CRF理论、工具包的使用及在NE上的应用

廖先桃

2006.4.6

提纲

- CRF理论
- CRF工具包的使用
- 基于CRF的NE识别
- 下一步工作

CRF理论

- CRF是Conditional Random Fields的缩写，即条件随机域
- CRF理论相关PPT由赵世奇友情赞助☺

提纲

- CRF理论
- CRF工具包的使用
- 基于CRF的NE识别
- 下一步工作

CRF工具包的使用

- CRF工具包
  - http://crf.sourceforge.net/doc/
    - java写的CRF工具包，有很详细的说明文档。
  - FlexCRFs: Flexible Conditional Random Fields
    - 作者:Xuan-Hieu Phan 和 Le-Minh Nguyen
    - C++语言实现
    - 下载地址：http://www.jaist.ac.jp/~hieuxuan/flexcrfs/flexcrfs.html
  - CRF++: Yet Another CRF toolkit
    - C++语言实现，有windows下运行的exe程序以及Linux下运行的版本
    - 下载地址： http://chasen.org/~taku/software/CRF++/#features

CRF++的使用(1)

- CRF++的安装
- 训练语料的格式
- 特征模板的格式
- 训练模型
- 识别
- 评测

CRF++的使用(2)

- CRF++的安装
  - 编译器要求：C++编译器(gcc 3.0或更高)
  - Linux安装命令(依次执行)：

```
% ./configure
% make
% su
# make install
```

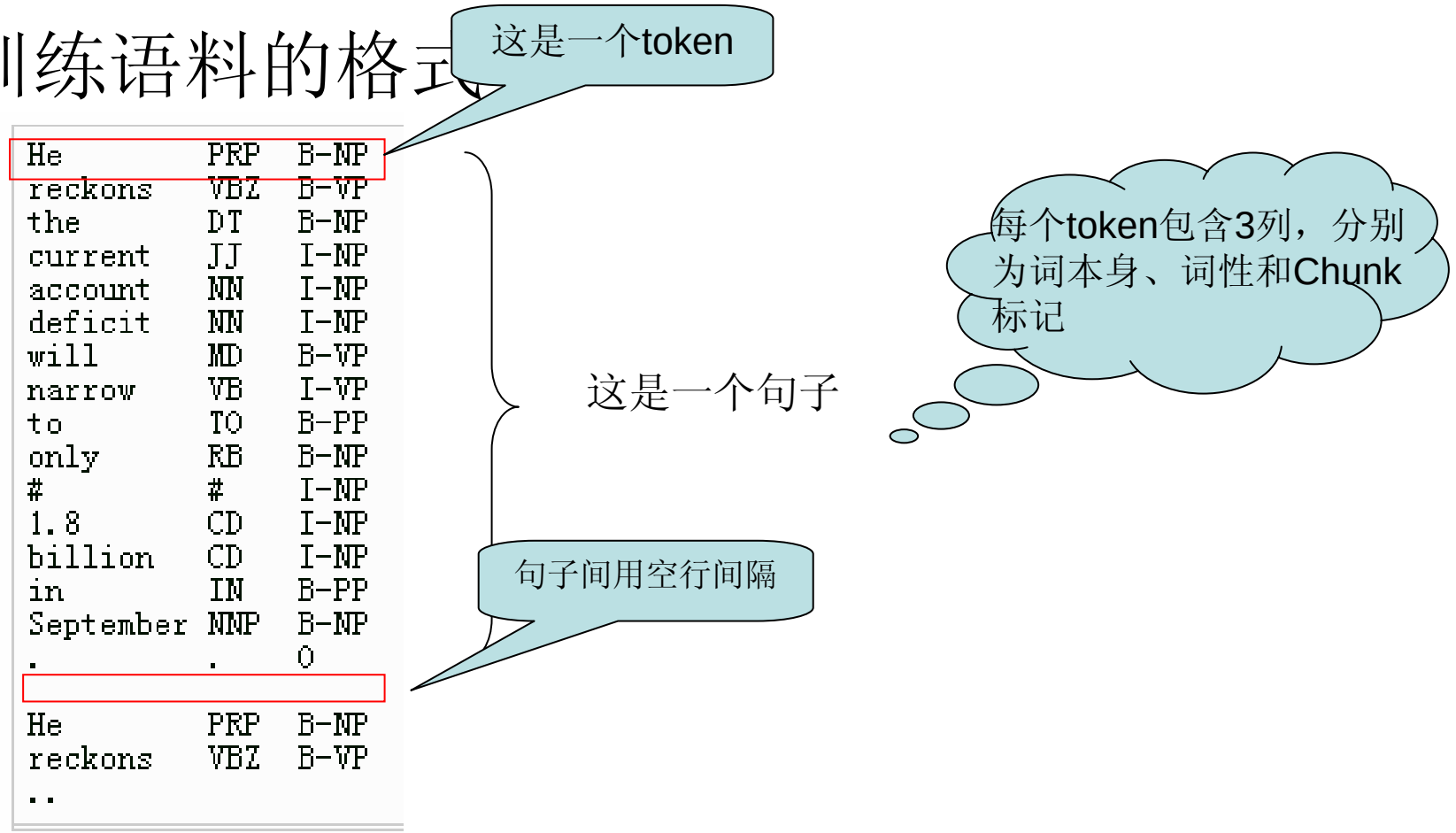
  - 注意：必须具有root帐号才能安装成功

CRF++的使用(3)

- 训练语料的格式
  - 训练和测试文件必须包含多个tokens
  - 每个token包含多个列
  - token的定义可根据具体的任务，如词、词性等
  - 每个token必须写在一行，且各列之间用空格或制表格间隔
  - 一个token的序列可构成一个sentence，sentence之间用一个空行间隔

CRF++的使用(4)

- 训练语料的格式



CRF++的使用(3)

- 特征模板的格式
  - 模板的基本格式为*%x[row,col]*，它用于确定输入数据中的一个token
  - 其中，row确定与当前的token的相对行数。col用于确定绝对列数。

CRF++的使用(3)

- 特征模板的例子
  - 训练语料

	col0	col1	col2	
	Input: Data			
r-2	He	PRP	B-NP	
r-1	reckons	VBZ	B-VP	
r0	the	DT	B-NP	<< 当前的token
r1	current	JJ	I-NP	
r2	account	NN	I-NP	

  - 特征模板

```
template      expanded feature
%x[0,0]             the
%x[0,1]             DT
%x[-1,0]            rokens
%x[-2,1]            PRP
%x[0,0]/%x[0,1]     the/DT
ABC%x[0,1]123       ABCthe123
```

CRF++的使用(3)

- 特征模板的类型
  - 第一种以字母U开头，为Unigram template。
  - 当模板前加上U之后，CRF会自动生成一个特征函数集合(func1 ... funcN)，如：

```
func1 = if (output = B-NP and feature="U01:DT") return 1 else return 0
func2 = if (output = I-NP and feature="U01:DT") return 1 else return 0
func3 = if (output = O and feature="U01:DT") return 1  else return 0
....
funcXX = if (output = B-NP and feature="U01:NN") return 1  else return 0
funcXY = if (output = O and feature="U01:NN") return 1  else return 0
...
```

CRF++的使用(3)

- 特征模板的种类
  - 一个模型生成的特征函数的个数总数为L*N，其中L是输出的类别数，N是根据给定的template扩展出的独立串(unique string)的数目。

CRF++的使用(3)

- 特征模板的种类
  - 第二种特征模板以B开头，即Bigram template
  - 它用于描述Bigram特征。系统将自动产生当前输出token与前一个输出token的组合。产生的可区分的特征的总数是L*L*N，其中L是输出类别数，N是这个模板产生的unique features数 。
  - 优点：提高识别效果
  - 缺点：当类别数很大的时候，这种类型会产生许多可区分的特征，这将会导致训练和测试的效率降低。

CRF++的使用(3)

- 特征模板的类型
  - 两种模板的区别
  - 注意：Unigram/Bigram是指输出token的Unigram/Bigrams，而不是特征
  - unigram: |output tag| × |从模板中扩展的所有可能串|
  - bigram: |output tag| × |output tag| × |从模板中扩展的所有可能串|

CRF++的使用(3)

- 特征模板的

#表示注释，将被忽略

```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[
U03:%x[
U04:%x[
U05:%x[
U06:%x[0,0]/%x[1,0]

U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]q
U13:%x[1,1]
U14:%x[2,1]
U15:%x[-2,1]/%x[-1,1]
U16:%x[-1,1]/%x[0,1]
U17:%x[0,1]/%x[1,1]
U18:%x[1,1]/%x[2,1]

U20:%x[-2,1]/%x[-1,1]/%x[0,1]
U21:%x[-1,1]/%x[0,1]/%x[1,1]
U22:%x[0,1]/%x[1,1]/%x[2,1]

# Bigram
B
```

为区分特征给特征的编号

```
The         DT   B-NP
pen         NN   I-NP
is          VB   B-VP << CURRENT TOKEN
a           DT   B-NP
```

得到的结果都为DT

```
U01:%x[-2,1]
U02:%x[1,1]
```

CRF++的使用(3)

- 训练模型
  - 使用crf_learn命令

```
% crf_learn template_file train_file model_file
```

  - 其中，template_file和train_file需由使用者事先准备好。crf_learn将生成训练后的模型并存放在model_file中。

CRF++的使用(3)

- 训练模型
  - 屏幕显示信息

```
% crf_learn template_file train_file model_file
CRF++: Yet Another CRF Tool Kit
```

iter：迭代次数
terr：和tags相关的错误率(错误的tag数/所有的tag数)
serr：与sentence相关的错误率（错误的sentence数/所有的sentence数）
obj：当前对象的值。当这个值收敛到一个确定的值是，CRF模型将停止迭代
diff：与上一个对象值之间的相对差

```
iter=0 terr=0.7494725738 serr=1 obj=2082.968899 diff=1
iter=1 terr=0.1671940928 serr=0.8831168831 obj=1406.329356 diff=0.3248438053
iter=2 terr=0.1503164557 serr=0.8831168831 obj=626.9159973 diff=0.5542182244
```

CRF++的使用(3)

- 识别
  - 使用crf_test 命令

```
% crf_test -m model_file test_files ...
```

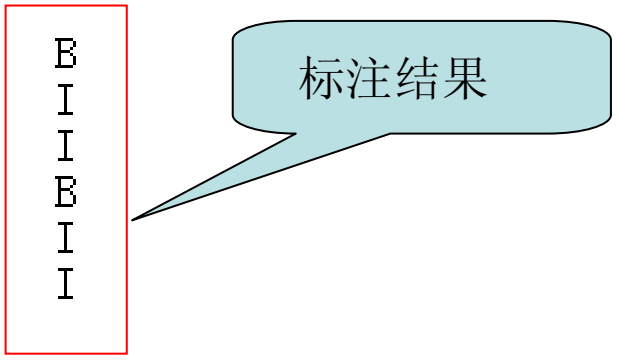
  - 其中，model_file是crf_learn创建的。在测试过程中，使用者不需要指定template file，因为，mode file已经有了template的信息。test_file是使用者想要标注序列标记的测试语料。这个文件的书写格式应该与训练文件一致。

CRF++的使用(3)

- 识别

```
% crf_test -m model test.data

Rockwell          NNP      B       B
International     NNP      I       I
Corp.             NNP      I       I
’s                POS      B       B
Tulsa             NNP      I       I
unit              NN       I       I
..
```



CRF++的使用(3)

- 评测
  - 使用评测程序conlleval.pl(CoNLL 2000评测程序)
    - 下载地址：http://www.cnts.ua.ac.be/conll2000/chunking/output.html
  - 测试语料的格式与前边的识别结果一致，但是token的每一列之间必须用空格间隔，否则评测程序不能正常执行
  - 评测指令：perl conlleval.pl < output.txt
    - Output.txt为待评测文件

CRF++的使用(3)

- 评测结果

```
processed 961 tokens with 459 phrases; found: 539 phrases; correct: 371.
accuracy:  84.08%; precision:  68.83%; recall:  80.83%; FB1:  74.35
             ADJP: precision:   0.00%; recall:   0.00%; FB1:   0.00
             ADVP: precision:  45.45%; recall:  62.50%; FB1:  52.63
               NP: precision:  64.98%; recall:  78.63%; FB1:  71.16
               PP: precision:  83.18%; recall:  98.89%; FB1:  90.36
             SBAR: precision:  66.67%; recall:  33.33%; FB1:  44.44
               VP: precision:  69.00%; recall:  79.31%; FB1:  73.80
```

识别类别

相应的指标

提纲

- CRF理论
- CRF工具包的使用
- 基于CRF的NE识别
- 下一步工作

基于CRF的NE识别(1)

- 训练语料：经IRLAS转换后的北大富士通1月份语料，共37426句
- 测试语料：经IRLAS转换后的北大富士通6月份语料前10000句
- 实验：
  - 采用复杂特征
  - 采用简单特征

基于CRF的NE识别(2)

- 实验1
- 特征
  - CRF特征16种：P-2 W-1 P-1 W0 P0 W1 P1 P-2/P0 P-2/P1 P-1/P0/P1 P0/P1 P2 P0/P2 P0/P1/P2 W0/P0 W-1/P-1 W1/P1
  - ME特征23种：T P-2 W-1 T-1 P-1 W0 P0 W1 P1 P-2/P0 P-2/P1 T-1/P0 P-1/T-1/P0 P-1/P0/P1 P0/P1 P2 P0/P2 P0/P1/P2 W0/P0 W-1/P-1 P-1/T-1 W1/P1 S E

基于CRF的NE识别(2)

- 特征模板

```
# Unigram
U00:%x[-2,1]
U01:%x[-1,0]
U02:%x[-1,1]
U03:%x[0,0]
U04:%x[0,1]
U05:%x[1,0]
U06:%x[1,1]
U07:%x[-2,1]/%x[0,1]
U08:%x[-2,1]/%x[1,1]
U09:%x[-1,1]/%x[0,1]/%x[1,1]
U10:%x[0,1]/%x[1,1]
U11:%x[2,1]
U12:%x[0,1]/%x[2,1]
U13:%x[0,1]/%x[1,1]/%x[2,1]
U14:%x[0,0]/%x[0,1]
U15:%x[-1,0]/%x[-1,1]
U16:%x[1,0]/%x[1,1]
# Bigram
B
```

基于CRF的NE识别(2)

- 实验1评测结果

表 1 利用复杂特征的基于 CRF 的 NE 识别的识别结果

类别	模型	识别个数	准确率	召回率	F 值
人名	CRF	3237	96.66%	96.54%	96.60%
	ME	3266	98.28	85.70	91.56
机构名	CRF	2513	89.65%	86.99%;	88.30%
	ME	2592	84.64	75.27	79.68
地名	CRF	5742	93.94%	95.42%	94.67%
	ME	5689	91.15	93.55	92.33
专有名词	CRF	524	84.35%	79.07%	81.63%
	ME	560	91.12	64.11	75.26
时间	CRF	32	90.63%	76.32%	82.86%
	ME	38	76	50	60.32
日期	CRF	1635	97.86%	97.21%	97.53%
	ME	1651	95.47	85.52	90.22
数量短语	CRF	6986	97.92%	98.40%	98.16%
	ME	7009	95.91	93.12	94.49
总的结果	CRF	20669	95.25%	95.21%	95.23
	ME	20805	93.37	88.38	90.81

基于CRF的NE识别(2)

- 实验2特征
  - CRF特征（7种）：P-2 W-1 P-1 W0 P0 W1 P1
  - ME特征（12种）：P-2 W-1 T-1 P-1 W0 P0 W1 P1 W2 P2 S E

基于CRF的NE识别(2)

- 实验2特征模板

```
# Unigram
U00:%x[-2,1]
U01:%x[-1,0]
U02:%x[-1,1]
U03:%x[0,0]
U04:%x[0,1]
U05:%x[1,0]
U06:%x[1,1]
U11:%x[2,1]
# Bigram
B
```

基于CRF的NE识别(2)

- 实验2评测结果

表 2 CRF 和 ME 仅利用简单特征的实验对比

类别	模型	识别个数	准确率	召回率	F 值
人名	CRF	3237	95.90%	96.76%	96.33%
	ME	3266	91.59	66.32	76.93
机构名	CRF	2513	89.45%	86.10%	87.74
	ME	2592	78.42	69.11	73.47
地名	CRF	5742	93.77%	95.10%	94.43%
	ME	5689	90.64	91.58	91.11
专有名词	CRF	524	83.84%	78.89%	81.29%
	ME	560	90.21	23.04	36.70
时间	CRF	30	90.00%	71.05%	79.41
	ME	38	39.13	23.68	29.51
日期	CRF	1635	97.45%	97.33%	97.39%
	ME	1651	86.78	66.06	75.02
数量短语	CRF	6986	97.93%	98.23%	98.08%
	ME	7009	93.47	84.00	88.48
总的结果	CRF	20669	95.03%	94.98%	95.00
	ME	20805	89.87	78.27	83.67

基于CRF的NE识别(2)

- 基于CRF的NE识别总结
  - 与ME相比，在同等条件下，CRF的实验结果明显优于ME，说明CRF的性能更好。
  - CRF对特征的融合能力比较强，即使是给出的很简单的特征，它仍然能达到较高的性能。而ME对特征的融合能力相对较弱。如果只给出简单的特征，它的性能，尤其是对复杂NE的识别效果会有较大的降低。

基于CRF的NE识别(2)

- 基于CRF的NE识别总结
  - CRF能更好的克服数据稀疏现象。对于实例较少的时间类NE来说，CRF的识别效果明显高于ME的识别结果。
  - 缺点：训练模型的时间比ME更长，且获得的模型很大，在一般的PC机上无法运行。

提纲

- CRF理论
- CRF工具包的使用
- 基于CRF的NE识别
- 下一步工作

下一步工作

- 进一步完成CRF的实验，找到CRF比ME性能好的原因
- 对NE识别结果进行错误分析，针对错误改进
- 修改Linux环境下的CRF代码，使之能在Windows下运行，并成为性能更好的NE模块

Thanks a lot!