

# README2

[1. 任务描述](#)

[2. 数据集](#)

[3. 最大匹配算法](#)

[4. 最少分词法（最短路径法）](#)

[5. 制作词云](#)

[注：](#)

[报告+代码提交注意事项](#)

## 1. 任务描述

利用课程中学过的最大匹配算法、最少分词法对中文语料进行分词。

您需要在给定的PKU词典的帮助下，编写两种最大匹配算法分别对数据集进行分词，并将分词结果与真实的分词结果文件进行对比，计算出精确率P、召回率R和F1值。

为了增强趣味性，选取你最爱的某本书的一个章节，使用你所实现的分词算法对该章节进行分词，去除停用词之后再进行词云的绘制。

## 2. 数据集

数据集来自人民日报语料库，包括三个文件，分别是 词典、待分词文件 和 分词对比文件。

## 3. 最大匹配算法

流程：

65-79页

1. 实现双向最大匹配算法
2. 将分词结果与真实结果进行对比，并计算精确率P、召回率R和F1值。

## 4. 最少分词法（最短路径法）

流程：

1. 实现最少分词算法
2. 将分词结果与真实结果进行对比，并计算P、R、F1

## 5. 制作词云

关于词云的内容请自行了解

流程：

1. 选取你最爱的某本书的一个章节，使用你所实现的分词算法对该章节进行分词
2. 将分词之后的词，去除停用词（停用词已放入 停用词 文件夹里）
3. 去除停用词之后就可以绘制词云了，下图是爬取了某一集《名侦探柯南》的弹幕所制作的词云，供参考。

绘制词云的参考代码如下：

```
# -*-coding:utf-8-*-
import jieba
from wordcloud import WordCloud
import matplotlib.pyplot as plt

stopwords = [line.strip() for line in open('Library/stopwords.txt', 'r', encoding='utf-8')]
with open('柯南/547.txt', 'r', encoding='utf-8') as f:
    txt = f.read()
    # 将jieba分词换成你所实现的分词算法
    words = jieba.cut(txt)

    sentences = ""
    for word in words:
        if word in stopwords:
            continue
        sentences += str(word)+' '
    # 生成词云就这一步
wordcloud = WordCloud(background_color='white',
                       font_path="Library/SourceHanSerif-Heavy.ttc",
                       width=2000,
                       height=2000).generate(sentences)

# 输出词云图片，自行学习matplotlib.pyplot如何使用
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig("547集琴酒词云")
plt.show()
```



思考，在我们分词的场景下，准确率和召回率如何求

## 改进

对于日期期这种固定形式的词组，若并未出现在词典中，分词的效果并不会很好，借用规则匹配出日期等有固定形式的词组会提升分词的效果。

[规则匹配 93-100页](#)

## 报告+代码提交注意事项

- 材料提交截止时间：2022-10-15 23:59
- 材料上传地址：  
<https://bhpan.buaa.edu.cn:443/link/254B1F3FA5F657679CC9A588EF78687F>  
有效期限：2022-10-15 23:59
- 附件打包示例

```
|——19XXXXXX-张三.zip
|   |——19XXXXXX-张三-中文分词.docx/.pdf
|   |——词云
|   |——代码
|   |   |——XXX.py
|   |   |——XXX.py
```

