**中文分词之最大匹配法**

**—— 文本语义挖掘**

|  |  |
| --- | --- |
| **学号：** | 1130310128 |
| **姓名：** | 杨尚斌 |
| **专业：** | 计算机科学与技术 |
| **指导老师：** | 赵铁军 |

1. **方法介绍**

分词是最小的能够独立活动的有意义的语言成分，英文单词之间是以空格作为自然分界符的，对中文的处理是基于自动切分的单字切分，或者二元切分。除此之外，还有最大切分（包括向前、向后、以及前后相结合）、最少切分、全切分等等。

词是最小的能够独立活动的有意义的语言成分，汉语中是以字为基本书写单位的，词与词之间没有明显的区分标记，因此在中文处理中，词的处理就是文本语义挖掘中很重要的一部分。

对中文的分词目前来说比较成型的分词思路有：基于词典，基于统计，基于规则。

本来主要使用的是基于词典的**最大匹配法**。这个方法在中文分词中比较简单，并且效果相对来说比较好，但是由于汉语语言的丰富，以及词典完备性的限制等还是使得这种方法难以在大规模的文本分词处理中使用。

该方法的主要原理如下：

最大匹配法分为正向和逆向，原理基本一致，这里就以正向最大匹配法为例说明该方法的原理。

首先我们可以规定一个词的最大长度，每次扫描的时候寻找当前开始的这个长度的词来和字典中的词匹配，如果没有找到，就缩短长度继续寻找，直到找到或者成为单字。

**实例：**

S1="计算语言学课程是三个课时" ,设定最大词长MaxLen = 5  ,S2= " "

字典中含有三个词：[计算语言学]、[课程]、[课时]

（1）S2=""；S1不为空，从S1左边取出候选子串W="计算语言学"；

（2）查词表，“计算语言学”在词表中，将W加入到S2中，S2=“计算语言学/ ”，并将W从S1中去掉，此时S1="课程是三个课时"；

（3）S1不为空，于是从S1左边取出候选子串W="课程是三个"；

（4）查词表，W不在词表中，将W最右边一个字去掉，得到W="课程是三"；

（5）查词表，W不在词表中，将W最右边一个字去掉，得到W="课程是"；

（6）查词表，W不在词表中，将W最右边一个字去掉，得到W="课程"

（7）查词表，W在词表中，将W加入到S2中，S2=“计算语言学/ 课程/ ”，并将W从S1中去掉，此时S1="是三个课时"；

（8）S1不为空，于是从S1左边取出候选子串W="是三个课时"；

（9）查词表，W不在词表中，将W最右边一个字去掉，得到W="是三个课"；

（10）查词表，W不在词表中，将W最右边一个字去掉，得到W="是三个"；

（11）查词表，W不在词表中，将W最右边一个字去掉，得到W="是三"

（12）查词表，W不在词表中，将W最右边一个字去掉，得到W=“是”，这时 W是单字，将W加入到S2中，S2=“计算语言学/ 课程/ 是/ ”，并将 W从S1中去掉，此时S1="三个课时"；

（13）S1不为空，从S1左边取出候选子串W="三个课时"；

（14）查词表，W不在词表中，将W最右边一个字去掉，得到W="三个课"；

（15）查词表，W不在词表中，将W最右边一个字去掉，得到W="三个"；

（16）查词表，W不在词表中，将W最右边一个字去掉，得到W=“三”，这时 W是单字，将W加入到S2中，S2=“计算语言学/ 课程/ 是/ 三/ ”，并 将W从S1中去掉，此时S1="个课时"；

（17）S1不为空，从S1左边取出候选子串W="个课时"；

（18）查词表，W不在词表中，将W最右边一个字去掉，得到W="个课"；

（19）查词表，W不在词表中，将W最右边一个字去掉，得到W=“个”， 这时W是单字，将W加入到S2中，S2=“计算语言学/ 课程/ 是/ 三/ 个/ "，并将W从S1中去掉，此时S1="课时"；

（20）S1不为空，从S1左边取出候选子串W="课时"；

（21）查词表，W在词表中，将W加入到S2中，S2=“计算语言学/ 课程/ 是/ 三/ 个/ 课时/ "，并将W从S1中去掉，此时S1=""。

（22）S1为空，输出S2作为分词结果，分词过程结束。

1. **实验设备**

* **软硬件设备**

**硬件设备**

Windows 10 专业版

Intel(R) Core(TM) i5-3230M CPU @2.60GHz

**软件设备**

Python

* **资源介绍**

**词典：**

注明 jieba 分词提供的通用词库

**测试文件：**

人民日报中的文章

1. **词典介绍**

* **总词数**

348982

* **最长词字数**

16

* **最长词字数个数**

16 词个数: 6

15 词个数: 9

14 词个数: 15

13 词个数: 26

12 词个数: 30

11 词个数: 77

10 词个数: 150

9 词个数: 225

8 词个数: 918

7 词个数: 1641

6 词个数: 3563

5 词个数: 84895

4 词个数: 131323

3 词个数: 114139

2 词个数: 11580

**词典下载链接：**https://github.com/yanyiwu/nodejieba/blob/master/dict/jieba.dict.utf8

1. **实验结果**

**测试语料：**

把便利“递”到群众手上（治国理政新思想新实践·新理念带来新变化）

(人民日报 2016-5-30)

**正确的个数**

466

**错误的个数**

200

**PRF:**

正确率：466/466+200 = 0.6996

召回率：666/711 = 0.9367

F: 0.6996 \* 0.9367 \* 2 / (0.6996 + 0.9367) = 0.8009

1. **错误分析**

由于实验采用的是最大匹配法，这种方法最后的结果中词典是很重要的一个因素，在语料的分析中，如果语料中的词没有登录词典，则无法对这种情况进行识别，由于这次使用的词典是通用词典，所以分析的效果上还是有些不是特别理想。除此之外，最大匹配法这种方法不能在语义上有相关的分析，所以对一些歧义不是能很好的进行消除。

整体来说，如果想要达到更好的分词效果，应该增大词典的容量与专业性，这样才能达到尽可能好的效果。除此之外，如果想要消除一些歧义，应该采用其他的方法相结合来处理这个问题。

1. **附录**

**实验语料：**

“把改革方案的含金量充分展示出来，让人民群众有更多获得感”

　　对于22岁的四川姑娘伍佳玲来说，玩自拍不是什么新鲜事儿，但让她感到新奇的是，在她的工作地重庆，如今办身份证用的照片也可以“自拍”“自选”。这不，5月27日下午，在重庆两江新区人和派出所的户籍大厅，伍佳玲就体验了一回。

　　在“自助拍照系统”前坐下，调整好位置和表情，伍佳玲按下了遥控器上的“拍照”键，面前的显示屏上立刻“跳”出了照片预览。连拍三张后，她选择了最满意的一张。“在重庆能办四川的身份证，已经是惊喜了，没想到还能玩自拍，真是高大上。”伍佳玲笑着说。

　　伍佳玲在重庆经营一家网店，几天前不小心把身份证丢了，挂失之后想着赶快补办。“一想到回老家去办，我头都大了。”她说，从重庆回老家，单程就5个多小时，车费加上食宿没有几百元下不来。“关键是现在生意忙，要走个两三天还真脱不开身。”伍佳玲说，“后来听说在重庆也能办，我就来了。真是太方便了！”

　　作为全国居民身份证异地受理首批试点省市之一，重庆市于2015年12月借助自主开发的信息系统，正式启动了与四川省的跨省异地受理工作，同时全面推行市内居民身份证异地受理工作。“今年5月16日我们按照统一要求，启用了全国居民身份证异地受理工作平台，将天津、湖南也纳入了跨省异地受理范围。”市公安局治安管理总队人口支队副支队长王劲松介绍，截至目前，全市共受理异地证10万余人次，其中跨省异地证3200多张。

　　翻开人和派出所今年以来的异地受理登记表，涉及了全市38个区县中的34个。“大部分是务工人员，有的边远区县回去一趟单程就要五六个小时。”副所长伍庆勇说，“能让他们少跑路，少请假，他们满意，我们也高兴。”

　　临近下班，派出所户籍大厅迎来了位“川妹子”刘倩，是西南大学的学生，接到电话后来取办好的身份证。“我还怕来晚了你们会下班呢。”刘倩说。“不会，你电话里说要来，我们就会等。”窗口民警冯璐璐告诉她，“我们还公布了预约电话，如果大家平时上班来不了，周末也可以预约。”

　　“小证件”事关“大和谐”。为做好这项工作，重庆市公安局从人员配备、业务培训、设备保障、作风建设等各方面统筹部署，扎实推进。“6月6日之前，我们还将把浙江、安徽、江西、湖北四省纳入跨省异地受理范围，全市所有的区县分局都要设立跨省异地受理点。”市公安局副局长郭金严说，要注重运用科技手段不断提高工作效率，确保改革举措落地生根，把更多的获得感“递”到群众手上。

　　惠民之举赢得了山城群众点赞。“改革不在大小，利民就好；红利不在厚薄，真实为贵。小小身份证，也是大文章。”网友“江边打鱼人”评价道。

**正确分词的原文：**

["“", "把", "改革方案", "的", "含金量", "充分", "展示", "出来", "，", "让", "人民", "群众", "有", "更", "多", "获得", "感", "”", "　", "

", "　", "　", "对于", "22", "岁", "的", "四川", "姑娘", "伍佳玲", "来说", "，", "玩", "自", "拍", "不是", "什么", "新鲜事儿", "，", "但", "让", "她", "感到", "新奇", "的是", "，", "在", "她", "的", "工作", "地", "重庆", "，", "如今", "办", "身份证", "用", "的", "照片", "也", "可以", "“", "自", "拍", "”", "“", "自选", "”", "。", "这", "不", "，", "5", "月", "27", "日", "下午", "，", "在", "重庆", "两江", "新区", "人", "和", "派出所", "的", "户籍", "大厅", "，", "伍", "佳", "玲", "就", "体验", "了", "一回", "。", "

", "　", "　", "在", "“", "自助", "拍照", "系统", "”", "前", "坐下", "，", "调整", "好", "位置", "和", "表情", "，", "伍", "佳", "玲", "按", "下", "了", "遥控器", "上的", "“", "拍照", "”", "键", "，", "面前", "的", "显示屏", "上", "立刻", "“", "跳", "”", "出", "了", "照片", "预览", "。", "连", "拍", "三张", "后", "，", "她", "选择", "了", "最", "满意", "的", "一张", "。", "“", "在", "重庆", "能办", "四川", "的", "身份证", "，", "已经", "是", "惊喜", "了", "，", "没想到", "还", "能", "玩", "自拍", "，", "真是", "高大", "上", "。", "”", "伍", "佳", "玲", "笑", "着", "说", "。", "

", "　", "　", "伍", "佳", "玲", "在", "重庆", "经营", "一家", "网店", "，", "几天", "前", "不", "小心", "把", "身份证", "丢了", "，", "挂失", "之后", "想着", "赶快", "补办", "。", "“", "一", "想到", "回老家", "去办", "，", "我", "头", "都", "大", "了", "。", "”", "她", "说", "，", "从", "重庆", "回老家", "，", "单程", "就", "5", "个", "多", "小时", "，", "车费", "加上", "食宿", "没有", "几百元", "下不来", "。", "“", "关键", "是", "现在", "生意", "忙", "，", "要", "走", "个", "两三天", "还", "真", "脱不开", "身", "。", "”", "伍", "佳", "玲", "说", "，", "“", "后来", "听说", "在", "重庆", "也", "能办", "，", "我", "就", "来", "了", "。", "真是太", "方便", "了", "！", "”", "

", "　", "　", "作为", "全国", "居民身份证", "异地", "受理", "首批", "试点", "省市", "之一", "，", "重庆市", "于", "2015", "年", "12", "月", "借助", "自主", "开发", "的", "信息系统", "，", "正式", "启动", "了", "与", "四川省", "的", "跨省", "异地", "受理", "工作", "，", "同时", "全面推行", "市内", "居民身份证", "异地", "受理", "工作", "。", "“", "今年", "5", "月", "16", "日", "我们", "按照", "统一", "要求", "，", "启用", "了", "全国", "居民身份证", "异地", "受理", "工作", "平台", "，", "将", "天津", "、", "湖南", "也", "纳入", "了", "跨省", "异地", "受理", "范围", "。", "”", "市", "公安局", "治安管理", "总队", "人口", "支队", "副", "支队长", "王", "劲松", "介绍", "，", "截至", "目前", "，", "全市", "共", "受理", "异地", "证", "10", "万余人次", "，", "其中", "跨省", "异地", "证", "3200", "多张", "。", "

", "　", "　", "翻开", "人", "和", "派出所", "今年以来", "的", "异地", "受理", "登记表", "，", "涉及", "了", "全市", "38", "个", "区县", "中", "的", "34", "个", "。", "“", "大部分", "是", "务工人员", "，", "有", "的", "边远", "区县", "回去", "一趟", "单程", "就要", "五六个", "小时", "。", "”", "副所长", "伍庆勇说", "，", "“", "能", "让", "他们", "少跑路", "，", "少", "请假", "，", "他们", "满意", "，", "我们", "也", "高兴", "。", "”", "

", "　", "　", "临近", "下班", "，", "派出所", "户籍", "大厅", "迎来", "了位", "“", "川妹子", "”", "刘", "倩", "，", "是", "西南", "大学", "的", "学生", "，", "接到", "电话", "后来", "取", "办好", "的", "身份证", "。", "“", "我", "还", "怕", "来", "晚", "了", "你们", "会", "下班", "呢", "。", "”", "刘", "倩", "说", "。", "“", "不会", "，", "你", "电话", "里", "说", "要来", "，", "我们", "就会等", "。", "”", "窗口", "民警", "冯璐璐", "告诉", "她", "，", "“", "我们", "还", "公布", "了", "预约", "电话", "，", "如果", "大家", "平时", "上班", "来", "不了", "，", "周末", "也", "可以", "预约", "。", "”", "

", "　", "　", "“", "小", "证件", "”", "事关", "“", "大", "和谐", "”", "。", "为", "做好", "这项", "工作", "，", "重庆市公安局", "从", "人员配备", "、", "业务培训", "、", "设备", "保障", "、", "作风", "建设", "等", "各", "方面", "统筹", "部署", "，", "扎实", "推进", "。", "“", "6", "月", "6", "日", "之前", "，", "我们", "还", "将", "把", "浙江", "、", "安徽", "、", "江西", "、", "湖北", "四省", "纳入", "跨省", "异地", "受理", "范围", "，", "全市", "所有", "的", "区县", "分局", "都", "要", "设立", "跨省", "异地", "受理", "点", "。", "”", "市", "公安局", "副局长", "郭金", "严说", "，", "要", "注重", "运用", "科技", "手段", "不断", "提高", "工作效率", "，", "确保", "改革", "举措", "落地生根", "，", "把", "更", "多", "的", "获得", "感", "“", "递", "”", "到", "群众", "手上", "。", "

", "　", "　", "惠民", "之举", "赢得", "了", "山城", "群众", "点赞", "。", "“", "改革", "不", "在", "大小", "，", "利民", "就好", "；", "红利", "不", "在", "厚薄", "，", "真实", "为贵", "。", "小小", "身份证", "，", "也", "是", "大", "文章", "。", "”", "网友", "“", "江边", "打鱼人", "”", "评价", "道", "。"]

1. **参考文献**
2. niuox. 中文分词算法之最大正向匹配算法（Python版）[EB/OL]. <http://blog.csdn.net/niuox/article/details/11248567> , 2013-09-06
3. fxsjy. jieba[EB/OL]. <https://github.com/fxsjy/jieba>
4. 52nlp. 中文分词入门之最大匹配法[EB/OL]. <http://www.52nlp.cn/maximum-matching-method-of-chinese-word-segmentation>.
5. hfgang. 中文分词基础原则及正向最大匹配法、逆向最大匹配法、双向最大匹配法的分析[EB/OL]. <http://blog.sina.com.cn/s/blog_53daccf401011t74.html> .
6. **源代码**

**统计词典：**

#coding=utf-8

f = open('jieba.dict.utf8.txt')

line = f.readline().decode('utf-8')

count = 0

max\_length = 0

count\_one = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

while line:

line\_array = line.split(" ");

line\_array\_first\_length = len(line\_array[0])

if (max\_length < line\_array\_first\_length):

max\_length = line\_array\_first\_length

count\_one[line\_array\_first\_length] = count\_one[line\_array\_first\_length] + 1

count =+ count + 1;

line = f.readline().decode('utf-8')

print u'总词数: %s'%count

print u'最长词字数: %s'%max\_length

print count\_one

f.close

**分词伪代码：**

window\_size=16

def max\_match\_segment(line, dic):

# write your code here

chars = line.decode("utf8")

words = []

idx = 0

while idx < len(chars):

matched = False

for i in xrange(window\_size, 0, -1):

cand=chars[idx:idx+i].encode("utf8")

if cand in dic:

words.append(cand)

matched = True

break

if not matched:

i = 1

words.append(chars[idx].encode("utf8"))

idx += i

return words

if \_\_name\_\_=="\_\_main\_\_":

try:

fpi=open(sys.argv[1], "r")

except:

print >> sys.stderr, "failed to open file"

sys.exit(1)

try:

dic = pickle.load(open(sys.argv[2], "r"))

except:

print >> sys.stderr, "failed to load dict %s" % sys.argv[2]

sys.exit(1)

try:

fpo = open("out.txt","w")

except:

print >> sys.stderr, "failed to load out.txt"

sys.exit(1)

for line in fpi:

fpo.write("\t".join( max\_match\_segment(line.strip(), dic) ))