《中文分词》实验报告

宋健 2017210752

1.基于字典匹配的分词

按照一定策略将带分析的汉字串与一个字典中的词条进行匹配,若在词典中找到某个字符串,则匹配成功,也称为机械匹配。按照扫描方向的不同可分为正向匹配和逆向匹配;按照长度不同可分为最长匹配和最小匹配。

1.1 正向最大匹配 MM

- 1)从左向右取待切分语句的前 m 个字作为匹配字段, m 最大为字典中词条的最大词长;
- 2)查找字典词条并进行匹配, 若匹配成功, 则将这个匹配字段作为一个词切分出来;
- 3)若匹配不成功,则将这个匹配字段的最后一个字去掉,剩下来的字符串作为新的匹配字段,继续进行再次匹配;
 - 4)重复 3), 直到切分出所有词为止。

1.2 逆向最大匹配 RMM

- 1)从右向左取待切分语句的前 m 个字作为匹配字段, m 最大为字典中词条的最大词长;
- 2)查找字典词条并进行匹配, 若匹配成功, 则将这个匹配字段作为一个词切分出来;
- 3)若匹配不成功,则将这个匹配字段的第一个字去掉,剩下来的字符串作为新的匹配字段,继续进行再次匹配;
 - 4)重复3), 直到切分出所有词为止。

1.3 双向最大匹配 BM

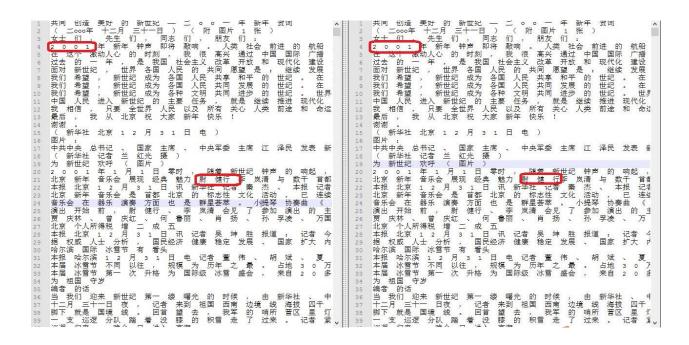
- 1) 先进行 MM 和 BM;
- 2)如果正反向分词结果词数不同.则取分词数量较少的那个;
- 3)如果分词结果词数相同:
 - a.分词结果相同,就说明没有歧义,可返回任意一个;
 - b.分词结果不同,返回其中单字较少的那个。

1.4 实验结果分析

本次实验分别实现了 MM、RMM 和 BM 算法,并分析比对其结果,其中,使用的字典由训练数据生成。三种算法的评分结果如下(对应的文件见 3.):

算法	训练集性能			测试集性能			
	Р	R	F	Р	R	F	
MM	97.7%	97.0%	97.4%	84.3%	90.7%	87.4%	
RMM	98.0%	97.3%	97.7%	84.5%	90.9%	87.6%	
ВМ	98.1%	97.3%	97.7%	84.5%	90.9%	87.6%	

可以看到,在本例中,BM和RMM在性能上基本相同,均比MM性能略好。下图左为MM的结果,右为RMM的结果。可以看到,二者对于非登录词同样性能低下(例如"2001年"),同时,正向、逆向对于语义消歧各有优劣(例如"尉健行")。总体而言,基于字典的方法配得上"机械分词"这一别称,属于比较暴力的分词方法。



2.基于序列标注和 CRF 的分词

2.1 算法原理

基于词典的分词过度依赖词典和规则库,因此对于歧义词和未登录词的识别能力较低; 其优点是速度快,效率高。CRF 的基本思路是对汉字进行标注即由字构词(组词),不仅 考虑了文字词语出现的频率信息,同时考虑上下文语境,具备较好的学习能力,因此其对 歧义词和未登录词的识别都具有良好的效果;其不足之处是训练周期较长,运营时计算量 较大,效率不如词典。

2.2 实验结果分析

本次实验采用 B、M、E、S 四种标签标注,使用 CRF++开源包对 CRF 算法进行仿真,实验结果如下(对应的文件见 3.):

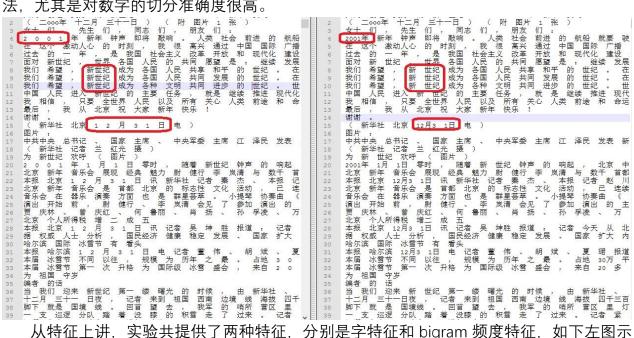
算法	训练集性能			测试集性能		
异 <i>広</i>	Р	R	F	Р	R	F
CRF_3f3c4t(template)	100%	100%	100%	92.2%	93.5%	92.8%
CRF_3f3c2t(template)	99.4%	99.4%	99.4%	92.3%	93.7%	93.0%
CRF_3f2c4t(template)	99.9%	99.9%	99.9%	92.4%	93.9%	93.1%
CRF_3f1c4t(template)	100%	100%	100%	92.5%	94.1%	93.3%
CRF_3f1c6t(template)	100%	100%	100%	92.4%	94.1%	93.3%
CRF_3f1c6t4(template4)	99.9%	99.9%	99.9%	92.4%	94.2%	93.2%
CRF_5f3c4t(template1)	99.7%	99.7%	99.7%	92.3%	93.7%	93.0%
CRF_7f3c4t(template2)	100%	100%	100%	92.2%	93.5%	92.9%
CRF_9f3c4t(template3)	100%	100%	100%	92.0%	93.3%	92.7%

算法名中 "f3" 表示在训练模板时取所有频度 3 以上的 n-gram 组合, "c4" 表示训练的精度阈值为 4.0, 其余以此类推。实验中共定义 4 个不同的模板文件(template、template1-4),不同的算法对应使用的模板见上表括号中标注。其中,template 为 crf++ 给出的样例模板,template1-3分别调整窗长为5、7和9,template4中去除trigram组合,

具体请见相应的模板文件。以 template 为例(如下图示),使用字特征,窗长为 3,其中 U08-09 相当于 Bigram,U05-07 相当于 Trigram。其余模板以此类推。

```
# Unigram
    U00:%x[-2,0]
 2
    U01:%x[-1,0]
    U02:%x[0,0]
    U03:%x[1,0]
    U04:%x[2,0]
    U05: x[-2,0]/x[-1,0]/x[0,0]
 8
    U06: x[-1,0]/x[0,0]/x[1,0]
    U07:%x[0,0]/%x[1,0]/%x[2,0]
10
    U08: x[-1,0]/x[0,0]
    U09:%x[0,0]/%x[1,0]
12
13
    # Bigram
14
    В
15
```

从结果中可以看到,整体而言,CRF 比基于词典的方法性能更优;CRF 对于训练集的你和程度相当高,在训练集上则在 f=1,窗长=3 时达到最优。下图中,左边是 BM 的结果,右边为该模型的测试结果。可以看到 CRF 对于未登录词的识别能力明显强于基于词典的方法,尤其是对数字的切分准确度很高。



从特征上讲,实验共提供了两种特征,分别是字特征和 bigram 频度特征,如下左图示,第一列为字特征,中间列为 B、E、M、S 标注,最后一列为当前字开始的 bigram 在全文中出现的次数。但是实际实验中仅使用了字特征,因为在尝试使用频度特征时,待提取特征数过多(使用 template 模板),导致内存崩溃,因而放弃。

```
CRF++: Yet Another CRF Tool Kit
Copyright (C) 2005-2013 Taku Kudo, All rights reserved.
       迈 B 33
       向 E 3
                                                                                   reading training data:
Done!46.22 s
       充 B 115
       满 E 13
                                                                                   Number of sentences: 1
Number of features: 359100672
Number of thread(s): 24
       希
            B 504
       望 E 38
                                                                                                            3
0.00010
4.00000
20
       的 S 468
8
       新 S 54
                                                                                   shrinking size:
       世 B 504
```

3.文件清单

文件/文件夹	备注
crf_model	训练得到的 CRF 模型集合
icwb2-data	数据集包
crfpp-master	CRF 包
result	分词结果集合
score	对各分词结果的评分文件集合
build_dict.py	由训练数据生成词典
tag_training.py	由训练数据生成标注训练集的方法
hw1_MM.py	MM 方法
hw1_RMM.py	RM 方法
hw1_BM.py	BM 方法
hw1_CRF.py	CRF 方法 (对应不同模型即为 2.基于序列标注和 CRF 的分
	词中所列各种算法)
pku_training_unsegged.utf8	还原的未分词训练数据