords) *and* (len(i) != 1):  
 meaningful\_words.append(i)  
  
result = " ".join(meaningful\_words)  
  
stylecloud.gen\_stylecloud(text=result,  
 font\_path='C:/Windows/Fonts/msyh.ttc',  
 output\_name='Depression.png',  
 icon\_name='fas fa-heart',  
 colors='white',  
 background\_color='#1A1A1A',  
 size=(1920, 1080),  
 invert\_mask=*False* )

1. 实验结果

通过词云的方式来展示实验结果：





**3、基于****TF-IDF的中文文本特征提取**

（对实验中得到的分词进行清洗）

1. 原始数据样例
2. TF-IDF的基本原理和具体实现方法
3. 实验结果分析

**4、中文分词方法对比分析（选做）**

（除jieba分词以外，任选一种分词方法，完成实验，并分析两种分词方法的时间性能和准确性）

1. 自选分词方法算法原理和算法流程概述
2. 分词的实现代码
3. 时间性能对比分析
4. 分词准确性对比分析

**5、文本特征提取方法探究**

（除TF-IDF以外，任选一种文本特征提取方法，完成实验）

1. 自选文本特征提取算法的原理和算法流程概述
2. 实现代码
3. 实验结果分析

## 三、 实验总结

（对实验工具的特点，个人的学习情况等进行总结）

1. 由于在爬取的过程中比较注重数据质量，所以爬取到的数据质量较高。于是在数据清洗的过程中就比较简单顺利。
2. 而由于数据清洗比较成功，于是乎分词的任务也变得简单了起来，我们先直接对数据进行分词操作后，发现一些无意义的词和标点符号较多，于是引入了5个常用停用词词库作为停用词词典，而后发现常规的精确模式和全模式并不能很好的切分一些术语，于是采用了基于GRU循环神经网络的paddle模式，最后发现一些药物名称并不能被很好的切分开来，于是增添了<https://github.com/xtea/chinese_medical_words>中的药物数据集作为词典，最终获得了不错的分词效果。