北京邮电大学　计算机学院

《自然语言处理》实验报告

姓名　　王睿嘉

学号 2015211906

班级 2015211307

Chinese Word Segmentation

一、 实验内容和环境描述

1. 实验内容

Chinese word segmentation: 20 points

-This task provides PKU data as training set and test set (e.g., you can use 80% data for model training and other 20% for testing), and you are free to use data learned or model trained from any resources.

Evaluation Metrics:
Precision = (Number of words correctly segmented)/(Number of words segmented) * 100%
Recall = (Number of words correctly segmented)/(Number of words in the reference) * 100%
F measure = 2*P*R / (P+R)

2. 环境描述

编程语言：python
集成开发环境：PyCharm
解释器版本：2.7.13

二、 输入输出定义

1. 输入定义

用户无需自行输入数据。

本实验以压缩包所含文件“Chinese text corpus.txt”为语料库，通过一定划分产生训练样本和测试文本。

2. 输出定义

控制台输出分词的准确率、召回率和 F 值。

三、 实验设计

1. 整体说明

本实验使用教师所提供的“Chinese text corpus.txt”语料库，前 80%的语料用作训练集，剩余部分作为测试文本。

完成了分词的任务，并有较高准确率、召回率和 F 值，效果较为理想。

实验流程如下：

a. 对原始语料进行预处理，删去无效信息。同时，将前 80%输出，用作训练集；剩余部分的原样和取消分词形式均输出，分别用作分词答案和测试文本；

b. 读取训练集，并生成词典；

c. 对测试文本进行分词，并将结果与答案进行比对，计算准确率、召回率和 F 值。

2. 关键点说明

2.1 分词算法

本实验采取简单有效的前向最大长度匹配算法。在训练过程中，记录最大词条长度。测试时，顺序扫描测试文本，从最大词条长度开始，按递减方式截取子串，并在已有词典中查找其是否存在。若存在，即匹配成功，更新相关记录。

2.2 未出现词

在分词过程中，需要对未出现词进行处理。本实验采取的方式是：从最大词条长度开始，按递减方式截取子串，若所有子串均未出现在词典中，说明存在未出现词。随之将匹配起点后移一位，重复匹配过程，若子串出现在词典中，两坐标间即为一个未出现词条，可进行分词。

2.3 性能评估

本实验采用准确率（precision）、召回率（recall）和 F 值来分别评估分词结果。

公式如下：

准确率＝正确分词个数/分词总个数*100%

召回率＝正确分词个数/答案中分词总个数*100%

F=2*准确率*召回率/(准确率+召回率)

四、 文件说明

共有两个源码文件：

Pre-Processing.py：对语料库进行预处理，删去无效信息。同时，将前 80%输出，用作训练集；剩余部分的原样和取消分词形式均输出，分别用作分词答案和测试文本。

Chinese Word Segmentation.py：对测试文本进行分词，并对其结果进行评估。

共有 5 个文本文件：

Chinese text corpus.txt：教师所提供的语料库。

train.txt：训练集。

test.txt：测试文本。

answer.txt：分词答案。

result.txt：分词结果。

五、 实验结果

如下表：

训练集比例	Precision	Recall	F
80%	90.04%	94.15%	92.05%

六、实验心得

在本次实验中，遇到的主要问题有以下两点：

a) 未出现词的处理。若通过不断增加子串长度来寻求词条匹配，存在未出现词时，会产生词条无限增长的情况，因而需要一个上限。在生成词典过程中，记录最大词条长度无疑是个较好的选择。分词时，顺序扫描测试文本，从最大词条长度开始，按递减方式截取子串，若存在未出现词，将匹配起点后移一位，重复匹配过程，该问题得到解决；

b) 换行符问题。首次运行时，准确率极低，反复检查代码和算法，并无明显漏洞。print 分词结果和答案的比对过程，发现分词结果每隔一行便会出现空行，导致结果与答案完全对不上，查看 txt 文件却无此现象，推断应是换行符出现了问题，却不知如何解决，这个问题困扰了我许久。最终发现 readlines()读入了换行符，分词时，换行符被当作最后一个未出现词记录了下标，而我在每行结束又通过代码 write()换行符至结果文件，因而存在两个换行符。至于文本文件显示无异常的原因是：在我的计算机上"\r\n"才显示换行，最终，该问题得到解决。

实践出真知，本次实验是对课堂所学知识的补充。从初拿到题目的无从下手，到查阅各种资料逐步分析出算法，加深了对分词算法的理解，收获颇丰。