面向中文电子病历的句法分析融合模型

蒋志鹏 [1, 2] 关 毅 [1]

摘 要 完全句法分析是自然语言处理 (Natural language processing, NLP) 中重要的结构化过程, 由于中文电子病历 (Chinese electronic medical record, CEMR) 句法标注语料匮乏, 目前还没有面向中文电子病历的完全句法分析研究. 本文针对中文电子病历模式化强的子语言特征, 首次以树片段形式化中文电子病历复用的模式, 提出了面向数据句法分析 (Data-oriented parsing, DOP) 和层次句法分析融合模型. 在树片段抽取阶段, 提出效率更高的标准树片段和局部树片段抽取算法, 分别解决了标准树片段的重复比对问题, 以及二次树核 (Quadratic tree kernel, QTK) 的效率低下问题, 获得了标准树片段集和局部树片段集. 基于上述两个树片段集, 提出词汇和词性混合匹配策略和最大化树片段组合算法改进面向数据句法分析模型, 缓解了无效树片段带来的噪声. 实验结果表明, 该融合模型能够有效改善中文电子病历句法分析效果, 基于少量标注语料 F1 值能够达到目前最高的 80.87 %, 并且在跨科室句法分析上超过 Stanford parser 和 Berkeley parser 2 % 以上.

关键词 中文电子病历, 完全句法分析, 面向数据句法分析, 层次句法分析

引用格式 蒋志鹏, 关毅. 面向中文电子病历的句法分析融合模型. 自动化学报, 2019, **45**(2): 276−288

DOI 10.16383/j.aas.2018.c170219

A Fusion Model for Chinese Electronic Medical Record Parsing

JIANG Zhi-Peng[1, 2] GUAN Yi[1]

Abstract Full parsing is an important structuring process of the natural language processing (NLP). However, its research on Chinese electronic medical record (CEMR) is currently a blank because of the lack of syntactical annotated corpus on CEMR. To make the best of the sub-language characteristic of strong pattern in CEMR, patterns reused is first formalized as tree fragment in CEMR, and a model integrating data-oriented parsing (DOP) and hierarchical parsing is proposed. In the extraction stage of tree fragments, we propose a more efficient standard tree fragment algorithm by solving repeated comparison of standard tree fragments, and a partial tree fragment extraction algorithm to substitute for the low-efficient quadratic tree kernel (QTK) algorithm to obtain a standard tree fragment set and a partial tree fragment set. Based on the two extracted tree fragment sets, a strategy matching word and part-of-speech (POS) synchronously and a maximal combination algorithm of tree fragments are proposed to improve DOP, and alleviate the noise caused by invalid tree fragments. Experimental results show that the fusion model based on DOP and hierarchical parsing can effectively improve the parsing effect for CEMR, and the F1 score reaches the highest 80.87 % based on a small number of annotated corpora, which is even 2 % higher than those of the two state-of-the-art parsers of Stanford and Berkeley in cross-department parsing.

Key words Chinese electronic medical record (CEMR), full parsing, data-oriented parsing (DOP), hierarchical parsing

Citation Jiang Zhi-Peng, Guan Yi. A fusion model for Chinese electronic medical record parsing. *Acta Automatica Sinica*, 2019, **45**(2): 276−288

电子病历是医务人员在医疗活动过程中使用医疗机构信息系统生成的文字、符号、图表、图形、数据、影像等数字化信息, 并能实现存储、管理、传输和重现的医疗记录[1−2]. 电子病历是一种极其宝贵的知识资源, 通过对电子病历进行自动分析, 可以为用户健康知识获取提供有力支持. 近几年, 电子病历的知识获取成为研究的热点问题[3−5], 其获取过程一般分为语言分析和信息抽取两个阶段进行, 词法分析和句法分析是主要的语言分析手段, 能够为信息抽取提供必要的条件. 例如, 电子病历的句子可能包含多种药物信息及不同的用药说明. 传统的规则表示仅在药名抽取上具有较高精度, 而用药频率、计量等重要信息的抽取需要结合上下文环境, 是规则表示无法处理的. 句法分析技术通过对句子进行结构化处理, 能够解析复杂的上下文环境, 为信息抽取提供丰富的上下文信息, 已经成为构建高精度信息抽取系统的重要组成部分.

中文电子病历 (Chinese electronic medical

收稿日期 2017-04-25 录用日期 2017-10-29
Manuscript received April 25, 2017; accepted October 29, 2017
国家自然科学基金 (71531007) 资助
Supported by National Natural Science Foundation of China (71531007)
本文责任编委 张民
Recommended by Associate Editor ZHANG Min
1. 哈尔滨工业大学语言技术中心网络智能研究室 哈尔滨 150001 2. 长安通信科技有限责任公司 北京 102209
1. Web Intelligence Laboratory, Language Technology Center, Harbin Institute of Technology, Harbin 150001 2. Changan Communication Technology Co., LTD, Beijing 102209

records, CEMR) 文本不同于一般的限定领域文本. 语料构建方面, CEMR 的标注工作要求兼顾医学和语言学的相关知识, 增加了标注者的学习成本, 整个标注过程更加费时费力. 文本差异方面, 除了词汇层面的差异外, CEMR 文本特有的书写方式加剧了与开放领域的文本差异. 具体表现为, 从文本构造上看, CEMR 是结构化数据和自由文本的结合, 自然语言处理的对象通常指的是自由文本. 根据 CEMR 词性标注准确率与未登录词率的相关性分析[6], CEMR 的自由文本可以分为叙述部分和罗列部分, 叙述部分多以长句的形式描述症状、检查等信息, 例如诊疗计划、诊断依据、病例特点、主诉等, 罗列部分多为医学术语及其修饰语的简单罗列, 例如临床初步诊断、临床确定诊断、门诊收治诊断、治疗效果等. 其中, 罗列部分充斥着开放领域语料鲜有的医学术语, 例如 "脑梗死" 在宾州中文树库 (Penn Chinese treebank, PCTB) 中从未出现过, 使得该部分的句法分析难以利用 PCTB 中学习的知识. CEMR 文本重复度高的特点导致叙述部分呈现强模式化现象, 例如 "伸舌"、"示齿" 这类缩略词构成的动宾短语在病例特点中频繁出现, 该类短文本多以标点符号分隔描述病情, 在 PCTB 中同样较少使用. CEMR 自由文本的这两部分特点使得直接应用 PCTB 训练的句法分析模型性能下降更为严重, Berkeley parser[7] 与 Stanford parser[8] 相比, 中文开放领域的最好结果下降了 30 % 左右.

本文将 CEMR 自由文本中重复出现的模式形式化为树片段, 树片段主要用于句法树重排序和面向数据句法分析 (Data-oriented parsing, DOP). 句法树重排序一般分为两步: 1) 由基本的句法分析器为每个句子产生一组候选句法树, 候选句法树的初始概率为最初排序的依据; 2) 根据句法树的额外的结构特征训练重排序模型, 对这组候选句法树重新排序, 通常使用树片段作为重排序的额外特征. 尽管重排序模型对句法分析精度有较大提升, 但是时空复杂度较高的缺点限制了其实际应用. DOP 技术首先由 Scha 在 1990 年提出, 之后由 Bod 逐步发展, 具体表达了如下假设: 人类对语言的领悟和创造依赖于以往具体的语言经验, 而不是依赖于抽象的语法规则[9]. 模型首先预设了一个具有带标短语结构树标注的语料库, 然后从这个语料库中抽取所有任意大小规模和复杂结构的片段. 其次, 通过对语料库中片段的组合操作来实现新输入的分析, 然后考虑输入的所有派生结果的概率总和的大小来选择最有可能性的分析结果[10].

在前期工作中, 我们对 CEMR 语料进行了词法统计分析[6], 其中逐点互信息的结果说明 CEMR 相比中文开放语料和英文 EMR 模式化程度更强, 这是由于 CEMR 的编写具有更强的目的性, 力求表述清晰、简洁, 导致重复使用相似的句法结构, 形成模式化的表述. 同时, CEMR 中包含不同的部分, 每个部分意图不同, 重复使用的短语结构也不尽相同, 这种模式化表述一般与陈述性语句混合使用. 以首次病程记录为例, 包括主诉、既往史、主观症状、客观检查、评估和诊断、诊疗计划六个部分, 其中频繁出现的重复结构如表 1 所示, 而传统的 PCFG 文法将句法树层数限制为 2, 难以直接表示这些模式, 例如 "伴心率失常室早"、"伴心率失常房颤"、"伴心动过速" 会形成两类重复模式, 即 "伴心率失常 + 疾病名" 和 "伴 + 疾病组", 这两类模式均不属于 PCFG 文法. DOP 希望抽取所有可能的树片段 (不限制层数), 能够更加直观、形象地表现 CEMR 中的模式. 我们以树片段替换的方式对初始的句法分析结果进行纠错, 类似于引入了句法树重排序中的结构特征, 但相比句法树重排序按模板抽取结构特征, DOP 的优势在于不限制树片段的形式, 能够保证其多样性, 并且在增加新语料时不需要重新训练模型. 尽管 DOP 的文法归纳过程不需要训练模型, 但是树片段的数量随树库规模呈指数级增长, 抽取树库中所有可能的树片段难以实现.

层次句法分析是一种快速的完全句法分析方法, 在前期工作中, 我们改进了层次句法分析模型[11], 并

表 1 重复模式样例

Table 1 Pattern samples repeated

文本类型	重复模式	举例
既往史	疾病史 + (时间)	(IP (NP 脑梗死病史) (QP 10 年))
	"承认/否认"+ 疾病史	(VP 否认 (NP 冠心病病史))
主观症状	名词 + 形容词	(IP 神志清楚)
	"伴"+ 症状 (组)	(VP 伴头晕)
客观检查	检查 + (": ") + 结果	(IP 钠离子: 129.3 mmol)
	无 + 疾病 (组)	(VP 无中枢性面瘫)

标注了小规模 CEMR 句法树库[12−13], 本文希望充分利用 CEMR 模式化强的特点, 进一步提升层次句法分析模型的精度. 本文的工作属于后处理融合, 基于改进的树片段的抽取算法, 将 DOP 中树片段的选择和替换操作融入层次句法分析过程中, 最终在 CEMR 上句法分析的 F1 值超过了目前最优的 Berkeley parser 和 Stanford parser. 模型整体架构如图 1 所示.

图 2 为面向数据句法分析与层次句法分析的融合示例, 输入为经过词性标注的 CEMR 句子 “对光反射存在, 双眼运动自如, 无面舌瘫”. 该句子依次经过词汇词性混合匹配、初选、筛选和组合过程, 获得了一个最优的树片段集合, 最终通过树片段替换的方式改进了层次句法分析结果. 图中虚线框部分为该实例中两个有效的替换.

1　相关工作

树核方法一般用于量化两棵句法树的相似程度, 并不能明确给出这两棵句法树的共有结构, Sangati 等[14] 在 2010 年首次将树核应用到树片段抽取中, Sangati 将树片段分为标准片段和局部片段, 抽取时对树库中所有树结点进行两两比较, 并提出基于树核的局部片段抽取算法, 该算法不只计算公共结点数量, 还保留了重复出现的最大公共树片段. 尽管 Sangati 能够从树库中抽取有效的树片段, 但是使用二次树核 (Quadratic tree kernel, QTK) 在树片段抽取时效率不高. 与 Sangati 相类似, Moschitti[15] 同样对树片段进行了划分, 并提出了基于快速树核 (Fast tree kernel, FTK) 的树结点匹配算法, 该算法的效率要高于 Sangati, 但是仅保留了树库的公共结点集而不是树片段. van Cranenburgh[16] 将 Moschitti 的快速树核算法改进为矩阵形式抽取树片段, 算法时间复杂度降为线性平均时间, 但整个抽取过程只适用于二叉树, 并且只抽取树库中的局部片段. 本文的树片段抽取以 Sangati 的工作为基础, 借鉴动态规划思想改进了标准树片段抽取算法, 并使用 Moschitti 的快速树核算法替换 Sangati 的二次树核算法, 提高了整个算法的执行效率.

中文方面 DOP 相关研究较少, 早期张玥杰等[10] 提出基于 DOP 框架的中文句法分析方法. 整个分析过程中, 句子需要依次经过词汇层与词性层的初选, 再从树片段库中获得与句子相匹配的片段组合形式, 利用 Kullback-Leibler 距离函数评估句子与初选结果的相似度, 分析结果为相似度大于阈值或相似度最高的片段组合. 本文的树片段处理过程相当于在张玥杰工作的基础上, 提出了一套适用于 CEMR 的树片段选择和替换方法. 由于我们的目的是通过树片段替换融合层次句法分析模型, 所以没有保留张玥杰的组合分析过程.

层次句法分析是一种高效的完全句法分析方法, 但是逐层组块分析导致错误累积问题严重, 在前期工作中, 我们提出了一种简单可行的错误预判及协同纠错算法[11], 每层组块分析时跟踪预判错误标注结果进入下一层, 利用两层预测分数相结合的方式协同纠错. 实验结果表明, 加入纠错方法后, 层次句法分析在保证解析速度的同时, 获得了与主流中文句法分析器相当的解析精度. 近几年, 由于单一模型的解析结果仍然存在局限性, 模型融合成为提高现有句法分析水平的主要途径. 模型融合一般指将不同句法分析器的结果融合成一个最终结果, 常见的融合方式包括将不同结果杂合成为一个结果[17−18], 或按照某些标准从多个结果中选择一个最优的结果[19].

2　标点符号分割与纠错算法

通过对 CEMR 标注语料进行统计分析[6], 我们发现标点符号在 CEMR 中所占比例高达 21.69 %,

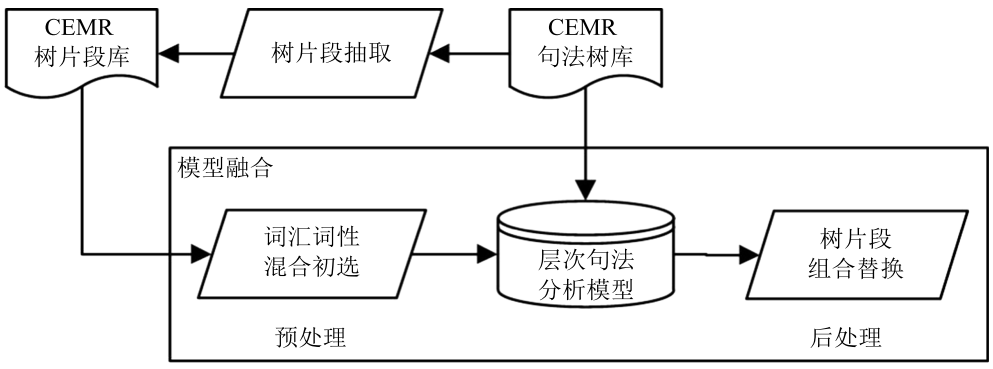

图 1　融合模型框架

Fig. 1　The framework of integrated model

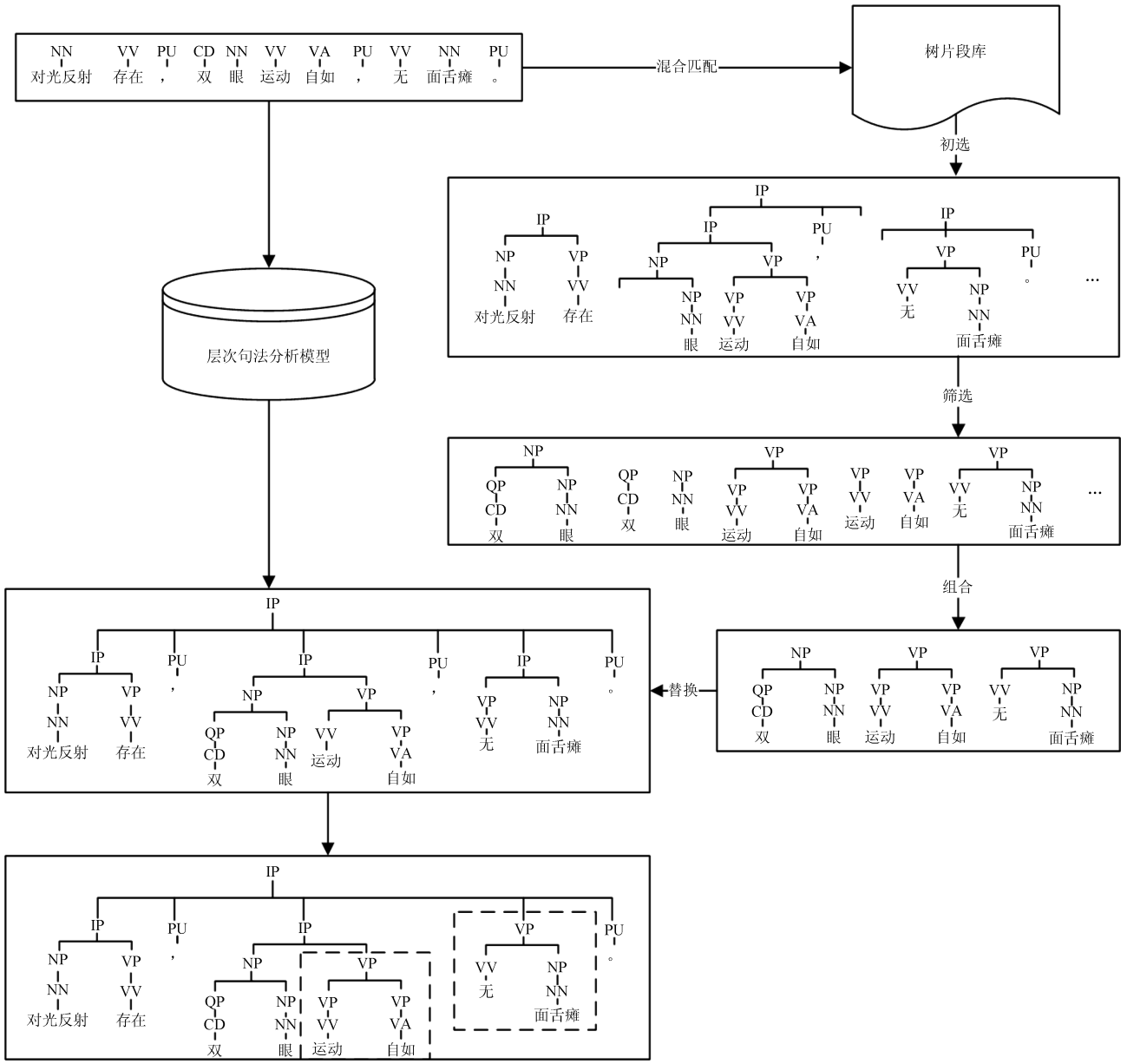

图 2 面向数据句法分析与层次句法分析融合示例

Fig. 2 The sample integrating DOP and hierarchical parsing

仅次于名词位居第二位, 并且高于开放领域文本 6.4% 之多. 由于层次句法分析框架是自底向上逐层序列化标注, CEMR 标点符号的频繁使用使得下列问题更为突出: 1) 句法树中高层标点符号成分较难确定; 2) 并列结构中标点符号成分难确定; 3) 逐层标注的错误累积导致标点对的内容无法成为单棵句法树. CEMR 中最常见的标点对为双引号和圆括号, 其内容经常被 CRF 模型分割成多棵句法树, 后续部分提及的标点对也默认为双引号和圆括号.

CEMR 中标点符号的使用同样有较强规律性, 例如, 经常出现在高层句法树中的标点符号多用于连接并列句法结构. 为缓解上述问题, 一方面, 本文围绕标点符号设计了辨识度更高的特征, 形成上下文词典, 以词典纠错的方式改进标点符号的标注效果. 另一方面, 使用标点对分割句子, 优先解析标点对的内部成分, 采用分割组合的方式进行句法分析.

结合层次句法分析模型[13] 在调试语料上的错误分析结果, 我们设计了上下文词典对标点符号进行纠错, 关键词为标点符号, 词典项如表 2 所示. 其中, 第 1 行为通用项 $\langle father, lfather, rfather\rangle$, father 表示父结点, lfather 表示父结点的左兄弟结点, rfather 为父结点的右兄弟结点. 第 2 行 aword 为相邻词, 相邻词的选取与 Collins 的头词类似, 目的是对子树边界位置的标点符号进

行消歧, 如果父结点是 NP, 则 aword 为左相邻词, 否则 aword 为右相邻词, 当 aword 为空时 $\langle lgfather, rgfather, lbword, rbword\rangle$ 生效, lgfather 和 rgfather 分别为祖先结点的左兄弟和右兄弟结点, lbword 为 lfather 的最右子结点, rbword 为 rfather 的最左子结点. 第 3 行和第 4 行的词典项与句法树高度相关, 即句子边界有助于高层标点符号纠错, 固定搭配有助于低层标点符号纠错. 当标点符号所在层数大于 3 时, $\langle lbbegin, rbend\rangle$ 判断当前位置是否为句首或句尾, 若不是则直接比对 aword 项.

表 2　上下文词典项概括

Table 2　Summary of elements of context dictionary

使用条件	词典项
/	$\langle father, lfather, rfather\rangle$
aword = NULL	$\langle lgfather, rgfather, lbword, rbword\rangle$
height > 3	$\langle lbbegin, rbend\rangle$
height < 4	$\langle aword\rangle$

标点符号的上下文词典纠错算法能够与多层协同纠错算法[11] 很好地配合使用. 当歧义项为标点符号时, 优先检索上下文词典, 选择匹配成功的候选结果作为最优解, 否则进入多层协同纠错算法消歧. 在检索上下文词典时, 为了减少词典中人工错误带来的干扰, 我们规定只有当前条件对应的元组完全匹配时, 才算匹配成功.

由于闭合的标点对必然会成为一个句法树, 所以省去了训练模型识别独立语块的过程, 在进行句法分析时, 如果存在标点对, 则优先解析标点对的内容, 再将标点对的句法树与其余部分重组成新的输入, 进行新一轮解析. 引入标点符号分割和纠错算法的层次句法分析流程如图 3 所示, 其中 “CRF 序列化标注” 和 “多层协同纠错” 是前期构建的基础模型 CLP[11].

3　树片段抽取

由于树片段不限制层数和结构, 导致其数量随语料规模呈指数级增长, 抽取时间远长于统计机器学习模型的训练时间, 当需要从更大规模语料中抽取树片段时, 抽取时间将成为其实际应用的瓶颈问题, 所以在树片段抽取上的改进工作主要围绕抽取效率展开.

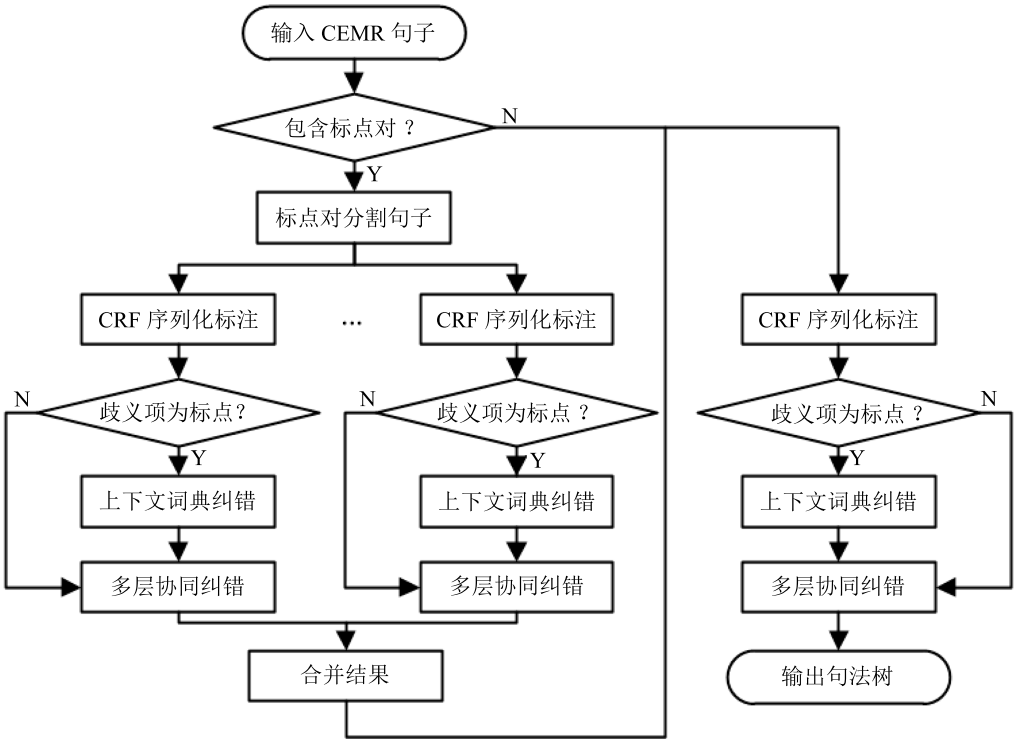

图 3　引入标点符号分割和纠错的句法分析流程

Fig. 3　The parsing process with segmentation and error correction for punctuation

3.1 树片段相关定义

参照 Sangati 等[14] 对树片段的划分方式, 首先给出标准树片段和局部树片段的相关定义, 然后为两类树片段分别设计算法抽取公共树片段, 形成两个树片段库.

定义 1 (标准树片段). 标准树片段 stf 是句法树 T 的连通子图, 并且 stf 中任意结点的子结点或为空, 或与 T 中对应结点的子结点相同.

定义 2 (局部树片段). 局部树片段 ptf 是句法树 T 的任意连通子图.

定义 3 (公共树片段). 公共树片段 ctf 是在句法树库中至少出现两次的标准树片段或局部树片段.

定义 4 (公共叶结点). 公共叶结点 ct 属于公共树片段 ctf, 并且 ct 的子结点或为空, 或不属于 ctf.

定义 5 (公共非叶结点). 公共非叶结点 cnt 的子结点不为空, 并且 cnt 及其子结点全部属于公共树片段 ctf.

从上述定义可以看出, 标准树片段相当于局部树片段的特殊形式, 所以其抽取和匹配的耗时更少, 为了对比两类树片段对模型融合的贡献, 本文分别抽取了两个树片段库. 公共树片段可以较好地表现 CEMR 中重复出现的模式, 下文提及的标准树片段和局部树片段都默认为公共树片段.

以 CEMR 中的句法树为例, 图 4 (a) 为 CEMR 句法树, 图 4 (b) 和图 4 (c) 分别表示从该句法树中抽取的部分标准树片段和局部树片段. 共现模式是 CEMR 中的高频模式, 图 4 中实线框内的标准树片段能够体现 “心率失常” 和 “室早” 这两个病症的共现模式, 而通过将 “室早” 泛化为名词词性, 形成虚线框内的局部树片段, 则能够表现 “心率失常” 与其他病症的共现模式, 例如常见的 “心动过速”、“房颤” 等. 由此可见, CEMR 中的共现模式与树片段能够很好地吻合, 所以本文将其形式化为树片段, 用于改善句法分析模型在 CEMR 上的解析效果.

3.2 标准树片段抽取

Sangati 等[14] 抽取标准树片段的过程由两部分组成, 第 1 部分遍历句法树库中每一对树结点, 算法时间复杂度为 $\mathrm{O}(n^2m^2)$, 其中 n 为句法树的数量, m 为单棵句法树的最大结点数, 第 2 部分抽取父子结点均相同的片段, 最坏情况下, 算法时间复杂度为 $\mathrm{O}(m^2)$, 最终算法时间复杂度为 $\mathrm{O}(n^2m^4)$. 为避免重复比对问题, 我们利用动态规划思想改进了 Sangati 的抽取算法, 如算法 1 所示.

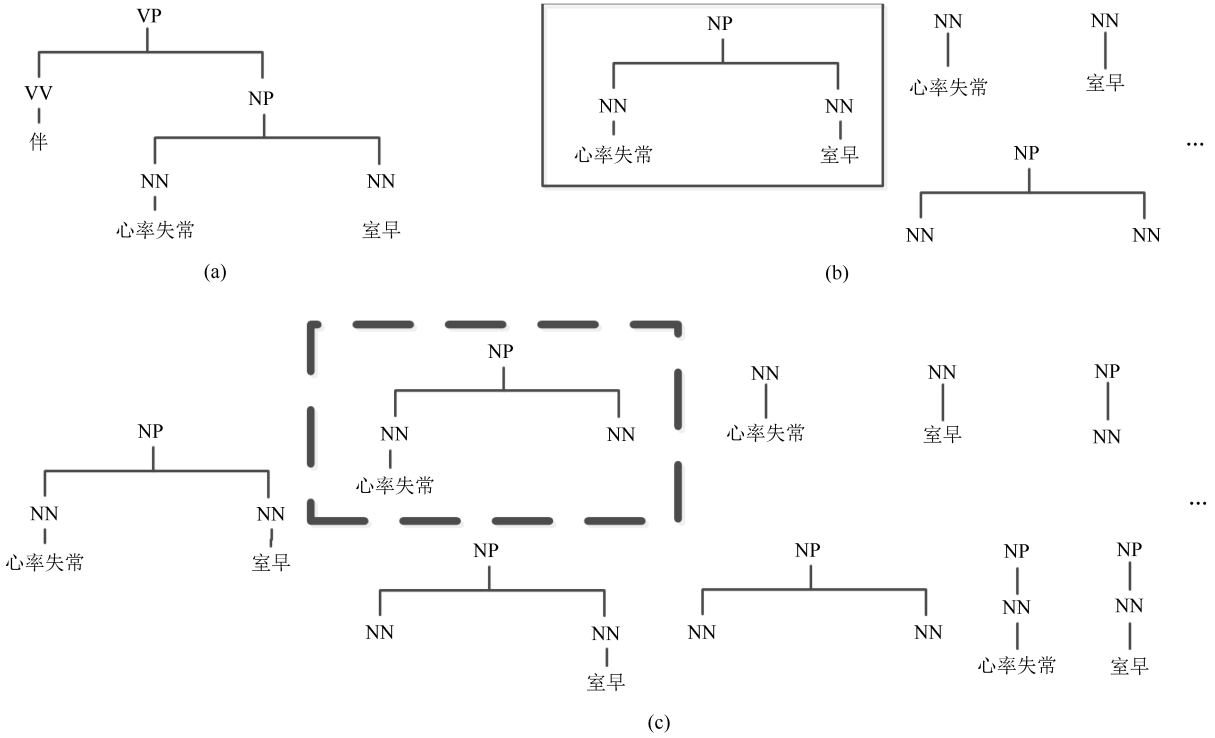

图 4 句法树及其片段样例

Fig. 4 Examples of a parsing tree and its fragments

算法 1. 标准树片段抽取算法

EX-TREEBANK-STF(T)
1. $N \leftarrow$ SIZE(T)
2. for $i \leftarrow 0$ to N
3. do for $j \leftarrow i+1$ to N
4. do for each $node_1 \in t_i$
5. do for each $node_2 \in t_j$
6. do if $node_1 \notin visited$ and $node_2 \notin visited$
7. then EX-TREE-STF($node_1$, $node_2, visited$)
8. add $visited$ to stf
9. return stf
10. EX-TREE-STF ($node_1, node_2, visited$)
11. $equal \leftarrow$ TRUE
12. $m \leftarrow$ CHILD-SIZE($node_1$)
13. $n \leftarrow$ CHILD-SIZE($node_2$)
14. if $node_1 = node_2$
15. then if $m = n$ and $m > 0$
16. then for $i \leftarrow 0$ to m
17. do if !EX-TREE-STF($node_1.child[i]$, $node_2.child[i], visited$)
18. then $equal \leftarrow$ FALSE
19. else $equal \leftarrow$ FALSE
20. if $equal =$ FALSE
21. then $equal \leftarrow$ TRUE
22. $node_1$ is a common terminal
23. else $node_1$ is a common non-terminal
24. add $node_1$ and $node_2$ to $visited$
25. else $equal \leftarrow$ FALSE
26. return $equal$

不同于 Sangati 的抽取算法, 本文的标准树片段抽取算法围绕公共非叶结点展开, 其中, SIZE 函数用于计算树库中句法树的数量, CHILD-SIZE 函数返回子结点数目, 算法第 1～5 行是树库结点对的遍历过程, 第 6 行的变量 $visited$ 用于保存已访问的公共非叶结点对. 在调用 EX-TREE-STF 函数抽取标准树片段之前需要查找 $visited$, $visited$ 中存在的公共非叶结点对不再遍历. 算法第 11～26 行用于识别公共非叶结点, 具体地, 在遍历结点对时, 当两个结点相同并且子结点数也相同时, 递归调用 EX-TREE-STF 算法遍历所有子结点, 如果所有子结点相同, 则该结点对为公共非叶结点对; 否则, 当两个结点相同, 但其子结点不同或为空时, 该结点对则为公共叶结点对. 由于不需要重复遍历公共非叶结点, 使得 EX-TREE-STF 算法时间复杂度降至 O(d), d 为单个结点的最大子结点数, 整个算法的时间复杂度降为 O(n^2m^2d).

3.3 局部树片段抽取

与标准树片段抽取算法类似, Sangati 等[14] 在抽取局部树片段时同样需要先遍历句法树库中每一对树结点, 再抽取局部树片段. 不同之处在于, 两个树结点可能共享多个公共子结点序列, 形成多种公共局部树片段的组合. 为抽取所有可能的公共局部树片段, Sangati 等提出了基于二次树核的结点映射算法, 该算法类似于最长公共子序列查找, 需要遍历每个子结点对, 其时间复杂度为 O(d^4), d 为单个结点的最大子结点数, 整个算法的时间复杂度为 O($n^2m^2d^4$).

事实上, 在抽取公共树片段时, 如果两个父结点的产生式不同, 其子结点没有必要再进行比较. Moschitti[15] 提出了基于快速树核的结点映射算法, 只考虑父结点产生式相同的结点匹配情况. 受该思想启发, 本文利用 Moschitti 的快速树核替换二次树核算法, 获取树结点映射集, 算法 2 是改进后的局部树片段抽取算法.

算法 2. 局部树片段抽取算法

EX-TREEBANK-PTF(T)
1. $N \leftarrow$ SIZE(T)
2. for $i \leftarrow 0$ to N
3. do for $j \leftarrow i+1$ to N
4. do $nodes_i \leftarrow$ SORT($tree_i$)
5. $nodes_j \leftarrow$ SORT($tree_j$)
6. $n_i \leftarrow$ POP($nodes_i$)
7. $n_j \leftarrow$ POP($nodes_j$)
8. while $n_i \neq \emptyset$ and $n_j \neq \emptyset$
9. do if PROD(n_j) > PROD(n_j)
10. then $n_j \leftarrow$ POP($nodes_j$)
11. else if PROD(n_i) < PROD(n_j)
12. then $n_j \leftarrow$ POP($nodes_i$)
13. else while PROD(n_i) = PROD(n_j)
14. do while PROD(n_i) = PROD(n_j)
15. do add $\langle n_i; n_j \rangle$ to $maps$
16. $n_j \leftarrow$ NEXT($nodes_j$)
17. $n_i \leftarrow$ POP($nodes_i$)
18. $fragset \leftarrow$ EX-TREE-PTF($n_i, n_j, maps$)
19. return $fragset$

算法 2 中, SORT 函数将结点的产生式按字母顺序排序, POP 函数返回有序表中首个结点并将其弹出, PROD 函数返回结点所在的整个产生式, NEXT 函数返回有序表中当前结点的下一个结点. 第 1～3 行为树库的遍历过程, 该过程与算法 1 相同. 第 4～17 行为基于快速树核的结点映射算法, 该算法首先对树的结点按字母顺序排序, 然后逐个比较产生式相同的结点, 对所有公共结点进行标记, 与标准树片段抽取不同, 这里不需要区分公共叶结点和非叶结点. 第 18 行递归调用 EX-TREE-PTF 算法获取 n_i 和 n_j 的公共局部树片段集合, 该算法与 Sangati 等[14] 提出的算法 3 相同, 功能是对结点映射组 maps 进行合并, 最终加入到整个局部树片段集合中.

由于产生式排序过程在进行结点映射之前仅执行一次, 时间复杂度为 $\mathrm{O}(m\log m)$, 结点映射过程最坏情况下执行 m^2 次, 而 EX-TREE-PTF 算法对整棵树也只需遍历一次, 所以改进后的局部树片段抽取算法时间复杂度为 $\mathrm{O}(n^2m^4)$.

4 DOP 与层次句法分析的融合模型

前文将 CEMR 重复出现的模式形式化为树片段, 并分别抽取标准树片段和局部树片段, 形成了两个树片段库. 与树片段关联最紧密的句法分析框架就是 DOP 框架, 然而单独使用 DOP 模型进行句法分析的效果并不理想. 模型融合是提高模型解析精度的有效方法, 将 DOP 与层次句法分析模型进行融合, 既合理利用了 CEMR 模式化强的特点, 又避免了单独使用 DOP 模型精度不高的问题.

目前常用的模型融合方式包括特征融合和后处理融合, 特征融合通过融合多个模型的特征, 对其进行共同训练, 联合解码. 后处理融合则更倾向于多个模型结果的重组或选择. 由于特征融合并不适用于 DOP 模型, 所以我们选择后处理融合进行扩展, 提出了面向 CEMR 的 DOP 与层次句法分析融合模型.

4.1 预处理

预处理阶段的工作分为两部分: 1) 树片段的匹配; 2) 基于匹配结果的树片段初选. 张玥杰等[10] 在匹配树片段时先匹配词汇再匹配词性, 只有词汇完全匹配的树片段才能直接成为最终结果. 与张玥杰等的做法不同, 我们没有将词汇和词性单独处理, 而是采用词汇和词性混合的方式匹配树片段, 不局限于词汇完全匹配的约束条件, 充分发挥词性在树片段匹配过程中的泛化作用. 树片段初选主要选择两类树片段, 一类是能够与输入句子完全匹配的树片段, 另一类树片段与输入句子部分匹配, 且存在不包含边界结点的独立子树, 这里边界结点特指处于树片段最左端或最右端的结点. 其中, 第一类树片段能够直接成为输出结果, 第二类树片段是后续树片段组合替换的基础.

初选的第二类树片段可能是不规范或不合法的树片段, 图 5 虚线框内为不规范的树片段, 该类树片段可能包含多棵子树, 将在第 4.2 节中进一步筛选. 另外, 仅依靠词汇词性匹配会抽取到不合法树片段, 如图 5 实线框所示, 由于只匹配到 “ “ (双引号) 脑卒中、(顿号) ”, 所以形成了错误的 NP, 为了避免抽取这类树片段, 增加了 “存在不包含边界结点的独立子树” 的约束, 在该约束条件下, 用于替换的树片段必须从初选树片段内部获得 (不含边界结点), 该约束条件虽然减少了初选树片段的数量, 但也解决了后续步骤中不合法树片段带来的错误替换问题.

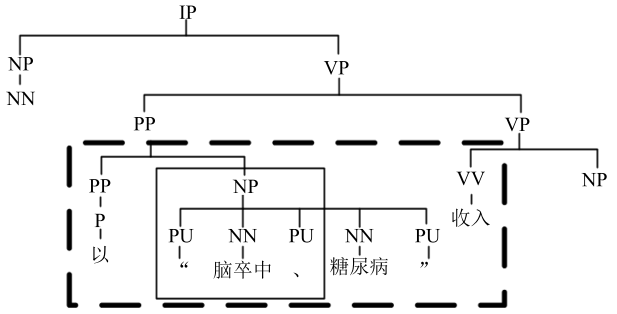

图 5 初选树片段样例

Fig. 5 The sample of selected tree fragment

受连续公共子序列查找算法启发, 本文提出了词汇词性混合初选算法, 如算法 3 所示.

算法 3. 词汇词性混合初选算法

WORD-POS-COMBINED-SELECT(ls_1, ls_2)

1. $len_1 \leftarrow$ SIZE(ls_1)
2. $len_2 \leftarrow$ SIZE(ls_2)
3. for $i \leftarrow 0$ to len_1
4. do for $j \leftarrow 0$ to len_2
5. if !UPDATE($ls_1[i], ls_2[j]$) or $i \in borders$
6. then continue
7. $x \leftarrow i+1$
8. $y \leftarrow j+1$
9. while $x < len_1$ and $y < len_2$
10. do if !UPDATE($ls_1[x], ls_2[y]$)
11. then if $x-i>2$ or $x-i=2$ and $i=0$ and $j=0$ or $x-i=2$ and $x=len_1$ and $y=len_2$
12. then add $\langle i, x-1\rangle$ and $\langle j, y-1\rangle$ to $borders$
13. break
14. else $x \leftarrow x+1$
15. $y \leftarrow y+1$
16. if $x=len_1$ or $y=len_2$
17. then if $x-i>2$ or $x-i=2$ and $i=0$ and $j=0$ or $x-i=2$ and $x=len_1$ and $y=len_2$
18. then add $\langle i, x-1\rangle$ and $\langle j, y-1\rangle$ to $borders$
19. return $borders$

算法 3 中, ls_1 为树片段的叶结点及其父结点集合, ls_2 为输入句子的词汇和词性集合. UPDATE 函数用于词汇和词性的混合匹配, 同时将局部树片段扩展为标准树片段, 具体分为 4 种情况: 1) 词汇与叶结点匹配, 直接返回 TRUE; 2) 词性与叶结点匹配, 将词汇追加为该叶结点的子结点, 返回 TRUE; 3) 词性与父结点匹配, 将词性替换为父结点, 返回 TRUE; 4) 词汇、词性均不匹配, 返回 FALSE. 算法第 11 行和第 17 行是第二类树片段的约束条件, 由于在后续工作中, 用于替换的句法树必须在树片

段内部且独立, 所以我们抽取的第二类树片段至少包含三个结点, 或处于句子边界位置且包含两个结点. 该算法最后返回所有匹配的边界对, 除了树片段与句子完全匹配的情况, 当局部树片段边界包含句子边界, 并且局部树片段为独立句法树时, 该局部树片段也能够直接输出, 否则该句子进入层次句法分析模型解析阶段. 算法单次执行的时间复杂度为 $\mathrm{O}(m \times w)$, 由于需要遍历整个树片段库, 所以最终时间复杂度为 $\mathrm{O}(n \times m \times w)$, 其中, n 为树片段个数, m 和 w 分别为树片段叶结点数以及输入句子的词数.

4.2　后处理融合

后处理阶段是基于预处理初选的树片段集和层次句法分析结果, 进行树片段的筛选、组合与替换, 进一步改进句法分析结果. 为了不影响层次句法分析结果中其他树片段, 用于替换的树片段必须从初选树片段内部获得, 并且是高度大于 1 的标准树片段, 即不考虑仅包含词汇和词性的树片段. 初选树片段可能包含多棵子树, 也可能与其他树片段发生交叉, 所以需要先对初选树片段进行筛选. 筛选后的树片段集合是不含边界结点的全部标准树片段, 以图 6 为例, 图 6 (a) 和图 6 (b) 分别为初选树片段与筛选后的树片段集合.

张玥杰等采用先拼接再计算相似度的方式组合树片段[10], 在实验中我们也尝试了他们的最左优先原则拼接树片段, 但该方式会生成更多无效句法树, 带来更多噪声. 我们的最终目的是以树片段替换方式改进已有句法树, 所以组合树片段时可以绕开拼接过程, 转变为最大化叶结点替换数目. 基于动态规划思想, 本文提出最大树片段组合算法, 如算法 4 所示.

算法 4. 最大树片段组合算法

MAX-SUBSTITUTE-SPAN($end, borders, spans$)

1. $i \leftarrow 0$
2. if $end > 0$
3. then $i \leftarrow$ MAX-SUBSTITUTE-SPAN($end-1$, $borders, spans$)
4. while $i <$ SIZE($borders$)
5. do if $borders[i].second > end$
6. then if SPAN(max) $= 0$
7. then add $borders[i]$ to max
8. break
9. else if $borders[i].second = end$
10. then add $borders[i]$ to now
11. if SPAN(now) $>$ SPAN(max)
12. then $max \leftarrow now$
13. $i \leftarrow i + 1$
14. if $spans[end]$ = null or spans[end] $< max$
15. then $spans[end] \leftarrow max$
16. return i

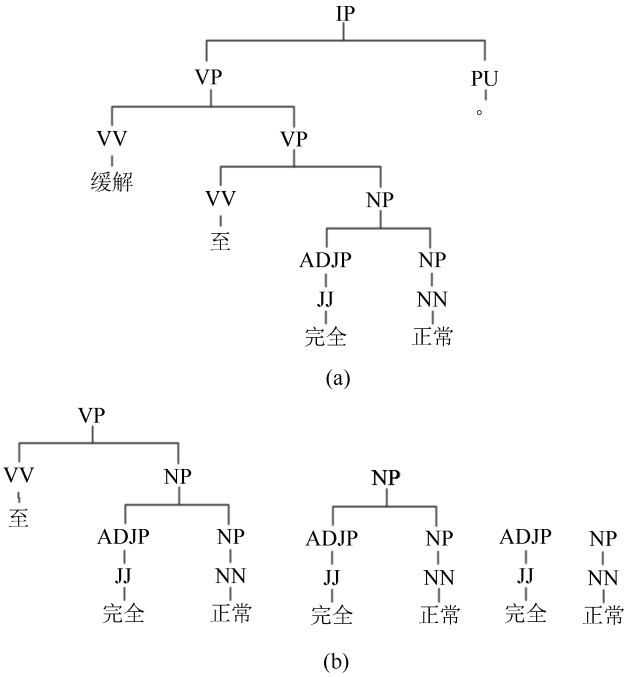

图 6　初选树片段与筛选树片段集合

Fig. 6　The selected tree fragment and its filtered tree fragments

算法 4 不需要考虑拼接时边界结点的匹配情况, 而是直接获得叶结点数最大时的树片段集合. 算法三个参数分别为: 当前树片段的右边界 (end), 按右边界从小到大排序的树片段边界集合 ($borders$), 所有右边界对应的最大树片段集合 ($spans$). now 和 max 用于保存当前树片段集合及当前右边界对应的最大树片段集合, SPAN 函数为 now 或 max 包含的叶结点数. 当两个树片段集合叶结点数相同时, 遵循高度优先原则, 选择层数更高的树片段集合, 因为叶结点相同时层数高的树片段往往会包含层数低的树片段. 最大树片段组合算法的时间复杂度为 $\mathrm{O}(p^2)$, p 为筛选后的树片段个数.

基于最大树片段组合算法获得树片段集合, 下一步进行树片段替换工作. 树片段替换依然遵循高度优先原则, 首先利用广度优先搜索遍历层次句法分析模型输出的句法树中所有非终结符, 当非终结符的叶结点与树片段的叶结点完全匹配时, 用树片段替换以该非终结符为根结点的子树, 直到集合内所有树片段替换完成后, 输出新的句法分析结果. 树片段替换算法的时间复杂度为 $\mathrm{O}(p \times s \times m)$, 其中 p 为筛选后的树片段个数, s 为层次句法分析模型输出的句法树中非终结符个数, m 为单个树片段的最大叶结点数. 另外, 在实验中发现, 先对最大树片段集合按叶结点数进行排序, 再选择前 5 个树片段替换, 比直接使用集合中的全部树片段替换效果更好.

5 实验结果与分析

在前期的工作中, 我们以 PCTB 标注规范为基础, 通过迭代的方式不断调整规范, 结合 CEMR 中特有的标注样例对 PCTB 规范进行细化、扩充、删减, 首次提出了适用于中文电子病历的分词、词性和句法标注规范, 并获得了较高的标注一致性和准确率, 其词性和句法标注体系与 PCTB 完全相同. 本文实验所用语料为前期标注的 CEMR 句法树库[12−13]. 该树库为神经内科和普通外科的出院小结和首次病程记录, 共 138 份 CEMR, 包含 2 555 棵句法树, 语料统计信息如表 3 所示. 语料采用的 5 折交叉验证的方式构造, 每次随机平分为 5 份, 4 份作为训练集, 1 份作为测试集, 同时从训练集随机抽取 1/5 作为调试集.

表 3 CEMR 句法树库统计信息

Table 3 Corpus statistics of CEMR treebank

科室	份数	句子数	词数
神经内科	70	1 486	28 189
普通外科	68	1 069	19 235
共计	138	2 555	47 424

5.1 树片段抽取实验

为了对比句法分析模型, 每次交叉验证从上述训练语料中抽取出现 2 次以上的标准树片段和局部树片段, 形成了两个树片段库, 并改进两类树片段的抽取算法, 从而提高抽取效率. 树片段抽取结果如表 4 所示, 实验环境为 64 位 Ubuntu 系统, 1.2 GHz 的 CPU. 从抽取速度可以看出, 利用 FTK 算法改进后, 抽取效率明显提高, 抽取速度相当于 QTK 的 4 倍左右. 表 4 中给出树片段种类数而非树片段数, 是因为树片段种类的多少与文本模式化强弱相关, 直接影响 DOP 的性能, 通常来说, 文本模式化越强树片段的种类应该越少. 从表中可以看出, 局部树片段的种类约为句子数的 19 倍, 明显多于标准树片段, 所以相比标准树片段抽取速度也更慢.

表 4 树片段抽取结果

Table 4 Results of fragment extraction

树片段类型	句法树数目	树片段种类	抽取速度 (秒/句)
局部树片段	958	18 267	7.38 (FTK)/27 (QTK)
标准树片段	958	4 514	4.21

5.2 DOP 与层次句法分析模型融合实验

本文的对比模型包括引入标点符号纠错和分割算法的层次句法分析模型 (CLPU)、引入 DOP 的融合模型, 以及开放领域最优的 Stanford parser 和 Berkeley parser. 首先在单一科室 CEMR 上进行训练测试, 即神经内科 CEMR 按 5 折划分, 随机抽取 1/5 作为测试集, 其余作为训练集. 实验结果如表 5 所示, 在前期工作中, CLP 和 Berkeley parser 在开放领域测试集 (TCT) 上句法分析精度相近[11], 这里利用上下文词典进一步修正了 CLP 结果中的标点符号错误, 并引入标点对分割组合策略, 得到了 CLPU 模型, 句法分析 F1 值达到 78.23 %. 受语料规模影响, Berkeley parser 产生了 52 个空输出, 导致 F1 值降为 78.17 %, 落后于 CLPU 结果. 为解决 Berkeley parser 的空输出问题, 本文对产生空输出的句子使用另一个 PTCB 训练的 Berkeley parser 解析, 最后 Berkeley parser (CEMR+PCTB) 的 F1 值能够提高到 79.8 %, 由于其他模型不涉及空输出问题, 所以不需要额外训练模型解析.

在模型融合方面, 本文首先对比了引入标准树片段 (SDOP) 和局部树片段 (PDOP) 的模型效果. 可以看出, 引入标准树片段和局部树片段后 F1 值均

表 5 神经内科 CEMR 句法分析结果

Table 5 Parsing results on CEMR of neurology department

模型	词性标注准确率 (%)	句法分析			解析速度 (秒/句)
		召回率 (%)	准确率 (%)	F1 值 (%)	
Berkeley parser	83.82	85.09	72.29	78.17	0.2
CLPU	89.39	78.88	77.58	78.23	0.4
CLPU + SDOP	89.78	80.16	78.26	79.2	0.4
Berkeley parser (CEMR + PCTB)	92.57	82.18	77.55	79.8	0.3
Stanford parser	93.76	80.1	80.01	80.35	0.1
CLPU + PDOP	89.9	80.52	80.52	80.52	0.9
CLPU + PDOP (TOP 5)	89.92	81.15	80.59	80.87	0.9

有较大的提升, 最多提高了 2.64 %. 虽然标准树片段产生错误替换的可能性更小, 但也限制了树片段的多样性, 导致最终与局部树片段的 F1 值相差 1.32 %. 另一方面, 通过错误分析发现, 单个树片段匹配的叶结点越多, 错误替换的概率越小, 所以对筛选组合后的树片段集合先排序再替换, 实验证明保留前 5 个树片段进行替换的 F1 值最高, 达到 80.87 %, 获得了目前 CEMR 上句法分析的最优结果.

为了对比各模型在不同领域中的性能, 本文在 PCTB 上进行了句法分析实验. 为公平起见, PCTB 上采用与 CEMR 相同的实验设置, 仅从 PCTB 中随机抽取 1 207 句训练, 277 句测试, 实验结果如表 6 所示. 根据已公布的结果, 当训练语料充足时 (训练测试语料的比例达到 50 倍), Stanford parser 和 Berkeley parser 在 PCTB 上句法分析 F1 值能够达到 84 % 左右, 而在表 6 中, 当 PCTB 的训练语料规模与 CEMR 同样小时, Berkeley parser 的 F1 值最高只有 64 % 左右, Stanford parser 更下降到 61.16 %, 相比之下, CEMR 在仅需要少量标注语料的情况下, F1 值却能够达到 80 % 左右, 在一定程度上也证明了其模式化强, 句法清晰的特点. 进一步对比各模型的实验结果, DOP 在 PCTB 上无论是解析速度还是 F1 的增幅上都有较大下降. 究其原因, 解析速度变慢是由于从 PCTB 中抽取的树片段库相比 CEMR 规模要大, 其局部树片段种类达到句子数的 43 倍之多, 所以在树片段的遍历匹配时需要消耗大量时间. 如此多的树片段种类说明 PCTB 并不像 CEMR 模式化那么强, 导致 DOP 带来的增幅仅为 0.42 %. 同时, 标点符号在 PCTB 所占比例的下降, 使得 CLPU 的 F1 值只有 63.6 %, 最终引入 DOP 后仍落后 Berkeley parser 0.36 %. 尽管本文提出的融合模型在 PCTB 上的表现并不理想, 但是也从另一个角度验证了其更加适用于模式化强的子语言文本的假设.

在跨科室句法分析实验中, 本文选择已标注的两个科室上进行交叉测试, 实验设置与表 5 相同. 分析结果如表所示. 从表 7 可以看出, 由于不同科室间的词汇存在较大差异, 使得词性标注时未登录词率升高, 各模型词性标注准确率均有不同程度的下降, 最终句法分析 F1 值只能达到 69 % 左右. 其中受未登录词率影响最大的是 Stanford parser, 特别是在普通外科训练、神经内科测试时, Stanford parser 词性标注准确率仅为 75.23 %, 导致最后句法分析 F1 值只有 57.85 %. 相比之下, 本文提出的 CLPU 及融合模型表现出色, 尽管在词性标注准确率上也有所下降, 但是标点符号的高使用率、文本模式化强的特点是不同科室的共性, 所以两个模型的跨科室性能相比 Stanfrod parser 和 Berkeley parser 有较大的优势, 融合 DOP 后 F1 值能够超过最好结果 2 % 以上. 但是, 在进行错误分析时, 我们发现词汇差异使 DOP 错误替换的几率增加, 导致引入 DOP 后词性标注准确率反而有较大下降, 所以相比 CLPU 的提升只有 1 % 左右.

最后, 综合比较各模型在单一科室 CEMR, PCTB 以及跨科室 CEMR 的句法分析性能. Stanford parser 的解析速度最快, 但是也最不稳定, 容易受未登录词影响, 当未登录词率较高时, 句法分析 F1 值有较大幅度的下降. Berkeley parser 的性能中规中矩, 但是对语料规模依赖较强, 当标注语料充足时, 句法分析性能更好, 但是考虑到 CEMR 的文本特殊性, 除了数据来源受限, 还要求标注者同时具备语言学和医学相关知识, 所以构建大规模 CEMR 语料库更加困难. 本文提出的 CLPU 和 DOP 的融合模型在 CEMR 上的句法分析性能要优于 Stanford parser 和 Berkeley parser, 特别是跨科室句法分析时表现更为突出, 证明引入树片段对于模式化强的子语言分析是行之有效的方法. 进一步对比 DOP 中各算法的时间复杂度, 由于初选过程需要遍历整个树片段库, 而树片段库的量级远大于其他参数, 占用了大量的解析时间, 导致融合模型的句法分析速度相比 Stanford parser 和 Berkeley parser 差距较大. 但是, 由于目前还没有在树片段的存储和查找上做任何优化工作, 所以未来在解析速度上仍有上升

表 6　PCTB 句法分析结果

Table 6　Parsing results on PCTB

模型	词性标注准确率 (%)	句法分析			解析速度 (秒/句)
		召回率 (%)	准确率 (%)	F1 值 (%)	
Stanford parser	86.05	62.94	59.48	61.16	0.1
CLPU	89.98	65.59	61.73	63.6	0.4
CLPU + SDOP (TOP 5)	87.19	65.66	62.46	64.02	1.6
Berkeley parser	82.34	66.67	62.38	64.46	0.2

表 7 跨科室 CEMR 句法分析结果

Table 7 Parsing results on cross-department CEMR

	词性标注准确率 (%)	句法分析			解析速度 (秒/句)
		召回率 (%)	准确率 (%)	F1 值 (%)	
源科室: 普通外科					
目标科室: 神经内科					
Berkeley parser	83.85	66.74	66.31	64.52	0.2
Stanford parser	84.69	67.69	65.51	66.58	0.1
CLPU	88.96	69.79	66.04	67.86	0.3
CLPU + SDOP (TOP 5)	79.53	70.6	67.92	69.23	1.2
源科室: 普通外科					
目标科室: 神经内科					
Stanford parser	75.23	58.19	57.51	57.85	0.1
Berkeley parser	82.17	67.64	64.63	66.11	0.2
CLPU	89.58	70.78	66.65	68.65	0.3
CLPU + SDOP (TOP 5)	83.75	71.4	67.93	69.62	0.9

空间, 例如遍历树片段库时引入剪枝策略, 通过对树片段进行编码提高匹配效率等.

6 结论

本文针对 CEMR 模式化强的特点, 提出以树片段形式化 CEMR 复用的模式. 首先描述了树片段的相关概念, 举例说明标准树片段和局部树片段的区别, 然后针对标准树片段的重复比对问题, 提出更加高效的标准树片段抽取算法, 又利用快速树核算法替换二次树核算法, 降低了局部树片段抽取算法的时间复杂度, 并通过实验对比了改进前后局部树片段抽取速度的变化, 最终获得了标准树片段库和局部树片段库. 基于抽取的树片段库, 提出了 DOP 与层次句法分析的融合模型, 该模型属于后处理融合, 以模型融合的方式避免了单独使用 DOP 模型精度较低的问题. 在预处理阶段, 提出了词汇词性混合匹配的初选策略, 充分发挥词性在匹配时的泛化作用. 在后处理阶段, 一方面, 从初选树片段内部筛选标准树片段, 减少了错误树片段的替换, 另一方面, 提出了最大化树片段组合算法, 简化 DOP 组合分析过程, 缓解了无效树片段带来的噪声. 实验结果表明, DOP 与层次句法分析的融合模型能够利用 CEMR 模式化强的特点, 有效改善句法分析效果.

CEMR 模式化强的特点为 DOP 创造了更大的发展空间. 未来的工作, 一方面可以探索更加深入的模型融合方法, 进一步提高 DOP 在融合模型中的比重; 另一方面, CEMR 高昂的句法标注成本使得无监督句法分析方法更具实用价值, 如何充分发挥 DOP 在无监督句法分析上的优势, 提出适用于 CEMR 的无监督 DOP 模型是值得研究的问题.

References

1 Ministry of Health of the People's Republic of China. The basic specifications of electronic medical records (trial). [Online], available: http://www.gov.cn/gzdt/2010-03/04/content_1547431.htm, March 4, 2010
(中华人民共和国卫生部. 电子病历基本规范 (试行). [Online], available: http://www.gov.cn/gzdt/2010-03/04/content_1547431.htm, March 4, 2010)

2 Yang Jin-Feng, Yu Qiu-Bin, Guan Yi, Jiang Zhi-Peng. An overview of research on electronic medical record oriented named entity recognition and entity relation extraction. *Acta Automatica Sinica*, 2014, **40**(8): 1537−1562
(杨锦锋, 于秋滨, 关毅, 蒋志鹏. 电子病历命名实体识别和实体关系抽取研究综述. 自动化学报, 2014, **40**(8): 1537−1562)

3 Jiang M, Huang Y, Fan J W, Tang B Z, Denny J C, Xu H. Parsing clinical text: how good are the state-of-the-art parsers? *BMC Medical Informatics and Decision Making*, 2015, **15**(S1): Article No. S2

4 Stubbs A, Kotfila C, Xu H, Uzuner Ö. Identifying risk factors for heart disease over time: overview of 2014 i2b2/UTHealth shared task Track 2. *Journal of Biomedical Informatics*, 2015, **58 Suppl**: S67−S77

5 Chen Y K, Lask T A, Mei Q Z, Chen Q X, Moon S, Wang J Q, Nguyen K, Dawodu T, Cohen T, Denny J C, Xu H. An active learning-enabled annotation system for clinical named entity recognition. *BMC Medical Informatics and Decision Making*, 2017, **17**(S2): Article No. 82

6 Jiang Zhi-Peng, Zhao Fang-Fang, Guan Yi, Yang Jin-Feng. Research on Chinese electronic medical record oriented lexical corpus annotation. *Chinese High Technology Letters*, 2014, **24**(6): 609−615
(蒋志鹏, 赵芳芳, 关毅, 杨锦锋. 面向中文电子病历的词法语料标注研究. 高技术通讯, 2014, **24**(6): 609−615)

7 Petrov S, Klein D. Improved inference for unlexicalized parsing. In: Proceedings of the 2007 Human Language Technologies: the Annual Conference of the North American Chapter of the Association for Computational Linguistics. New York, USA: ACL, 2007. 404−411

8 Klein D, Manning C D. Fast exact inference with a factored model for natural language parsing. In: Proceedings of the 2003 Advances in Neural Information Processing Systems. Massachusetts, USA: MIT Press, 2003. 3−10

9 Bod R. A computational model of language performance: data oriented parsing. In: Proceedings of the 14th Conference on Computational Linguistics: Volume 3. New York, USA: ACL, 1992. 855−859

10 Zhang Yue-Jie, Zhu Jing-Bo, Zhang Yue, Yao Tian-Shun. Implementing Chinese parsing based on DOP technique. *Journal of Chinese Information Processing*, 2000, **14**(1): 13−21
(张玥杰, 朱靖波, 张跃, 姚天顺. 基于 DOP 的汉语句法分析技术. 中文信息学报, 2000, **14**(1): 13−21)

11 Jiang Zhi-Peng, Guan Yi, Dong Xi-Shuang. A Chinese hierarchical parsing approach based on multi-layer collaborative correction. *Journal of Chinese Information Processing*, 2014, **28**(4): 29−36
(蒋志鹏, 关毅, 董喜双. 基于多层协同纠错的中文层次句法分析. 中文信息学报, 2014, **28**(4): 29−36)

12 Jiang Z P, Zhao F F, Guan Y. Developing a linguistically annotated corpus of Chinese electronic medical record. In: Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Belfast, UK: IEEE, 2014. 307−310

13 Jiang Z P, Dai X, Guan Y, Zhao F F. A lexical and syntactic analysis system for Chinese electronic medical record. *International Journal of u- and e- Service, Science and Technology*, 2016, **9**(9): 305−318

14 Sangati F, Zuidema W, Bod R. Efficiently extract recurring tree fragments from large treebanks. In: Proceedings of the 7th International Conference on Language Resources and Evaluation. New York, USA: ELRA, 2010. 219−226

15 Moschitti A. Making tree kernels practical for natural language learning. In: Proceedings of the 2010 European Chapter of the Association for Computational Linguistics. Trento, Italy: EACL, 2006. 24

16 van Cranenburgh A. Extraction of phrase-structure fragments with a linear average time tree-kernel. *Computational Linguistics in the Netherlands Journal*, 2014, **4**: 3−16

17 Yang L E, Sun M S, Cheng Y, Zhang J C, Liu Z H, Luan H B, Liu Y. Neural parse combination. *Journal of Computer Science and Technology*, 2017, **32**(4): 749−757

18 Choe D K, McClosky D, Charniak E. Syntactic parse fusion. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: ACL, 2015. 1360−1366

19 Narayan S, Cohen S B. Diversity in spectral learning for natural language parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: ACL, 2015. 1868−1878

蒋志鹏 哈尔滨工业大学博士研究生. 主要研究方向为自然语言处理, 电子病历文本分析.
E-mail: hit.jiang@hotmail.com
(**JIANG Zhi-Peng** Ph. D. candidate at Harbin Institute of Technology. His research interest covers natural language processing and text analysis on electronic medical records.)

关 毅 哈尔滨工业大学教授. 主要研究方向为智能信息检索, 网络挖掘, 自然语言处理, 认知语言学. 本文通信作者.
E-mail: guanyi@hit.edu.cn
(**GUAN Yi** Professor at Harbin Institute of Technology. His research interest covers intelligent information retrieval, web mining, natural language processing, and cognitive linguistics. Corresponding author of this paper.)