Chinese word segmentation说明文档

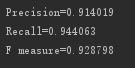
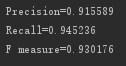
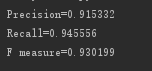
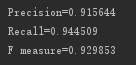
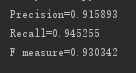
1. 任务定义

设计一个中文分词器，对所给的中文语料数据进行分词。通过所给的PKU语料数据，其中已做好分词与词性标注，对该语料进行训练，得到一个中文分词器，并对原始语料进行分词，与正确的分词结果进行比较，用Precision、Recall和F Measure方法进行评估。

1. 输入输出

输入为nlpdata.txt（所给的语料集）。

输出Precision、Recall、F measure。

（随机选取的训练集与测试集，多次测量体现平均性能）

同时输出一个test\_segment.txt，这是测试集经过该中文分词器进行分词得到的结果。

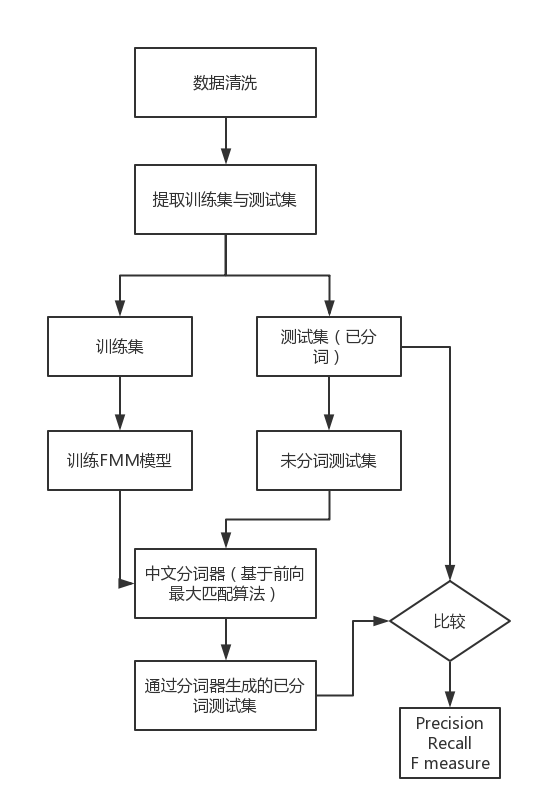
另一个输出test\_origin.txt，这是测试集原本的正确分词，即已分好词的语料集中的部分。

1. 源码运行环境

使用python 3.6进行编写，在Windows 10上运行通过。源代码为ChineseSegment.py。

1. 方法描述

基本算法描述如下图：



* 数据处理

观察所给的语料集，其中：

每一条语料前面都有一个日期；（日期为0-9 与 '-' 组成）

每一条语料内还有换行；（换行为'\r\n'）

每一个分好的词后都有一个词性标注；（由'\'与英文字母组成）

有的词用[]括起来，其中右括号后面跟着一个nt；

有的词还会有拼音标注，{}括号内为拼音。（括号内由英文字母与数字组成）

使用正则表达式寻找所有符合上述特征的数据，其中的特点在括号内已标出，将之剔除，完成数据的清洗。

* 训练集与测试集

在处理数据时可以得到语料的数量，我们选择其中的20%作为测试集，80%作为训练集。生成一个随机数序列，从语料集中取该序列对应的语料，保证测试集的随机性。

刚开始使用sklearn模块中的随机划分训练集方法，发现并不难，可以手动实现，于是将其注释掉。

取到的测试集是一个已分好词的语料集，将其输出到test\_segment.txt，作为已分好词的参考。我们对其进行还原，即把分好的词还原为还未分的状态，用作中文分词器的输入。

* 前向最大匹配分词与正确分词判断

FMM定义：通过切分出单词串，与词库进行对比，若该单词串在词库中已存在，则划分该词为一个词，否则，词串中的字数减一，再进行比较。若词串中只剩下单个字，则进行单独划分。

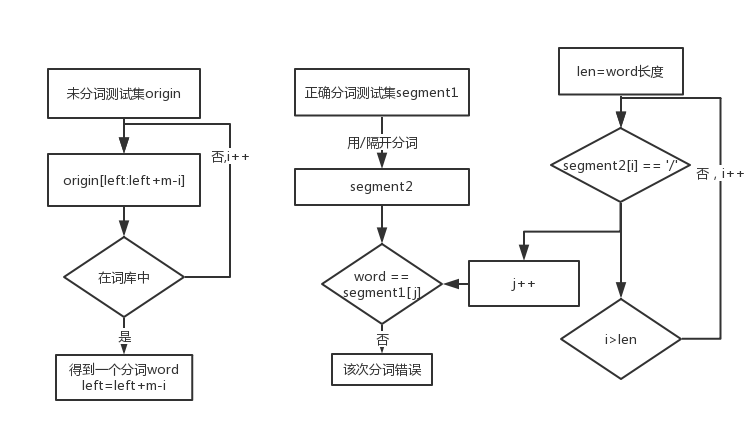
词库：对训练集中的数据进行统计，整合到一个词典（dict）里，若该词（key）在训练集中存在，则对应的值（value）为1。该词典就是上述的词库。（一开始直接在训练集中进行查找，但数量太过庞大，导致查找速度很慢，于是进行统计再查找）

因为测试集数据太过庞大（大概30万个字），不可能从最后一位开始进行前向匹配，这会导致时间复杂度为阶乘级别。于是计算词库中已划分的词的最大字数m，将m作为FMM算法词串的最大字数，即最大匹配词长为m个字，若词库中不存在该词，则词串中的字数减1，一直到词串中只剩一个字（若还未匹配到）。

定义FMM匹配词长为一个滑动窗口，即窗口的右边界会不停左移，若匹配到词库中有该词，则左边界等于右边界，添加分界符，进行下一次匹配。

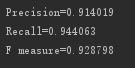
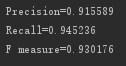
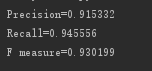
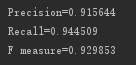
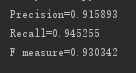
最后进行分词正确的判断。设未分词测试集origin经过一次分词器得到的一个分词为word，需要判断word与正确分词测试集segment1里相对应的分词是否正确匹配。此时应该解决的问题是，如何判断word对应segment1里的哪一个分词。要找到segment1里对应word的词，于是设置一个指针j指向segment1里的一个分词。将segment1的所有分词，用'/'连接起来，得到一个超长字符串segment2，设置一个指针i指向segment2里的一个字。

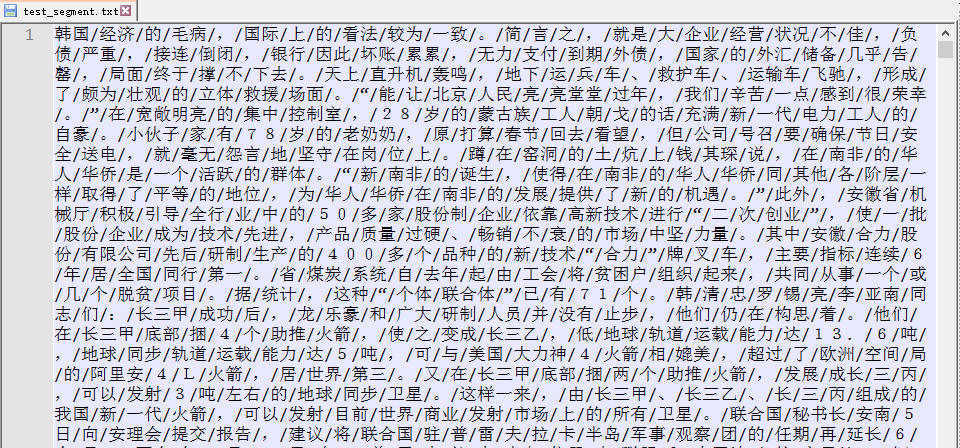
现在根据一次分词得到的word的长度（字的个数），来控制i的移动，每次分词后i将会移动word长度个单位。若i指向分词符'/'，则指针j +1，就指向segment1里的下一个分词。我们只需判断word是否等于segment1[j]，就能知道该次分词是否正确。

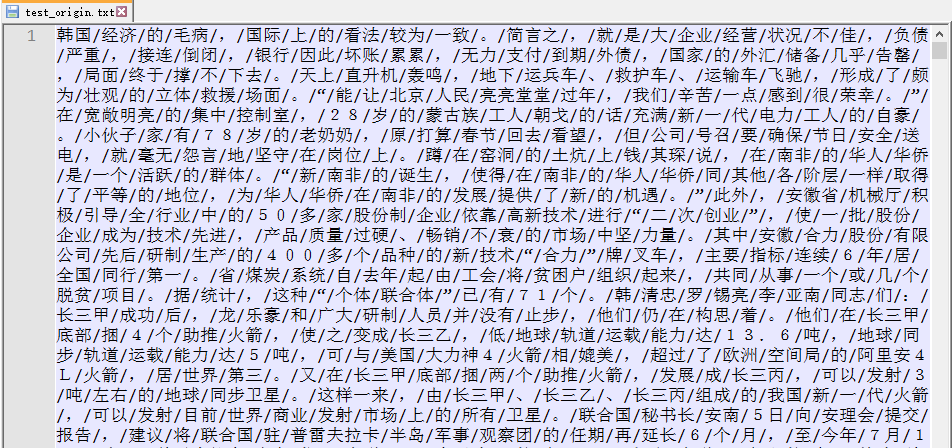
该部分算法流程如下：

1. 结果分析（性能评价）

由于训练集与测试集都是在所给的语料库中随机提取，故每次测量的Precision、Recall和F measure都会不一样，在此测试了5次作平均评价。



 test\_segment是经过分词器所分的词，test\_origin是原语料库中的分词，将两者进行对比，可以发现基本一致。



可以看到Precision稳定在91.5%，Recall=94.5%，F measure=93%。对于来自同一个语料库的训练集与测试集，这个模型正确分词的概率还是相当高的。但是由于只有一个语料库，测试集中会出现一些词从未出现在训练集中，称为数据稀疏，此时就很难进行分词，或者可以说是分词失败。

考虑这样一种情况，本文所用来进行训练的语料集是中国日报的内容，如果我们选择一些游戏评测来进行测试，那么结果肯定会有所偏差。因为偏向方面不同，游戏评测中的许多专有名词都无法匹配到训练词库中，导致分词结果将会很僵硬。这也是前向最大匹配算法的一个缺点，即若训练词库中并不存在，则会一个一个地进行分词。