统计自然语言处理基础
项目报告

王巍

班号: 1503103
学号: 1150340114

August, 17th 2018

1 分词任务

1.1 定义

中文分词指的是将一个汉字序列切分成一个个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。[1]

1.2 实例

- 例如一个较为简单的分词任务:

  我国律师工作是随着改革开放和民主法制建设的加强而建立和发展起来的。

  的分词结果为:

  我国/律师/工作/是/随着/改革/开放/和/民主/法制/建设/的/加强/而/建立/和/发展/起来/的/。

- 再例如一个由于中文的特性, 很可能会被错误分词的例子:

  请把手拿开

  的分词结果为:

  请把/手/拿开

  但由于“把手”本身也组成词, 因此也有可能形成错误的分词:

  请/把手/拿开

- 除此之外, 还存在本身具有歧义的句子, 比如:

  乒乓球拍卖完了

  “乒乓球/拍卖”以及“ 乒乓/球拍/卖”都是从语法上合理的分词, 但语义上的合理性有所不同.

1.3 常用方法

- 基于字典匹配的方法

  基于字典匹配的方法又称为机械分词法, 根据扫描的方向和不同长度的匹配情况, 可以分为正向最大匹配法, 逆向最大匹配法, 最少切分以及双向最大匹配法, 在本次项目中, 使用正向最大匹配法结合规则取得了较好的效果, 算法的细节将在下一节讨论.

- 基于句法的分词方法

  基于句法的分词方法使用计算机模拟人对句子的理解, 达到识别词的效果, 配合句法语义系统使用. 对于句子本身从语法上没有歧义, 但因为构词而容易产生错误切分的句子, 基于句法的分词方法能够较好的解决这种问题. 如上一小节中的第二个例子, 错误的分词方法在语法上存在错误. 但是, 中文的句子构成十分复杂, 句法分析通常难以做到较高的准确率.

- 基于统计的分词方法

  基于统计的分词方法包括 HMM, CRF 等, 这类方法将分词视为分类问题, 考察一个字的有限上下文根据统计概率确定其标签, 并通过标签解码出分词结果. 在消除歧义方面, n-gram 方法可以在一定程度上解决句子的固有歧义. 如上一小节中的第三个例子, "乒乓球/拍卖" 的分词方式出现的频率显然低于 "乒乓/球拍/卖" 的分词方式, 通过这种方法可以选出更为合理的分词.

  本次项目中也实现了基于 HMM 算法的分词, 在没有结合字典, 规则, 且训练语料较小的情况下, HMM 算法的表现不算很好, 算法的细节也将在下一节中讨论.

2 算法

2.1 最大正向匹配算法 (MMSEG)

2.1.1 形式化定义

给定词典 $D(w, f)$ (w 是单词文本, f 是单词频率). 对于汉字序列 S, 在 D 上找到所有可能的词块序列 $C_{1,2\cdots,n}$ ($C_i = w_1[\_w_2[\_w_3]]$, w 是 D 中的单词). 使得对于算法的词块评分函数 P, 词块序列的评分之和 $\sum_{i=1}^{n} P(C_i)$ 之和最大的词块 $C^{\max}_{1,2\cdots,n}$ 即为算法的输出分词方案.

2.1.2 建模方法

最大正向匹配算法 (MMSEG) 是基于字典匹配的分词方法, 简单高效, 在一定程度上能够解决纯词典方案无法解决的分词歧义问题.

MMSEG 算法分为简单算法和复杂算法, 其中简单算法对于一个句子, 找到其所有的可能前缀; 而复杂算法对于一个句子, 前向寻找其可能的 "词块", 即三个或以下个词组, 对可能的词组应用四个启发式规则来解决分歧, 这四个规则即是算法中的词块评分函数, 分别是:

- 备选词块的长度最大:

  例如 "长春市长春药店" 的分词方案有:

  - 长春市 _ 长春 _ 药店 _
  - 长春市 _ 长 _ 春药 _
  - 长春 _ 市长 _ 春药 _
  - 长春 _ 市 _ 长春 _
  - 长 _ 春 _ 市长 _

  第一种组合长度最长, 因此采用第一种分词方案.

- 备选词组合的平均词长最大:

  例如 "国际化" 的分词方案有:

  - 国际化 _
  - 国际 _ 化 _
  - 国 _ 际 _ 化 _

  使用规则 1 无法过滤, 但第一种分词方案的平均长度 3 为最大, 因此采用第一种分词方案.

- 备选词组合的变化最小

  例如 “北京大学生” 的分词方案有:

  - 北京大学 _ 生 _
  - 北京 _ 大学生
  - 北京 _ 大学 _ 生 _
  - 北京 _ 大 _ 学生 _
  - 北 _ 京 _ 大学生 _

  经过前两条规则过滤, 剩余:

  - 北京大学 _ 生 _
  - 北京 _ 大学生

  本条规则在应用中以方差作为度量, 第一种方案方差为 1.5, 第二种为 0.5. 因此选择第二种方案

- 备选词组合中，单字词的出现频率统计值最高: 例如 “设施和服务” 的分词方案有:

  - 设施 _ 和服 _ 务 _
  - 设施 _ 和 _ 服务 _
  - 设 _ 施 _ 和服 _

  经前面规则过滤得:

  - 设施 _ 和服 _ 务 _
  - 设施 _ 和 _ 服务 _

  根据本条规则, 显然单字 “和” 的频率要大于单字 “务” 的频率, 因此选择第二种方案

此外, 在实际应用中, 数字和 url 等的切分需要通过规则来加以约束, 具体方案是对数字和 url 分别用占位符替换, 并备份原数字和 url, 在分词结束后进行还原.

基于字典匹配的方法无法正确地识别新词, 因此对训练语料或词典的大小有一定的要求, 在实验时, 也发现随着词典数量的增加, 算法在开发集上的准确率会逐步提高, 但使用多个来源不同的词典时, 单词对应的频率将失去意义.

2.1.3 算法伪代码

Algorithm MMSEG-Tokenizer(sentence, dictionary)

Require: chunk has structure with properties corresponding to measurements of 4 rules

```
backups ← special patterns in sentence
replace special patterns in sentence with placeholder
words ← empty list
cursor ← 0
repeat
    chunks ← Get-Chunk(sentence[cursor:])
    if chunks.length > 1 then
        chunks ← chunk with max total length {rule 1}
    end if
    if chunks.length > 1 then
        chunks ← chunk with max average length {rule 2}
    end if
    if chunks.length > 1 then
        chunks ← chunk with min standard deviation {rule 3}
    end if
    if chunks.length > 1 then
        chunks ← chunk with max single word frequency {rule 4}
    end if
    if chunks.length > 1 then
        chunks ← random chunk in chunk {ambiguity cannot be resolved}
    end if
    cursor ← cursor + chunks[0].length
    add chunk to words
until cursor == length of sentence
result ← word joined by delimiters
restore digits and urls with backups RETURN result
```

Algorithm Get-Chunk(sentence)

Require: prefix trie already built from dictionary or corpus

```
chunks ← empty list
for word_1 in prefixes of sentence do
    append chunk(word_1) to chunks
    for word_2 in prefixes of sentence[word_1.length:] do
        append chunk(word_1, word_2) to chunks
        for word_3 in prefixes of sentence[word_1.length + word_2.length:] do
            append chunk(word_1, word_2, word_3) to chunks
        end for
    end for
end for
return chunks
```

2.2 隐马尔科夫模型 (HMM)

2.2.1 形式化定义

已知观察序列 $Y = (Y_1, Y_2, \cdots, Y_n)$ 和模型 θ, 求解隐藏状态 $X = (X_1, X_2, \cdots, X_n)$, 最大化条件概率 $P(Y \mid X, \theta)$, 并根绝隐藏状态序列 X 解码得到分词结果.

2.2.2 建模方法

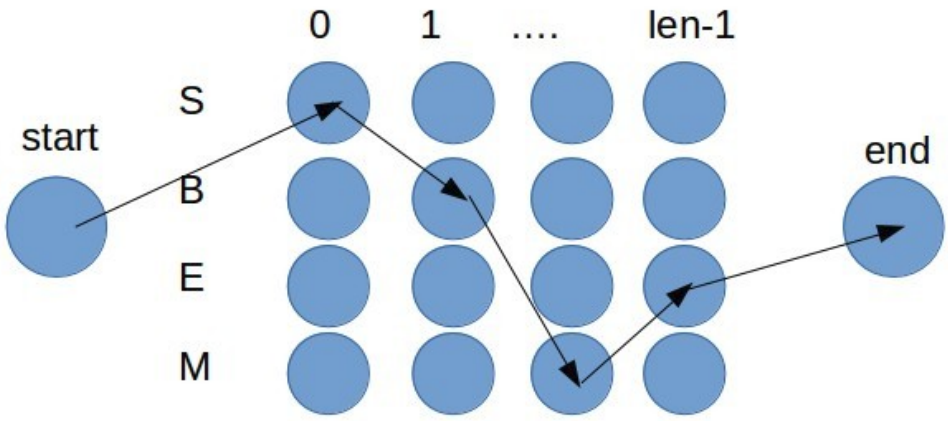
HMM 模型的参数是一五元组 $< StatusSet, ObservedSet, TransProbMatrix, EmitProbMatrix, InitStatus >$ 组成的, 其中:

- StatusSet: 状态值集合, 在分词任务中, 有四种状态 B,M,E,S, 分别表示起始字, 中间字, 结束字, 单字成词.
- ObservedSet: 观察值集合, 即在训练过程中或预定义的词典中的所有字
- TransProbMatrix: 转移概率矩阵, 马尔科夫链的特点是在 $T = i$ 时刻的状态 Status(i), 只和 $T = i$ 之前的 n 个状态有关,HMM 中假设 n=1, 因此在这里状态转移矩阵是一个 4*4 的矩阵, 分别对应前一个状态的 BMES 到下一个状态的 BMES 的转移概率.
- EmitProbMatrix: 发射概率矩阵, 发射概率是一个条件概率, 对观察值 i 和状态 j, 发射概率表示为 $P(Observed[i] \mid Status[j])$, 由观察值独立性假设, 有:

$$P(Observed[i], Status[j]) = P(Status[j]) * P(Observed[i] \mid Status[j])$$

- InitStatus: 初始状态分布, 即各个作为第一个字出现的概率分布, 显然, 状态 M 和 S 的初始概率为 0

HMM 模型求解中文分词问题的核心在于求解隐藏状态序列的 Viterbi[**?**] 算法, 该算法通过动态规划的方式求解状态转移图上概率最大的路径.



2.2.3 算法伪代码

训练 HMM 模型的过程虽然代码冗长, 但实际上只是简单的对训练语料的概率统计的过程, 因此在算法部分, 主要说明 Viterbi 算法的伪代码

Algorithm Viterbi(obs)

probs ← empty list of dictionary
path ← emtpy dictionary
for state in StatusSet **do**
  probs[0][state] ← $P(state \mid obs[0])$
  add state to path
end for
for $i = 1$ to $obs.length$ **do**
  new-path ← empty list of dictionary
  for state in StatusSet **do**
    from-state ← $argmax_{state}\ P(prev - state) * P(state \mid prev - state) * P(obs[i] \mid state)$
    new-path ← path[from-state] concat state
  end for
  update path with new-path
end for
return path with largest probs

3 实验结果

3.1 数据规模

- 训练数据:

  使用 ir.hit.edu.cn 提供的训练语料, 包含 7000 行, 185815 个词, 20941 个不重复的词, 此外 MMSEG 方法使用清华大学开发语料库 “诗词”,“成语”,“地名” 词库作为补充.

- 测试数据:

  使用 ir.hit.edu.cn 提供的开发语料, 其中语料 1 包含 1000 行, 26853 个词, 6889 个不重复的词; 语料 2 包含 201 行, 7402 个词, 2392 个不重复的词.

3.2 评价方法

评价方法使用 ir.hit.edu.cn 提供的 eval.py, 使用权重相等的准确率和召回率的综合性能指标 $F1 - measure$ 其中准确率 Precision:

$$Precision = \frac{\text{正确切分的词数}}{\text{切分出词的总数}}$$

召回率 Recall:

$$Recall = \frac{\text{正确切分的词数}}{\text{应切分出词的总数}}$$

评价指标:

$$f_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

3.3 准确率

3.3.1 MMSEG

- 开发语料 1:

  #recall 24573
  #gold 26853
  #predict 28650
  f: 0.884168423328

- 开发语料 2:

  #recall 6328
  #gold 7402
  #predict 8186
  f: 0.811906594817

3.3.2 HMM

- 开发语料 1:

  #recall 21520
  #gold 26853
  #predict 27029
  f: 0.798782524776

- 开发语料 2:

  #recall 5066
  #gold 7402
  #predict 6962
  f: 0.70537454748

参考文献

[1] https://baike.baidu.com/ 中文分词

[2] Chih-Hao Tsai *A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm* http://technology.chtsai.org/mmseg/

[3] P.Antal *Hidden Markov Models* antal@mit.bme.hu