自然语言处理实验报告

实验二

学 院： 信息工程学院

指导教师： 孙 媛

班 级： 19计算机2班

学生姓名：王 君 、高 雯 慧、路 琪

学 号：19040119、19040126、19040127

日期：2020 年10月21日

目 录

一、实验内容

二、实验目的

三、实验原理

四、实验步骤

五、主要程序模块和框架

六、实验结果

七、总结

八、程序亮点

摘 要

通过近几年的发展，互联网已经离我们不再遥远。互联网上的信息也在急剧膨胀，在这海量的信息中，各类信息混杂在一起，要想充分利用这些信息资源就要对它们进行整理，如果由人来做这项工作，已经是不可能的，而如果面对中文信息不采用分词技术，那么整理的结果就过于粗糙，而导致资源的不可用。通过引入分词技术，就可以使机器对海量信息的整理更准确更合理，效率也会大幅度的提高。

并且智能计算就是让机器“能看会想，能听会讲”。要想实现这样的一个目标，首先就要让机器理解人类的语言，只有机器理解了人类的语言文字，才使得人与机器的交流成为可能。在人类的语言中，“词是最小的能够独立活动的有意义的语言成分”。从现阶段的实际情况来看，英文已经跨越了分词这一步，也就是说在词的利用上已经先一步，并且已经展现了良好的应用前景，无论是信息检索还是主题分析的研究都要强于中文，所以对于中文来讲，将词确定下来是理解自然语言的第一步，只有跨越了这一步，中文才能像英文那样过渡到短语划分、概念抽取以及主题分析，以至自然语言理解，最终达到智能计算的最高境界。所以中文分词对我们来说意义十分重大。

**实验二 采用正向、逆向最大匹配算法进行文本分词**

**一、实验内容**

1、对语料库的文本进行分词并存储。分别采用正向最大匹配算法、逆向最大匹配算法进行分词。

2、以jieba分词的分词结果作为标准语料，计算P、R、F值

**二、实验目的**

1、熟悉掌握反向最大匹配算法、正向最大匹配算法、双向最大匹配算法和掌握Mmseg算法

2、掌握分词方法

3、了解中文分词的作用和意义

**三、实验原理**

**正向最大匹配算法**

假定分词词典中的最长词有i个汉字字符，则用被处理文档的当前字串中的前i个字作为匹配字段，查找字典。若字典中存在这样一个i字词，则匹配成功，匹配字段被作为一个词切分出来。如果词典中找不到这样的一个词，则匹配失败，将匹配字段的最后一个字去掉，对剩下的字串重新进行匹配。如此进行下去，直到匹配成功，即切分出一个词或剩余字串的长度为零为止。这样就完成了一轮匹配，然后去下一个i字串进行匹配处理，直到文档被扫描完成。

**反向最大匹配算法**

根据事先最备好的词典，从后往前进行匹配，假如有匹配到的词直接分割，若没有的话减少最左边一个词然后继续匹配，直至所有的句子匹配完毕。

**双向最大匹配算法**

基于正向、反向最大匹配，FMM和BMM两种算法都分词一遍，然后根据大颗粒度词越多越好，非词典词和单字词越少越好的原则，选取其中一种分词结果输出。

**Mmseg算法**

Mmseg分词算法的主要思想是每次从一个需要分词的句子里，按照从左到右的顺序，识别多种不同的3个词组合。然后根据mmseg的4条消岐规则选取最优分词组合。 然后选择备选词组合中的第1个词，作为1次迭代的分词结果；剩余的词(即这个句子中除了第一个已经分出的词的剩余部分)继续进行下一轮的分词运算。为传统的前向最大匹配算法加入了上下文信息，解决了其每次选词只考虑词本身，而忽视上下文相关词的问题。

**中文分词**

中文分词是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂的多、困难的多。

**四、实验步骤**

**（一）文章分割**

1. 读取txt文档
2. 除去标点符号只留中文
3. 分割文章存入txt文档中

**（二）中文分词**

**正向最大匹配**

1、一般从一个字符串的开始位置，选择一个最大长度的词长的片段，如果序列不足最大词长，则选择全部序列。

2、首先看该片段是否在词典中，如果是，则算为一个分出来的，如果不是，则从右边开始，减少一个字符，然后看短一点的这个片段是否在词典中，依次循环，逐到只剩下一个字。

3、序列变为第2步骤截取分词后，剩下的部分序列

**反向最大匹配**

反向最大匹配法的基本原理与正向最大匹配法类似只是分词顺序变为从右至左。一般从一个字符串的开始位置，选择一个最大长度的词长的片段，如果序列不足最大词长，则选择全部序列。

**双向最大匹配**

双向最大匹配算法的原理就是将正向最大匹配算法和逆向最大匹配算法进行比较，从而选择正确的分词方式。

1.比较两种匹配算法的结果

2.如果分词数量结果不同：选择数量较少的那个

3.如果分词数量结果相同

​ （1）分词结果相同，返回任意一个

​ （2）分词结果不同，返回单字数较少的一个

​ （3）若单字数也相同，任意返回一个

1. **用jieba库分词**

调用settings，并依次打开测试文档中的txt文件运用replace将文件中的换行(\n)以及空格(“ “)。后运用jieba库进行分词，并将结果存入jieba.txt中。

**（四）计算PRF值**

1. 正确率 = 提取出的正确信息条数 / 提取出的信息条数

2. 召回率 = 提取出的正确信息条数 / 样本中的信息条数

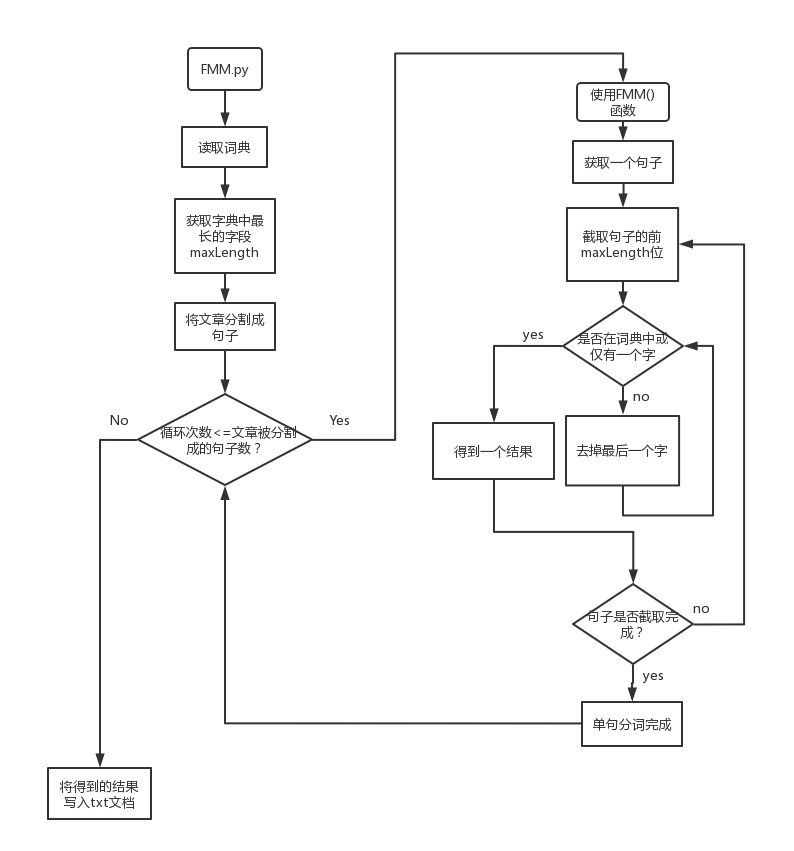
3. F值 = 正确率 \* 召回率 \* 2 / (正确率 + 召回率)

**五、主要程序模块和框架**

**FMM.py**

首先读取词典，然后在FMM\_result()函数中获取字典中最长的字段maxLength，调用settings()将文章按照标点分割成句子，循环调用获取一个句子分词的函数FMM()将整片文章的句子都完成中文分词以获取整个文章的分词，最后在save()函数中调用FMM\_result()函数得到的结果并写入txt文档内。

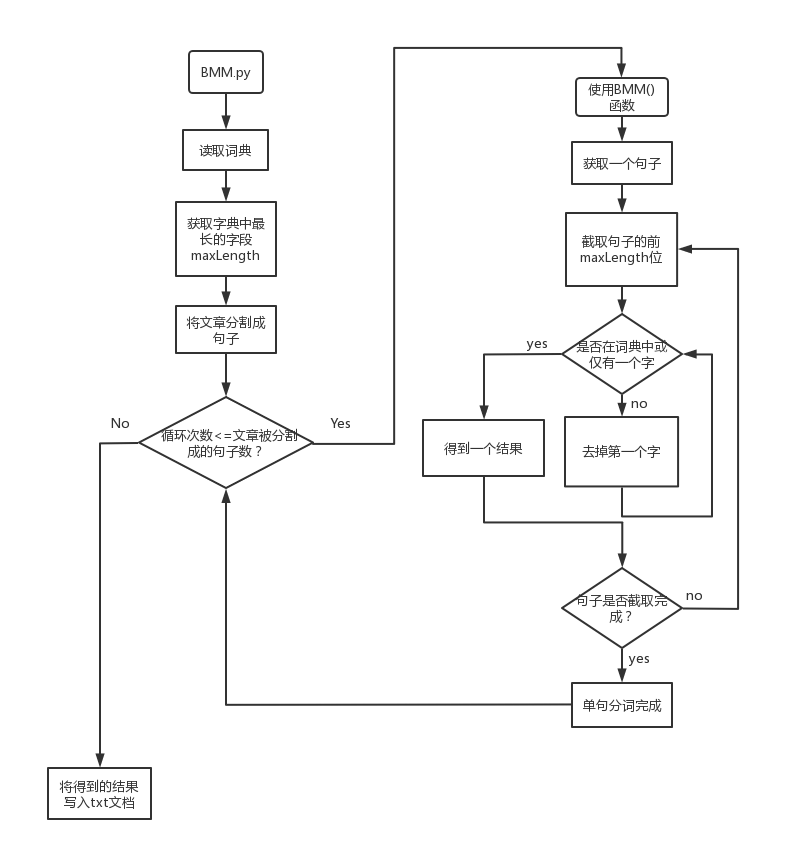
FMM()函数的作用是从左往右切分获取一个句子的分词。首先读取一个句子sentence，然后从左向右截取句子的maxLength（匹配长度）位，如果在词典中无法匹配，去掉最后一个字继续进行匹配，直到得到一个分词结果。用sentence减去得到的分词结果继续进行截取，直到句子sentence分词完成。



**BMM.py**

首先读取词典，然后在BMM\_result()函数中获取字典中最长的字段maxLength，调用settings()将文章按照标点分割成句子，循环调用获取一个句子分词的函数BMM()将整片文章的句子都完成中文分词以获取整个文章的分词，最后在save()函数中调用BMM\_result()函数得到的结果并写入txt文档内。

BMM()函数的作用是从右往左切分获取一个句子的分词。首先读取一个句子sentence，然后从右向左截取句子的maxLength（匹配长度）位，如果在词典中无法匹配，去掉最后一个字继续进行匹配，直到得到一个分词结果。用sentence减去得到的分词结果继续进行截取，直到句子sentence分词完成。

****

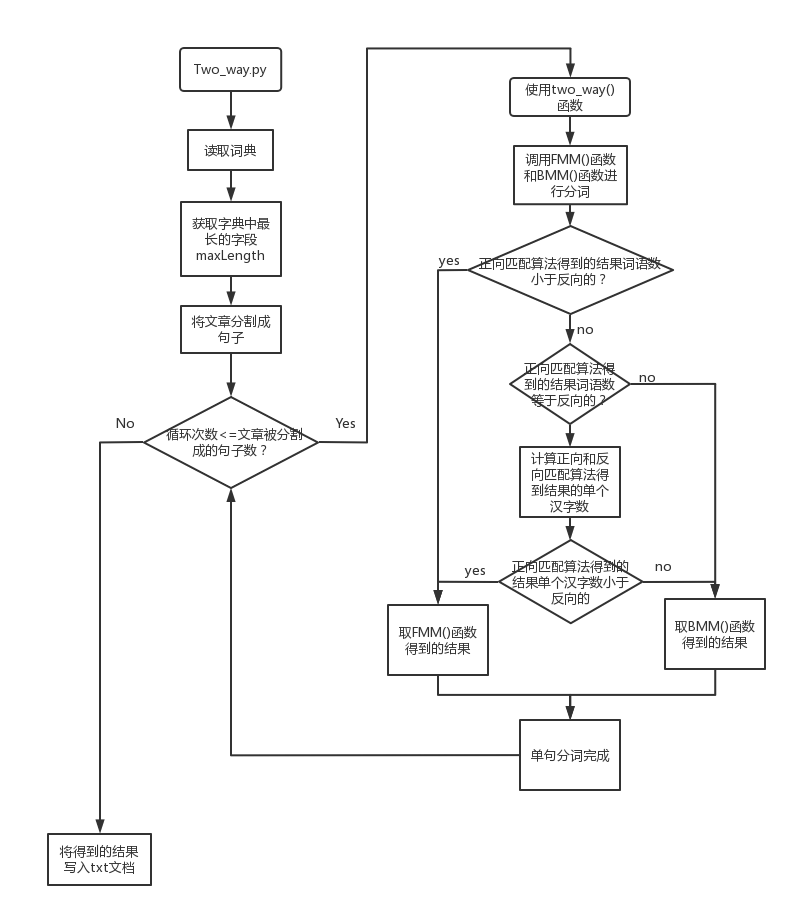
**Two\_way.py**

首先读取词典，然后在two\_way\_result()函数中获取字典中最长的字段maxLength，调用settings()将文章按照标点分割成句子，循环调用获取一个句子分词的函数two\_way()将整片文章的句子都完成中文分词以获取整个文章的分词，最后在save()函数中调用two\_way\_result()函数得到的结果并写入txt文档内。

count\_single\_char()函数用来计算中文分词后的句子所包含的单个汉字的个数。

two\_way()函数的作用是获取一个句子的分词。

分别调用FMM()和BMM()函数得到一个句子的两种分词结果，比较两个函数的到的结果：如果正向匹配算法得到的结果词语数小于反向的，取FMM()函数的结果；正向匹配算法得到的结果词语数大于反向的，那么取BMM()函数的结果；如果相等，调用count\_single\_char()函数计算正向和反向匹配算法的到结果的单个汉字数，如果正向匹配算法得到的结果单个汉字数小于反向的，取FMM()函数的结果，反之取BMM()函数的结果。



**PyMmseg.py**

1.调用settings中的 readFile()

sentence = readFile():将文章[犬夜叉]奈落夫人的穿越记事-扶华.1772167.txt的”\n”与” ”替换成”,”;

Dict, maxLength = readDicProocess():加载词典，求取字典中的最长字段;

readAllTxt():读取所有的训练文档，即train\_txt文件夹;

2.类1群：

(1)Word类

相当于C语言中的结构体，用来存单词和词频的

(2)Chunk类

Chunk用来实现具体的切割判断方法的前期预处理计算的，其中包含依据上下文分出的一组词和相关的属性的函数，包括totalWordLength()(总长度)、averageWordLength()(平均长度)、standardDeviation()(标准差)和wordFrequency()(自由语素度)

(3)ComplexCompare类

用来具体实现mmseg算法的四种评估方法的

3.

(1)加载词组字典和字符字典

1.loadDictChars函数：首先打开字符字典，然后逐行读取字典，获取最长字符长度；

2.loadDictWords函数：首先打开词组字典，然后逐行读取字典，获取最长词组长度；

(2)判断该词word是否在字典dictWord中

getDictWord函数：运用get()来确定该word是否在字典dictWord中

(3)开始加载字典

run函数：调用函数loadDictChars()和loadDictWords()来加载字典chars.dic和words.dic

4.类2：定义了一系列属性

Analysis类：具体实现切割算法的

1).判断该字符是否是中文字符（不包括中文标点）

isChineseChar函数：使用ord()函数获取ASCII给定字符的值，并判断是否在中文字符范围内

2).判断是否是ASCII码

isASCIIChar函数：利用string库：如果为string.whitespace空格和string.punctuation标点字符，返回False，最后返回string.printable能够被打印的ASCII字符串

3).判断是否为其他字符

getNextToken函数：调用isChineseChar()和isASCIIChar()，返回不在这两个函数中的字符

4).得到下一个切割结果

getNextToken函数：调用isChineseChar函数与getNextChar函数，判断self.text[self.pos]是否为中文字符，若是，另token=调用getChineseWords函数切割出的中文词；若不是，另token=调用getASCIIWords

函数切割出的非中文词。

5).切割出非中文词

getASCIIWords函数：调用了getNextChar函数，isASCIIChar函数，isChineseChar函数

6).切割出中文词，并且做处理

getChineseWords函数：首先判断chunks长度，如果>1:调用ComplexCompare类中的四个mmseg算法的四种过滤原则函数处理；如果==0，返回空。处理后，定义word = chunks[0].words，token =””,length =0,将word中的词依次写入token中，并统计其长度，最后返回token。

7).三重循环来枚举切割方法，这里也可以运用递归来实现

createChunks函数：调用了getMatchChineseWords函数，

8).用正向最大匹配算法结合字典来切割中文文本和其他文本

getMatchChineseWords：首先在循环中调用了getDictWord函数并将确定的词组放入列表中成为词典，循环结束后，若词典列表中没有词组，则放置个‘X’，将文本长度标记为-1。最后返回词典列表。

5.cuttest函数：

调用Analysis类，完成文本分割

6.Mmseg函数

调用settings中的 readFile()

sentence = readFile():将文章[犬夜叉]奈落夫人的穿越记事-扶华.1772167.txt的”\n”与” ”替换成”,”，然后利用split将文章按照标点符号分割成句子，然后依次调用cuttest函数分割句子并赋值至result，最后返回result。

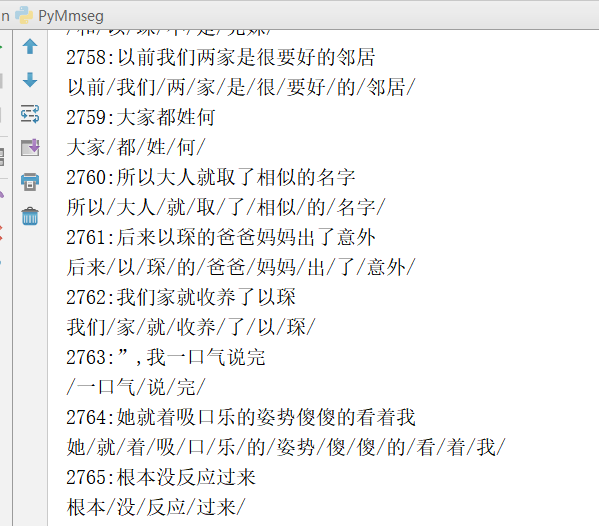
7.save函数

将所有训练文档中的文档依次进行分词处理，并赋值给result，然获将result写入MMSEG.txt中。

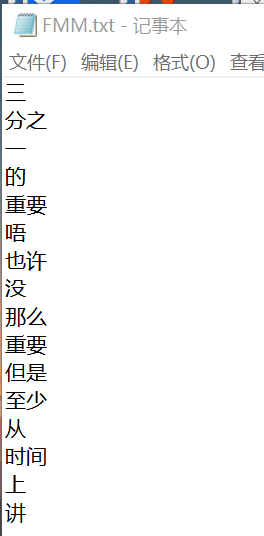
**六、实验结果**

结果保存在data文件夹里

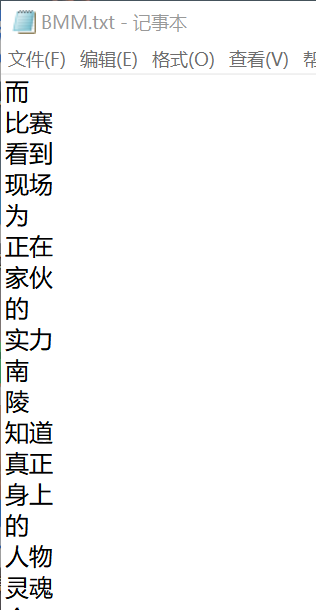
PyMmseg.py



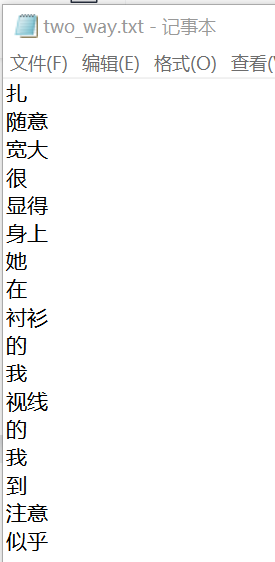
FMM.py 正向最大匹配



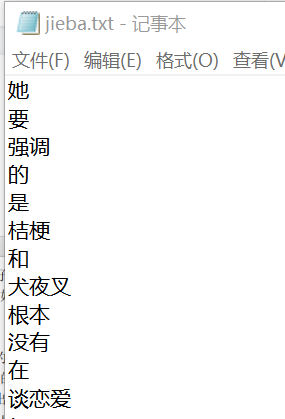
BMM.py 反向最大匹配



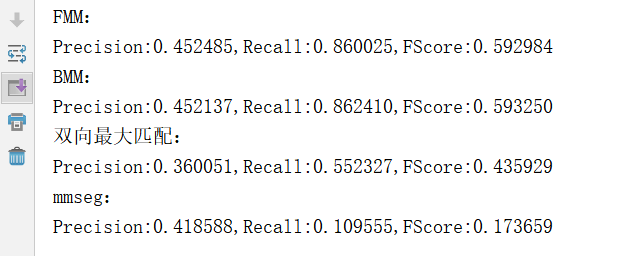
Two\_way.py 双向最大匹配



Jieba\_text.py



Caculat\_P\_R\_F.py ：计算PRF值



**七、总结**

通过此次实验，掌握了反向最大匹配算法、正向最大匹配算法、双向最大匹配算法和Mmseg算法等一系列算法，可以对语料库的文本进行分词并存储。同时，我们对我国中文分词发展现状和中文分词的作用和意义有了更加深刻的了解和认识。

**八、程序亮点**

1、编写了双向最大匹配算法。双向最大匹配算法在一定程度消除了歧义。研究表明，中文中90.0%左右的句子，正向最大匹配法和逆向最大匹配法完全重合且正确，只有大概9.0%的句子两种切分方法得到的结果不一样，但其中必有一个是正确的，只有不到1.0%的句子，使用正向最大匹配法和逆向最大匹配法的切分虽重合却是错的，或者正向最大匹配法和逆向最大匹配法切分不同但两个都不对。

2、编写了mmseg算法。Mmseg算法的运行速度较快，效果较好。有两个词典，一个字符词典一个词组词典分别判断，有四种消歧规则，使得得到的结果更加准确。在简单和复杂的匹配算法中都使用了最大匹配规则来消除分词的歧义。剩下的三个规则没有（而且不能）应用于简单匹配算法。四个消歧规则为：规则 1：最大匹配；规则 2：最大平均单词长度；规则 3：单词长度的最小方差；规则 4：单字单词的语素自由度的最大和。

3、在settings.py进行文章分割的时候，使用了字典，大大提高了将文章分割成句子的速度。字典几秒钟，列表几小时。

4、如果遇到词典中没有的词语，会将其加进词典，完善充实词典。