华东师范大学数据学院上机实践报告

 课程名称:信息检索
 年级:2018
 上机实践成绩:

 指导教师:张蓉
 姓名:孙秋实

上机实践名称: 汉语分词: 最大匹配方法 学号: 10185501402 上机实践日期: 2021/9/16

Part 1

实验目的

(1) 编写程序,完成基于最大匹配方法的汉语分词

(2) 评价分词结果

Part 2

实验任务

- (1) 完成基于最大匹配方法的汉语分词程序编写。
- (2) 使用 Precision/Recall/F1-Score 来评价分词结果
- (3) 案例分析与可视化

Part 3

使用环境

- (1) Google Colab
- (2) Python 3.7

Part 4

实验过程

Section 1

最大匹配方法简介

中文分词的目的是将汉字序列切分为词序列,而最大匹配算法则是一种基于词典的贪心算法,我们拥有一个词典,其中有所有可能的词语,即除了单字,分词结果中的每个词均要在词典中出现。从句首开始,向右截取最大长度,组成当前词和字典中的词逐一进行匹配。若匹配成功,则移向下一位置,若失败,则缩短1个字的长度,重复该匹配动作,直到匹配成功或是只剩单字。

Section 2

中文分词:最大匹配算法的实现

实现最大分词算法,其思想比较简单,已在简介中阐述,算法的核心函数如下 (其他如读写文件等次要代码在报告中已省略,详情请见 Jupyter 记事本)

```
def MaximumMatching(corpus):
 time_start=time.time()
 match answer = []
 sentence_num = 0
 for line in corpus:
  per_sentence_answer = []
  pos = 0
  while pos<len(line): # 终止条件为pos到了当前这个句子的最大长度
     current_line_len = len(line)
     start_len = min(max_word_len, current_line_len-pos-1) # start_len指初始状态的最长匹配长度
     # 此处需要确定最大单词长,可能是10,也可能是当前匹配后剩下的部分
     for current_len in range(start_len,0,-1): # 从最大长度开始减,每次失败就-1
         if (line[pos:pos+current_len] in dict_array):
           per_sentence_answer.append(line[pos:pos+current_len]) #
               往单句匹配结果加入当前匹配好的词
           pos=pos+current_len-1
         # position移动到匹配完单词后的下一个位置
         elif current len == 1: # 只有单字可匹配则直接加入匹配结果中
           per_sentence_answer.append(line[pos:pos+current_len])
  # 将当前句子的分词结果以列表形式append到答案中
  match_answer.append(per_sentence_answer)
  sentence_num+=1
 time_end=time.time()
 print("Total sentence number for word seg:",sentence_num)
 print('Total time cost',time_end-time_start,'s')
 return match_answer
```



1 match_result = MaximumMatching(corpus)

Total sentence number for word seg: 803
Total time cost 10.873929500579834 s

图 1: 算法执行

如图 1 所示总共为 803 个句子进行分词, 时间开销 10.87 秒。

分词结果存储在 corpus.out.txt, 如图 2 所示

```
corpus.out.txt X
 1 戴相龙 说 中国 经济 发展 为 亚洲 作出 积极 贡献
 2 新华社 福冈 5月 11日 电 (记者 乐绍延)
 3 中国人 民 银行 行长 戴相龙 今天 在 亚洲 开发 银行 第30 届 年会 的 "亚洲 未来 30 年 "研讨会 上 说 ,
                                                                                          中国 的 经济 发展
4 戴相龙 在 发言 时 说 , 中国 的 发展 得益于 亚洲 国家 和 地区 的 经济 发展 与 合作 , 与 亚洲 的 繁荣 息息相关 。
5 他 指出 , 随着 经济 的 持续 增长 和 改革 开放 政策 的 深入 , 中国 将 在 亚洲 经济 区域 合作 中 发挥 更 积极 的 作用
 6 中国 经济 的 快速 增长 将 为 亚洲 地区 创造 更 多 的 贸易 机会 , 在 今后 四 年 中 , 中国 将 为 世界 提供 将近 7000亿7 关于 香港 回归 中国 后 的 国际 金融 地位 问题 , 戴相龙 强调 , 香港 的 国际 金融 地位 不但 能够 维持 , 而且 还 会 得到 カ
                                                                          中国 将 为 世界 提供 将近 7000亿
8 在 谈到 亚洲 经济 的 发展 前景 时, 戴相龙 认为 , 亚洲 经济 将 继续 保持 稳定 的 发展 蒴头 , 9 戴相龙 同时 指出 , 亚洲 经济 发展中 还 存在 工资 上涨 过快 削弱 竞争力; 高级 研究 、 管理 人
                                                                                  仍 将 成为 推动 世界 经济
                                                                           管理 人才 严重 匮乏 ; 能源 、
10 戴相龙 认为 , 要 保持 亚洲 地区 经济 增长 , 既 需要 亚洲 各 国 继续 开发 利用 自身 的 经济 潜力 , 也 需要 进一步 加强 \mathbb P 11 亚洲 国家 和 地区 今后 除了 在 商品 、 投资 领域 加强 合作 外 , 还 应 在 科技 和 环保 以及 货币 政策 和 金融 监管 方面 加 12 亚洲 开发 银行 总裁 佐藤光夫 主持 了 这 次 研讨会 。
13 日本 前 首相 宫泽喜一 、 印度 财政部 长 奇丹巴拉姆 和 芬兰 环境 部长 佩卡·哈维斯托 也 在 研讨会 上 发 了 言 。
15 美国 罗素 二O -- 二O 协会 考察 中国 中西部 地区
16 新华社 重庆 五月 十二日 电 ( 记者 李佩 )
                         每 年 都 要 选择 一 个 新兴 市场 作为 投资 机会 的 美国 罗素 二〇 -- 二〇 协会 , 如今 把 目
17 拥有 万亿 美元 可投 资金 、
             这 个 协会 的 成员 对 刚刚 升为 直辖市 的 重庆市 进行 了 三 天 的 参观 考察
18 从 十日 起 .
19 重庆 位于中国的西南地区,是中国西部最大的工业城市,也是中国西部利用外国投资最多的城市。
20 今年三月十四日重庆成为中国第四个、也是西部唯一的一个直辖市。
21 在 重庆 期间 , 协会 成员 听取 了 市政府 关于 重庆市 发展 战略 、 对外 开放 优惠 政策 及 投资 环境 和 合作 机会 的 介绍 ,
22 此前 , 这 个 协会 的 成员 还 考察 了 中部 工业 名城 武汉 。
23 罗素 二〇 一一 二〇 协会 是 由 世界 最大 的 养老金 管理 公司 弗兰克·罗素 公司 董事长 乔治·罗素 于 一九九〇年 六月 发起 成立
24 据 介绍 , 该 协会 在 中国 现有 投资 达 二十亿 美元 , 此 次 访 华 的 主要 目的 是 考察 中国 中西部 地区 的 投资 环境
25 (完)
26 青岛 优化 资本 结构 促进 企业 规模 扩大
27 新华社 青岛 五月 十九日 电 ( 记者 刘海民 )
```

图 2: corpus.out.txt

Section 3

评价分词结果

将获得的分词结果保存为 corpus.out.txt 后,与 corpus.answer.txt 进行匹配,衡量该分词任务的效果。该部分的核心函数如下

```
def check_performance(match_result,correct_match):
 OutputFile = open("performance.txt","w",encoding="utf8")
 OutputFile.write("Index\t Precision\t Recall\t F1-score\n")
 # 写Header
 compare_len = len(correct_match)
 check_index = 0
 for i in range(compare len):
   check_index += 1 # 句子序号
  temp_len = len(match_result[i]) # 当前校验的句子的分词结果的长度
  valid_tokens = correct_match[i].split() # 正确的分词结果
   valid_len = len(valid_tokens) # 正确的分词结果的长度
   correct_sum = 0 # 维护一个变量记录多少个正确
  answer_pos = valid_pos = 0 # 位置指针
  seg_len = val_len = 0 # 两个变量来表示长度位置
  # 开始和答案配对
   current_corpus_len = len(corpus[i]) - 1 # -1是因为下标从0开始
   while(seg_len<current_corpus_len and val_len<current_corpus_len):</pre>
   # 被校验的分词结果和正确结果任意一个达到了最大长度则终止
  # 以下匹配答案,需要注意长度不一,分为三种情况
      if seg_len == val_len:
       if (match_result[i][answer_pos] == valid_tokens[valid_pos]): correct_sum += 1
```

华东师范大学数据科学与工程学院学生实验报告

```
seg_len += len(match_result[i][answer_pos]) # 第i个句的当前位置
val_len += len(valid_tokens[valid_pos])
answer_pos += 1
valid_pos += 1
elif seg_len > val_len:
val_len += len(valid_tokens[valid_pos]) # 当前tokens前移一位
valid_pos += 1
else: # 小
seg_len += len(match_result[i][answer_pos]) # 当前的position前移一位
answer_pos += 1
# 计算三个指标 Precision Recall 和 F1-Score
Precision = correct_sum/temp_len
Recall = correct_sum/valid_len
F1_score = cal_F1(Precision,Recall) #函数如上
OutputFile.write("%d\t %.2f\t %.2f\t %.2f\n" % (check_index, Precision, Recall, F1_score))
```

该部分输出的示例如表 1 所示

表 1: 分词结果评价(前十句)

Index	Precision	Recall	F1-score
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	0.95	0.95	0.95
4	1.0	1.0	1.0
5	1.0	1.0	1.0
6	1.0	1.0	1.0
7	1.0	1.0	1.0
8	1.0	1.0	1.0
9	0.98	0.96	0.97
10	1.0	1.0	1.0

最后将分词结果的 Precision/Recall/F1-Score 单独输出到文件中,将输出结果转为 csv 格式后,借助 Pandas 可以查看总体的分词效果。

```
print("average Precision",np.mean(df[' Precision']))
print("average Recall",np.mean(df[' Recall']))
print("average F1-Score",np.mean(df[' F1-score']))

# average Precision 0.9947198007471995
# average Recall 0.9923412204234126
# average F1-Score 0.99348692403487
```

总体评价:

- Precision:0.9947
- Recall:0.9923

• F1-Score:0.9934

Section 4

案例分析

可视化 803 条句子的 Precision/Recall/F1-Score,可以看到存在一个极端值,如图 3 所示

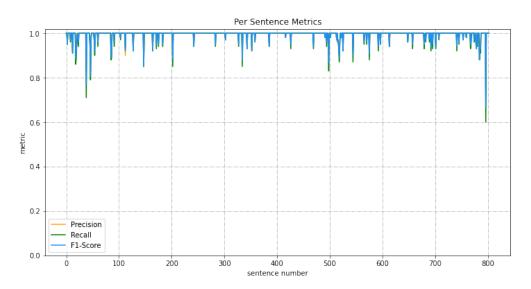


图 3: Word Segmentation Evaluation

分析得知在这个数据集内,表现最差的为第 797 句,单独对其进行分析。第 797 行文本为"【水蒸蛋糕】",最大匹配分词结果为【水蒸蛋糕】,答案为【水 蒸蛋糕】,水蒸这个词在词典里,所以被匹配了,整个句子比较短,所以导致了准确率和召回率都出现了明显的下降,但并不影响该算法在整体文本上的有效性。