基于扩展 Viterbi 路径的概率 Earley 算法

韩习武[1,2] Roland Hausser[2]
(黑龙江大学计算机科学技术学院 哈尔滨 150080)[1]
(德国埃尔兰根纽伦堡大学计算语言学研究室 埃尔兰根 91054)[2]

摘 要 概率 Earley 句法分析算法采用 Viterbi 路径构造输入序列的句法树，由于文法限制，存在空树问题。提出了扩展启始状态、省略未覆盖句首和补充未覆盖子树等方法来对 Viterbi 路径进行扩展，解决了绝大多数空树问题，并有效提高了 Earley 算法的整体性能。

关键词 Viterbi 路径，句法分析，概率 Earley 算法
中图法分类号 TP181 **文献标识码** A

Probabilistic Earley Algorithm Based on Extended Viterbi Path

HAN Xi-wu[1,2] Roland Hausser[2]
(School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China)[1]
(CLUE at University of Erlangen-Nuremberg, Erlangen 91054, Germany)[2]

Abstract Probabilistic Earley algorithm applies Viterbi path to construct parse trees for input sequences, but the grammar limits often result in many empty trees. This paper exploited optional start states, new sequential beginnings and more subtrees to extend the traditional Viterbi path, removed almost all empty trees, and improved the general parsing performance of Earley algorithm.

Keywords Viterbi path, Parsing, Probabilistic earley algorithm

句法分析一般被认为是从句子的单词串得到句法结构的过程。句法分析的研究是自然语言处理领域的一个重要组成部分，是该领域其他更加实用的技术(如自动构建词汇知识库[1]、信息抽取和机器翻译[2]等等)的一种重要预处理手段。随着互联网汉语文本的爆炸式增长，汉语信息处理的重要性与日俱增。然而，汉语句法分析技术，由于起步较晚、树库资源缺乏等原因，不如英语句法分析技术发展得迅速。目前的汉语句法分析技术还不能满足各种中文信息处理系统的基本要求，所以该项研究既意义重大又任重道远。

我们以概率 Earley 算法[3]为框架，构建了上下文无关的汉语句法分析系统，并通过扩展启始状态、省略语法未覆盖句首以及补充可能子树等手段，实现了对传统 Viterbi 路径的有效改善，解决了句法分析中的大部分空树问题，大幅度提高了 Viterbi 路径句法分析的性能。本文第 2 节将简要介绍概率 Earley 算法和 Viterbi 路径；第 3 节详细阐述我们对经典 Viterbi路径的扩展方法；第 4 节给出了在清华树库 TCT[4]上的句法分析实验结果；最后总结全文。

1 概率 Earley 句法分析

Earley 算法是一种基于线图的、兼具自顶向下和自底向上优势的句法分析方法，它应用推导进行规则搜索，然后以规约排除无关规则，可以在 $O(n^3)$ 时间内处理任何上下文无关文法[5]。概率 Earley 算法能够同时计算输入字符串的前向(forward)概率和内部(inner)概率[3]，前向概率可筛选输入串合法前缀，内部概率可排序局部子树。规则搜索完成之后，经典概率 Earley 算法一般以 Viterbi 路径输出句法树。

1.1 概率 Earley 算法

非概率 Earley 算法是应用给定的上下文无关产生式集合构建输入串最左推导过程的动态规划方法。对于某一产生式规则 $X \rightarrow \lambda\mu$，点规则 $X \rightarrow \lambda \cdot \mu$ 表示 λ 分析已经完成，而 μ 尚待分析[1]。算法针对输入串中的每一个单词间隔生成有序状态集合，每一个状态 $(X \rightarrow \lambda \cdot \mu, [i,k])$ 是一个二元组，其中包括：

- 当前正在匹配的产生式规则 $(X \rightarrow \lambda\mu)$；
- 该规则中的当前匹配位置，表示为 $\cdot$ 或 k；
- 该规则在输入串中的最初启始位置 i。

涉及输入串位置 k 的所有状态构成集合 $S(k)$，句法分析的运行从只包含最顶层启始规则状态 $(\rightarrow \cdot S, [0,0])$ 的状态集 $S(0)$ 开始，然后反复迭代以下 3 个步骤。

- 预测：对于状态集中 $S(k)$ 的每一个形如 $(X \rightarrow \lambda \cdot Y\mu, [i,$

到稿日期：2010-02-09 返修日期：2010-04-29 本文受国家自然科学基金(60773069,60873169)资助。

韩习武(1972－)，副教授，主要研究方向为自然语言处理，E-mail：hxw@hlju.edu.cn；**Roland Hausser**(1946－)，教授，主要研究方向为自然语言处理。

[1]按照惯例，λ，μ 和 γ 等拉丁字母代表由终结符和/或非终结符组成的任何字符串，X 和 Y 等大写英文字母代表单个非终结符，而 a 和 b 等小写英文字母代表单个终结符。

$k]$)的状态，为所有左部为 Y 的产生式规则生成状态($Y\rightarrow\cdot v,[k,k]$)，并添加至集合 $S(k)$；

·扫描：对于状态集中 $S(k)$ 的每一个形如($X\rightarrow\lambda\cdot a\mu,[i,k]$)的状态，如果输入串中当前符号为 a，则添加状态($X\rightarrow\lambda a\cdot\mu,[i,k+1]$)至集合 $S(k+1)$；

·完成：对于状态集中 $S(k)$ 的每一个形如($X\rightarrow v\cdot,[i,k]$)的状态，查找状态集 $S(i)$ 中所有形如($Y\rightarrow\lambda\cdot X\mu,[j,i]$)的状态，生成新状态($Y\rightarrow\lambda X\cdot\mu,[j,k]$)，并添加至集合 $S(k)$。

对于每一个输入符号和相应的状态集合，Earley 算法穷尽执行以上 3 个步骤，直到没有任何新的状态生成为止。如果到达终结状态，输入序列的句法树可以通过回溯一条预先存储的路径来实现。

在此基础上，概率 Earley 算法利用概率上下文无关文法为每一个 Earley 状态计算两个概率值，即前缀概率或前向概率、串概率或内部概率。

·前向概率 $\alpha(X\rightarrow\lambda\cdot\mu,[i,k])$ 是对长度为 k，且以状态($X\rightarrow\lambda\cdot\mu,[i,k]$)结尾的所有路径的概率累积；

·内部概率 $\gamma(X\rightarrow\lambda\cdot\mu,[i,k])$ 是对长度为 $k-i$，且以状态($X\rightarrow\cdot\lambda\mu,[i,i]$)开始，以状态($X\rightarrow\lambda\cdot\mu,[i,k]$)结尾的所有路径的概率累积。

概率 Earley 状态($X\rightarrow\lambda\cdot\mu,[i,k],[\alpha,\gamma]$)为三元组，增加了前向概率 α 和内部概率 γ 的存储。句法分析的过程与非概率算法基本相同，只是在前述 3 个迭代步骤中分别增加了前向概率和内部概率的计算。计算方式如下：

·预测：对于状态($X\rightarrow\lambda\cdot Z\mu,[i,k],[\alpha,\gamma]$)$\Rightarrow$($Y\rightarrow\cdot v,[k,k],[\alpha',\gamma']$)

$$\alpha'=\alpha'+\alpha*R(Z\overset{*}{\Rightarrow}_{L}Y)P(Y\rightarrow v) \quad (1)^{2}$$

$$\gamma'=P(Y\rightarrow v) \quad (2)$$

·扫描：对于状态($X\rightarrow\lambda\cdot a\mu,[i,k],[\alpha,\gamma]$)$\Rightarrow$($X\rightarrow\lambda a\cdot\mu,[i,k+1],[\alpha',\gamma']$)

$$\alpha'=\alpha,\gamma'=\gamma$$

·完成：对于状态($X\rightarrow v\cdot,[i,k],[\alpha'',\gamma'']$)和($Y\rightarrow\lambda\cdot X\mu,[j,i],[\alpha,\gamma]$)$\Rightarrow$($Y\rightarrow\lambda X\cdot\mu,[j,k],[\alpha',\gamma']$)

$$\alpha'=\alpha'+\alpha*\gamma'' \quad (3)$$

$$\gamma'=\gamma'+\gamma*\gamma'' \quad (4)$$

1.2 Viterbi 路径

对于文法 G 的字符串 x 的 Viterbi 分析是在 x 的所有可能推导中赋予 x 最大概率的推导之一。在马尔可夫信息源或隐马尔可夫模型中，Viterbi 路径是指某一事件观察序列背后隐含状态的最大似然序列。对概率 Earley 算法稍加修改即可在句法分析时构建 Viterbi 路径，然后沿该路径回溯即可得到输入句子的句法树或其它表示形式的分析结果，修改方法如下：

·为每一个 Earley 状态增加 Viterbi 概率的计算 V；

·Viterbi 概率的传播方式与内部概率相同，但在完成步骤中原概率累积改为求最大概率，意即以上式(4)由式(5)代替；

$$\gamma'=\mathrm{Max}_{\gamma'}(\gamma*\gamma'') \quad (5)$$

·记录最大概率状态($Y\rightarrow\lambda X\cdot\mu,[j,k]$)的同位置前驱状态($X\rightarrow v\cdot,[i,k]$)为 Viterbi 路径前驱。

如果到达最终状态($\rightarrow S\cdot,[0,l]$)，其中 l 是输入句子的单词数目，即输入序列长度，就可以应用递归回溯子程序 Back-trace 构造对应于该 Viterbi 路径的句法树。该子程序以某 Earley 状态($X\rightarrow\lambda\cdot\mu,[i,k]$)为参数输入，输出 i 到 k 之间的句子片段的 Viterbi 句法分析树。

Back-trace($X\rightarrow\lambda\cdot\mu,[i,k]$)：

1. 如果 $\lambda=\varepsilon$，返回根节点为 X，且没有子节点的句法树；

2. 否则，如果 λ 以终结符 a 结尾，令 $\lambda'a=\lambda$，然后递归调用本程序如下

$T=$Back-trace($X\rightarrow\lambda'\cdot a\mu,[i,k-1]$)

连接标识为 a 叶子节点为子树 T 的根节点的最右子节点，并返回 T；

3. 否则，如果 λ 以非终结符 Y 结尾，令 $\lambda'Y=\lambda$，沿 Viterbi 路径查找当前状态的前驱状态($Y\rightarrow v\cdot,[j,k]$)，然后分别递归调用本程序两次如下

$T=$Back-trace($X\rightarrow\lambda'\cdot Y\mu,[i,j]$)

以及

$T'=$Back-trace($Y\rightarrow v\cdot,[j,k]$)

连接子树 T' 为子树 T 的同根节点右子树，并返回 T。

2 扩展 Viterbi 路径

Earley 句法分析是积极算法，它同时存储完成的和未完成的状态或线图边。如果某一程度为 l 的输入序列的启始状态($\rightarrow\cdot S,[0,0]$)能够到达某一最终状态($\rightarrow\cdot S,[0,l]$)，我们说 Earley 句法分析的识别模块成功结束。然后，相应句法树可以通过调用($\rightarrow S\cdot,[0,l]$)来构造。但是，由于给定文法的不完全覆盖，往往不能成功到达任何最终状态($\rightarrow\cdot S,[0,l]$)，就会导致很多只有根节点 S 的空句法树。为了解决这一问题，我们对经典 Viterbi 路径进行了如下扩展。

2.1 扩展启始状态

根据我们对部分语料的分析，造成空树的重要原因之一是：句法树库上的文法未能够覆盖输入句子结尾的某些终结符，或已覆盖输入句子的所有终结符但同时导致推导序列结尾出现过多非终结符，但当前句子却不能匹配这些非终结符，即句法分析中可能存在扫描不足和/或扫描过剩的现象。

针对这种现象，我们提出扩展 Viterbi 启始状态的方法。因为 Earley 线图同时存储了未完成状态，所以从某一未完成状态出发构造非完整子树。当句法分析没有到达最终状态($\rightarrow\cdot S,[0,l]$)时，我们将假设 Earley 算法的启始状态是($X\rightarrow\cdot\lambda\mu,[0,0]$)，且终止状态是($X\rightarrow\lambda\cdot\mu,[0,m]$)，其中 μ 可以为空，m 是输入序列的最大已扫描子串 λ 的长度，且 $m\leqslant l$。这时就可以调用子程序 Back-trace($X\rightarrow\lambda\cdot\mu,[0,m]$)，回溯至状态($X\rightarrow\cdot\lambda\mu,[0,0]$)时，停止并输出根节点为 X 的关于 λ 的部分子树。

2.2 省略未覆盖句首

造成空树的另一个原因是：句法树库上的文法集合中没有以输入句子的最左终结符为右部最左可能终结符号的产生式规则，或尽管存在该规则但却不能匹配规则右部第一个首符号之后的某些终结符，并因此无法得到任何完成状态，即句

[2] $R(Z)\overset{*}{\Rightarrow}_{L}Y$ 是左角关系矩阵中 RL 的一个非零值，表示 Y 出现在非终结符 Z 的某些产生式规则右部最左边的可能性[3]。

法分析中可能存在扫描不能启动和/或扫描不完全的现象。

针对第二种现象，我们递归省略输入句子的当前未覆盖首终结符，直到能够输出一棵分析子树为止。我们以(→·S,[i,i]),0<i<l,为参数，调用子程序 Back-trace；如果没有得到分析子树，则令 i=i+1，并继续调用 Back-trace(→·S,[i,i])。

2.3 补充未覆盖子树

以上两种方法能够有效地解决 Earley 句法分析的空树问题，但只能为输入句子输出部分子树，不能够覆盖整个句子，因此补充未覆盖子树可以进一步提高句法分析的整体性能。

对于第一种情况下省略的句子结尾的子序列，如果该子序列长度等于一，则将其作为已输出子树的右兄弟节点连接到当前根节点；如果该子序列的长度大于等于二，我们可以再次调用已经扩展了的概率 Earley 句法分析器，得到未覆盖子树，并作为已输出子树的右兄弟子树连接到当前根节点；如果反复调用 Earley 句法分析器仍不能得到任何子树，则将省略子序列按原有顺序作为已输出子树的右兄弟节点逐一连接到当前根节点。

对于第二种情况下省略的句子首部的子序列，在理论上来说是无法构造任何子树的，因为当前文法不包含任何以该子序列为右部左角的产生式规则。然而，根据我们对具体语料的观察分析，发现造成这种情况的大部分原因是句首存在两个以上的 wLB 标记，这种标记的单词一般为左括号(、【、「、〖、〔,左引号‘、“,左书名号《、〈等等。如果输入句子没有语法错误，则当前已输出子树的右侧也必然省略了相应的 wRB 标记。

在句法树库中，大部分这种情况的外层节点都与内部子树的顶层节点相同，因此我们在句法分析时就以同时进行左右扩展的方法补充这种未覆盖子树，并标记完整句法树的根节点为已输出子树的根节点。

3 句法分析实验

基于以上理论，我们在中文信息学会句法评测 CIPS-ParsEval-2009 的实验语料上进行了一系列句法分析实验。

3.1 语料与任务

此次评测句法分析任务的语料由清华大学模式识别国家重点实验室自然语言处理组改编自清华树库 TCT，该句法树库的标记体系共包括 70 个词性标记和 16 个句法标记[4]。

用于评测的数据集合共包括约 45 万词汇，约 4 万句子，其中新闻类占 36%、百科学术类占 56%、文学类占 8%。以上数据分割为训练集和测试集，训练集合大约包括 35 万词汇，3.3 万句法树，句子平均长度为 11 个单词；测试集合大约包括 10 万词汇，7000 汉语句子，句子平均长度 12 个单词(此处没有计算那些只包括一个子节点的平凡句法树)。

评测任务的输入是经过正确切分和词性标注处理的汉语句子；输出是相应句子的完整句法结构树，各个节点以 np，vp，ap 等句法标记标注；评测指标基于括号匹配，包括标记准确率(Precision)、标记召回率(Recall)和 F1 值。

3.2 实验结果与分析

在句法分析实验中，我们直接从评测语料训练集中抽取文法规则，去除那些只出现一次的规则，未做任何概率平滑。实验包括：原始 Viterbi 路径句法分析，扩展启始状态，省略未覆盖句首和补充未覆盖子树。原始 Viterbi 路径句法分析在测试集合上共产生 684 棵空树，接近整个测试集的 10%，扩展启始状态的方法去除了大约 91%的空树，省略未覆盖句首的方法又去除了 8%，剩余的空树大多是由测试数据的文法错误造成。各部分实验的句法分析性能如表 1 所列。

表 1 句法分析实验结果

括号匹配(%)	Precision	Recall	F1
原始 Viterbi 路径	74.35	73.21	73.78
扩展启始状态	77.06	75.78	76.41
省略未覆盖句首	77.17	76.20	76.68
补充未覆盖子树	77.95	77.47	77.71

扩展启始状态的方法使得句法分析的 F1 值提高了 2.63%，是最有效的 Viterbi 扩展；省略未覆盖句首的方法使得 F1 值提高了 0.27%；补充未覆盖子树的方法使得 F1 值提高了 1.03%。总体上来看，我们对原始 Viterbi 路径的扩展使得概率 Earley 句法分析的整体性能提高了 3.93%。

结束语 本文针对概率 Earley 算法在句法分析中产生的空树问题，提出了扩展 Viterbi 路径的改进方法，包括扩展启始状态、省略未覆盖句首和补充未覆盖子树。在清华树库 TCT CIPS-ParsEval-2009 句法评测任务上的实验结果表明：扩展的 Viterbi 路径排除了绝大多数空树问题，有效地提高了 Earley 句法分析的整体性能。此后的研究工作将主要侧重于如何为已构造子树和未覆盖成分选择最优根节点。

参 考 文 献

[1] Han Xi-wu, Zhao Tie-jun, et al. Cross-lingual Syntactic Subcategorization Analysis Based on Chinese and English Sentence Pairs [C] // Proceedings of the First International Conference on Global Interoperability for Language Resources. 2008: 97-104

[2] Liu Yang, Liu Qun, Lin Shou-xun. Tree-to-String Alignment Template for Statistical Machine Translation [C] // Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL. 2006: 609-616

[3] Stolcke A. Parsing Algorithm that Computes Prefix Probabilities [J]. Computational Linguistics, 1995, 21(2): 1-36

[4] Zhou Qiang. Build a Large-Scale Syntactically Annotated Chinese Corpus [C] // Proceedings of 6th International Conference of Text, Speech and Dialogue (TSD2003). 2003: 106-113

[5] Earley J. *An efficient context-free parsing algorithm* [J]. Communications of the Association for Computing Machinery, 1970, 13(2): 94-102

(上接第 176 页)

[19] Henzinger T A, Manna Z, Pnueli A. Timed transition systems [A] // Real-time: Theory in practice [C]. Volume 600 of Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 1992: 226-251

[20] Alur R. Timed automata [A] // Computer aided verification [C]. Volume 1633 of Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 1999: 8-22

[21] Behrmann G, David A, Larsen K G. A tutorial on UPPAAL [A] // Formal methods for the design of real-time systems [C]. Volume 3185 of Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 2004: 200-236