layout: post
title: FMM/RMM Chinese Word Segmentation
subtitle: Natural Language Processing homework
date: 2017-11-13
author: WeiXiao
header-img: img/tag-bg-o.jpg
catalog: true
tags:
- NLP

- Python

任务定义

- Chinese word segmentation
  - This task used PKU_training set and PKU_testing set as input set.After training,set output.utf8 as a imput to score compare with gold_data.
- 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=

$$(2*P*R)/(P+R)$$

输入输出

- input
  - pku_train.utf8(local:1.utf8)
  - pku_test.utf8(local:2.utf8)

- output
    - 3.utf8
- test
    - pku_test_gold(local:4.utf8)
    - score

方法描述

- Using way

```
python fmm.py 1.utf8 2.utf8 3.utf8 FMM/RMM
```

- Training two way
    - FMM

```
def FMM(mydict, sentence):#前向最大匹配
    ruleChar = []
    ruleChar.extend(numMath)
    result = []
    start = 0
    senlen = len(sentence)
    while start < senlen:
        curword = sentence[start]
        maxlen = 1
        # 首先查看是否可以匹配特殊规则
        if curword in numMath:
            maxlen = rules(sentence, start)
        # 寻找以当前字开头的最长词
        if curword in mydict:
            words = mydict[curword]
            for item in words:
                itemlen = len(item)
                if sentence[start:start + itemlen] == item and itemlen > maxl
en:
                    maxlen = itemlen
        result.append(sentence[start:start + maxlen])
        start = start + maxlen
    return result
```

- RMM

```
    def RMM(mydict, sentence):
    ruleChar =[]
    ruleChar.extend(numMath)
    result=[]
    senlen=len(sentence)
    start=senlen
    while(start<senlen):
        curword=sentence[start]
        maxlen=1
        if(curword in mydict):
            words = mydict[curword]
            for item in words:
                itemlen=len(item)
                if sentence[start-itemlen:start] ==item and itemlen>maxlen:
                    maxlen=itemlen
        if curword in numMath_suffix or curword in numMath:
            maxlen=rrules(sentence,start)
        result.append(sentence[start-itemlen:start])
        start = start -maxlen
    return result
```

结果分析

- FMM

```
TRUE WORDS RECALL:  1.000
TEST WORDS PRECISION:   1.000
=== SUMMARY:
```

```
=== TOTAL INSERTIONS:	2144
=== TOTAL DELETIONS:	4594
=== TOTAL SUBSTITUTIONS:	6996
=== TOTAL NCHANGE:	13734
=== TOTAL TRUE WORD COUNT:	106822
=== TOTAL TEST WORD COUNT:	104372
=== TOTAL TRUE WORDS RECALL:	0.892
=== TOTAL TEST WORDS PRECISION:	0.912
=== F MEASURE:	0.902
=== OOV Rate:	0.943
=== OOV Recall Rate:	0.886
=== IV Recall Rate:	0.988
###	4.utf8	2144	4594	6996	13734	106822	104372	0.892	0.912	0
.902	0.943	0.886	0.988
```

- Using score to test the presious of FMM,the result is:
  - gold
    - presious:1.000 Recall:1.000
  - FMM
    - presicous:0.912 Recall:0.892
- RMM

```
=== SUMMARY:
=== TOTAL INSERTIONS:	1462
=== TOTAL DELETIONS:	6531
=== TOTAL SUBSTITUTIONS:	10724
=== TOTAL NCHANGE:	18717
=== TOTAL TRUE WORD COUNT:	109441
=== TOTAL TEST WORD COUNT:	104372
=== TOTAL TRUE WORDS RECALL:	0.842
=== TOTAL TEST WORDS PRECISION:	0.883
=== F MEASURE:	0.862
=== OOV Rate:	0.936
=== OOV Recall Rate:	0.840
=== IV Recall Rate: 0.883
### 4.utf8  1462    6531    10724   18717   109441  104372  0.842   0.883   0
.862   0.936   0.840   0.883
```

- Using score to test the presious of RMM,the result is:
    - gold
        - presious:1.000 Recall:1.000
    - FMM
        - presicous:0.883 Recall:0.842

源码运行环境

- Mac OS 10.13
- Pycharm
- Python 3.6.1