基于动作建模的中文依存句法分析*

段湘煜，赵军，徐波

中国科学院自动化研究所模式识别国家重点实验室 北京 100080

摘要：决策式依存句法分析，也就是基于分析动作的句法分析方法，常常被认为是一种高效的分析算法，但是它的性能稍低于一些更复杂的句法分析模型。本文将决策式句法分析同产生式、判别式句法分析这些复杂模型做了比较，试验数据采用宾州中文树库。结果显示，对于中文依存句法分析，决策式句法分析在性能上好于产生式和判别式句法分析。更进一步，我们观察到决策式句法分析是一种贪婪的算法，它在每个分析步骤只挑选最有可能的分析动作而丢失了对整句话依存分析的全局视角。基于此，我们提出了两种模型用来对句法分析动作进行建模以避免原决策式依存分析方法的贪婪性。试验结果显示，基于动作建模的依存分析模型在性能上好于原决策式依存分析方法，同时保持了较低的时间复杂度。

关键词：中文依存句法分析；决策式依存分析；动作建模

中图分类号：TP391　　文献标识码：A

Chinese Dependency Parsing Based on Action Modeling

Xiangyu Duan, Jun Zhao, Bo Xu

Institute of Automation, Chinese Academy of Sciences, Beijing, 100080

Abstract: Action-based dependency parsing, also known as deterministic dependency parsing, has often been regarded as an efficient parsing algorithm while its parsing accuracy is a little lower than the best results reported by more complex parsing models. In this paper, we compare action-based dependency parsers with complex parsing methods such as generative and discriminative parsers on the standard data set of Penn Chinese Treebank. The results show that, for Chinese dependency parsing, action-based parsers outperform generative and discriminative parsers. Furthermore, we propose two kinds of models for the modeling of parsing actions in action-based Chinese dependency parsing. We take the original action-based dependency parsers as baseline systems. Results show that our two models perform better than the baseline systems while maintaining the same time complexity, and our best result improves much over baseline.

Key Words: Chinese dependency parsing; deterministic dependency parsing; parsing action modeling

1　介绍

句法分析是自然语言处理（NLP）的重要任务之一。这项研究的主流是统计的方法，主要有产生式句法分析模型和判别式句法分析模型。这些模型应用不同的优化目标来训练模型参数，并使用诸如动态规划等一些非决策式的方法来计算各候选树的概率，具有最大概率的句法树被最后输出。如果应用重排序（reranking），则输出前 n 个概率最大的树，随后用排序算法对这些树进行重排序。

这些方法都取得了较好的性能，但是由于要计算各个候选树的整体概率，时间复杂度很高。与之对比，决策式句法分析是一种高效的句法分析算法，它将句法分析动作一步步作用于输入句子之上，时间复杂度被降低到线性或二次方于句子长度。最先决策式方法被用于依存句法分析 [1][2][3]。后来，Sagae 和 Lavie [4] 以及 Wang 等 [5] 将决策式分析方法应用于短语结构句法分析。

在标准数据集宾州英文树库上，决策式句法分析器显示了在时间效率上的巨大优势，但分析准确率要低于当前最好的英文分析性能。在本文中，对于中文依存句法分析，我们分别

*收稿日期：2007-5-18，定稿日期：2007-6-30

基金项目：国家自然科学基金项目（60673042）；国家高技术研究发展计划（2006AA01Z144）；北京市自然科学基金（4052027，4073043）

作者简介：段湘煜（1976－），男，博士，自然语言处理。

采用了 Yamada 和 Matsumoto [2] 的算法以及 Nivre 和 Scholz [3] 的算法，并将这两个决策式依存句法分析算法同产生式句法分析器及判别式句法分析器做了比较，试验数据采用宾州中文树库 5.0 版 [6] 。结果显示，决策式依存分析器要明显优于产生式句法分析器和判别式句法分析器。

更进一步，我们发现决策式句法分析器是贪婪的，在分析过程的每一步，只有最有可能的分析动作会被采纳，以至丢失了对整个分析过程中的所有分析动作的全局视角。基于此，我们提出了两种模型对分析动作进行建模。试验结果显示，在性能方面，基于动作建模的依存句法分析器要优于原决策式依存句法分析器，同时保持了较低的时间复杂度。

本文组织如下：在第二节中将介绍原决策式依存句法分析器；在第三节，我们将阐述两种基于动作建模的依存句法分析方法的细节；试验及结果将在第四节中阐述；第五节是得出的结论。

2 决策式依存句法分析器

这一节将介绍适合于中文依存句法分析的两种决策式依存分析方法，它们分别由 Yamada 和 Matsumoto [2] 以及 Nivre 和 Scholz [3] 提出。决策式句法分析方法是将分析过程看成是一步步作用于输入句子之上的分析动作的序列。分析动作主要是建立词和词之间的依存关系，本文中依存关系是有方向的箭头而不输出依存关系的类型。由于分析动作的集合只有有限个元素，我们可以训练出关于分析动作的分类器。在测试时，由训练出的分类器来决定分析动作。

为了解释决策式依存分析方法如何进行，下面简要介绍 Yamada 的方法。由于 Nivre 采用相似的决策式方法进行依存句法分析，只是使用了不同的数据结构和分析动作，因此我们省略了对 Nivre 方法的介绍。

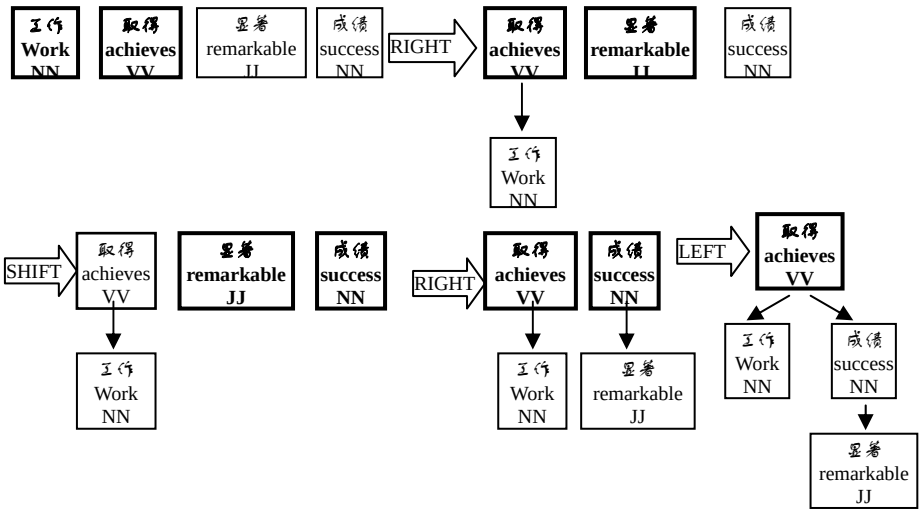

图 1：Yamada 方法的分析过程的图例，例句为“工作取得显著成绩”。

在 Yamada 的方法中共有三种分析动作被用来建立两个焦点词之间的依存关系。两个焦点词是指在当前分析状态下，当前子树的根节点和其后续（右）子树的根节点。每当采用一个分析动作时，就会得到一个新的分析状态，即得到一个部分分析完的依存树。特征主要是围绕这两个焦点词来提取的。在训练阶段，特征及其对应的分析动作组成训练数据。在测试

阶段，由分类器在获得的特征的基础上确定分析动作。当不再有依存关系需要建立时，分析过程结束。关于三种分析动作的细节如下：

LEFT：建立右焦点词依存于左焦点词的依存关系。

RIGHT：建立左焦点词依存于右焦点词的依存关系。

SHIFT：不建立依存关系，只转移句法分析的焦点，即新的左焦点词是原来的右焦点词，依此类推。

关于这三种分析动作和依存分析过程的图例如图 1，焦点词位于粗体框中。

依存关系的建立只是针对焦点词，即子树的根节点。一旦 LEFT 或 RIGHT 被采用，两个焦点词中的一个会成为另一个的子节点，从而在此后的依存分析中不会再被考虑。

共有两种情况会采用 SHIFT。一种情况是在焦点词之间不存在依存关系；另一种情况是在焦点词之间存在依存关系，但是对于被支配的焦点词，仍然存在其他子树的根节点是其子节点。对于第二种情况，焦点词之间的依存关系不会被建立，而是采用 SHIFT，从而使被支配的焦点词有机会在接下去的分析中成为父节点。在被支配焦点词的所有子节点均依存过来后，其与另一个焦点词之间的依存关系将会被建立。

3 基于动作建模的依存句法分析

在第二章中介绍的决策式依存分析方法是贪婪的方法。在给定当前分析状态下，他们在每一步分析中只选择最有可能的分析动作，而丢失了对以前和将来分析动作的全局视角。为了克服贪婪的缺点，本文提出两种基于动作建模的依存分析模型：动作链模型、n 阶段模型。

3.1 动作链模型

整个决策式句法分析过程可以看作是马尔可夫链。在每一步分析中会有若干个候选分析动作。句法分析的目标是在马尔可夫假设下寻找最有可能的分析动作序列。具体如下：

首先定义依存树的概率。给定输入的原始句子 S，依存树 T 的概率为：

$$P(T \mid S) = \prod_{i=1...n} P(d_i \mid d_1 ... d_{i-1}, S) \qquad (1)$$

其中 d_i 指在时刻 i 的分析动作。

引入变量 $context_{d_i}$ 表示分析动作 d_i 被采纳后得到的分析状态。

$$context_{d_i} = Construct(d_i, context_{d_{i-1}}) \qquad (2)$$

其中 Construct 函数定义了分析动作 d_i 作用在 $context_{d_{i-1}}$ 上后可以得到的分析状态。

假设采用了一系列动作 $d_1 d_2 ... d_n$ 后得到一系列分析状态 $context_{d_1} context_{d_2} ... context_{d_n}$，则公式（1）中的每一个 d_i 可以替换为 $context_{d_i}$，则公式（1）转换为如下的形式：

$$\prod_{i=1...n} P(context_{d_i} \mid context_{d_1}, ..., context_{d_{i-1}}) \qquad (3)$$

$$\approx \prod_{i=1...n} P(context_{d_i} \mid context_{d_{i-1}}) \qquad (4)$$

$$= \prod_{i=1...n} P(d_i \mid context_{d_{i-1}}) \qquad (5)$$

在马尔可夫假设下，由（3）式可以推出（4）式。请注意（5）式相关于分析动作的分类器，它表示在分析状态 $context_{d_{i-1}}$ 下采用分析动作 d_i 的概率是多少。如果我们训练出一个

可以输出概率的关于分析动作的分类器，就可以通过计算分析动作概率的乘积来计算 *P(T|S)*。在本文中所使用的分类器是 Libsvm [12] ，此软件支持多分类问题和带有概率输出的训练与预测。

在此模型中，目标是选择一个分析动作序列，使得这一序列所建造的依存树具有最大的概率。

$$\max P(T \mid S) = \max_{d_1 \ldots d_n} \prod_{i=1 \ldots n} P(d_i \mid context_{d_{i-1}}) \qquad (6)$$

因为这一模型选择的是最有可能的动作序列，而不是在每一步上选择最有可能的动作，所以避免了原始决策式句法分析器的贪婪性质。

动作序列的解码算法为 Viterbi，其时间复杂度是原始的决策式依存分析算法的 m 倍，这里 m 指的是分析动作集合中元素的个数。在 Yamada 的方法中，分析动作集合共有 3 个分析动作，其原始算法的时间复杂度在最差情况下与输入句长成二次关系；在 Nivre 的方法中，分析动作集合共有 4 个分析动作，其原始算法的时间复杂度和输入句长成线性关系。

3.2 n 阶段模型

汉语存在两种容易引起混淆的情况。一个是在第二章中介绍的关于 SHIFT 的歧义。SHIFT 可以在两种情况下采用，这就引起了歧义。SHIFT 歧义发生的典型情况是：在当前分析状态下，一个焦点词仍然有除了另一个焦点词之外的其它子树的根作为其子节点，虽然焦点词之间存在依存关系，但仍然采用 SHIFT。在中文中，这种情况常常发生在动词和介词身上，因为动词和介词是可以支配其右方的词的。我们将其归类为 V-V 和 V-P，V-V 是指两个焦点词都是动词，V-P 是指左焦点词是动词，右焦点词是介词。除了 SHIFT 的歧义，另一种易混淆的情况是动词和名词之间的依存关系难以判断。在中文中，动词可以支配或者修饰其右方的名词，而没有任何词语形态上的变化信息作为指导。我们将这种情况归类为 V-N，左焦点词是动词，右焦点词是名词。

Jin et al. [7] 曾提出过 2 阶段的方法来解决 V-V 的情况。在他们的方法中，关于 V-V 的分析动作在第一个阶段总是 ***SHIFT***。第一个阶段后，由于动词之间的依存关系没有被建立，分析的结果是一个子树的序列。在第二个阶段，这些留下的子树被重新分析，从而得到一棵完整的依存句法树。2 阶段方法的优点是，在经过第一个阶段的分析后，会产生一个更加清晰的上下文环境以便在第二个阶段判断动词间的依存关系。这样可以避免过于贪婪地建立动词间的依存关系，将这些关系的确定留在第二个阶段解决。

我们还观察到另两个易混淆的情况，即 V-P 和 V-N。关于这两种情况的 2 阶段模型我们也做了试验并进行了比较。接着我们提出了 n 阶段模型，这是一种序贯的模式，将所有易混淆的情况集成起来处理。本文中针对三种易混淆的情况，即：V-P、V-N 和 V-V。在第一个阶段，关于 V-P、V-N 和 V-V 的分析动作均为 ***SHIFT***。接着在每一个阶段我们只针对一种情况建立依存关系而 ***SHIFT*** 其它种情况，直至这三种情况全被处理且所有的依存关系被建立。总体共有 4 个阶段，但阶段的个数可以很灵活，这依赖于我们想处理多少种易混淆的情况，这也是为什么这个模型叫做 n 阶段模型的原因。

4 实验及结果

4.1 实验设置

试验数据取自宾州中文树库（CHTB）5.0 版 [6] 。树库共有 50 万词，大部分取材于新华社新闻，Sinorama 新闻杂志以及香港新闻。为了在训练集、开发集和测试集中平衡各种语料来源，我们将语料分割为如表 1 所示。

	CHTB	词数
训练集	001-815，1001-1136	434,936
开发集	886-931，1148-1151	21,595
测试集	816-885，1137-1147	50,319

表 1：数据集的分割。

宾州中文树库是短语结构树库，其由短语类型和语法功能联合标注。为了将短语结构转化为依存结构树库，我们需要抽取短语的头的规则。在 Sun 和 Jurafsky [8] 的工作中，他们报告了一系列抽取短语头的规则，并取得了在产生式方法中最好的句法分析性能。我们采用了此规则。

试验采用一下评价指标：

依存正确率（DA）：除了标点及根结点，全部词中被分配了正确的头节点的词的百分比。

根节点正确率（RA）：根节点中正确的根节点的百分比。

完全匹配率（CM）：所有的句子中依存结构全部正确的句子的正确率。

4.2 决策式句法分析同产生式句法分析及判别式句法分析的比较

我们实现了两种原始决策式依存分析方法 [2][3]。为了比较，我们使用 dbparser——由 Daniel M. Bikel [9]实现的产生式句法分析器，以及 MSTParser——由 Ryan Mcdonald [10] 实现的判别式依存句法分析器。其中 dbparser 是 Collins 短语结构句法分析器 [11] 的模仿版本，并不是依存句法分析器，但我们可以使用抽取短语头的规则来获得依存结构。我们将本文使用的抽取短语头的规则也应用于 dbparser 的训练和预测上。各句法分析器性能比较如表 2。

	DA	RA	CM
Yamada	**82.82**	**70.13**	**30.39**
Nivre	82.69	68.19	29.82
dbparser	80.13	70.09	27.56
MSTParser	81.26	69.03	25.72

表 2：原始决策式依存句法分析器同产生式句法分析器和判别式依存句法分析器性能的比较。

从表 2 中可以看到，决策式依存分析方法要明显优于产生式句法分析方法和判别式句法分析方法。有趣的是，Wang [5] 的中文短语结构句法分析也取得了类似的结果。但对于英文的句法分析来说，结果则相反，决策式方法的性能要低于另两种方法。原因可能在于中文是比英文更灵活的语言，中文没有时态、语态和形态上的变化，是一种语义驱动的语言。所有这些灵活性决定中文的句法分析需要更丰富的特征。决策式句法分析可以提供丰富的特征，因为给定当前的分析状态，决策式句法分析方法可以在任何地方灵活地抽取特征。但是在产生式句法分析和判别式句法分析中，特征仅限于在局部以避免计算复杂度呈指数级增长。对

于产生式句法分析，特征一般仅限于与一个短语的生成式规则相关，是上下文无关的。对于判别式依存句法分析，特征一般仅与一个依存边相关，虽然包含了一些静态特征（只和原始输入句子相关）[10] 。

4.3 基于动作建模的依存句法分析器的性能

表 3 中为基于动作建模的依存句法分析器的性能。同表 2 相比，基于动作建模的依存句法分析器均优于原始的决策式依存句法分析器。其中动作链模型取得最优的效果。

	动作链模型			n 阶段模型			原始方法		
	DA	RA	CM	DA	RA	CM	DA	RA	CM
Yamada	**83.47**	**68.23**	**31.44**	83.42	68.98	30.71	82.82	70.13	30.39
Nivre	**83.20**	**68.08**	**30.39**	83.06	68.16	30.13	82.69	68.19	29.82

表 3：基于动作建模的依存句法分析器的性能。

n-phase 列仅列出了在所有 2 阶段模型和序贯模型中最优的性能。在序贯模型中序贯的顺序为 V-P，V-N，V-V，这是以各种易混淆的情况的难易程度来排序的。除了 V-N，其它 n 阶段模型均提升了原始决策式依存句法分析器的性能，且 V-P 取得了在 n 阶段模型中最优的性能。但是提升程度并不如我们的预期，这是由于前一阶段的错误会传播到下一个阶段，这同样解释了为什么序贯模式的性能低于 V-P 和 V-V。

		DA	RA	CM
动作链模型	***ideal***	88.64	84.78	46.35
	first	**84.05**	**73.39**	**32.34**
Yamada		82.82	70.13	30.39

表 4：关于输出 n-best 结果的依存句法分析性能，n 取 20。

我们也采用分析动作模型进行了输出 n-best 结果的实验。实验中采用的决策式分析框架是 Yamada 的方法。性能如表 4，其中行 ***first*** 表示的是在 n-best 结果中位列第一的依存树的性能。从中可以看到，相比于原始决策式依存分析方法，依存正确率（DA）提升了约 1.2%，根节点正确率（RA）提升了约 3.3%，完全匹配率（CM）提升了约 2.0%。从试验结果可以看出，当为每个分析动作保存 n 个最优结构，可以有效地避免贪婪性，正确的依存树在分析结束时有可能排在第一位。行 ***ideal*** 表示的是一个“完美”的性能，设想有一个“完美”的依存句法分析器可以从最优的 n 个结果中挑出最正确的结果，这里最正确的结果是指在同测试集相比较后挑出的正确率最高的一个依存句法树。可以看到依存正确率达到 88.64%，明显高于原始决策式依存分析器的正确率 82.82%，而且取得了 46.35%的完全匹配率，远远高于原始的 30.39%。对重排序（reranking）的研究将具有潜力提升中文依存句法分析器的性能。

5 结论

本文将中文决策式依存分析方法同产生式方法和判别式方法做了比较。实验结果显示决策式依存分析方法取得了三者中最好的性能。由于原始的决策式依存分析方法是贪婪的方法，我们为了避免这种贪婪性，提出了两种基于动作建模的依存分析方法。结果显示基于动作建模的依存分析方法要显著优于原始的决策式依存分析方法。

参考文献：

[1] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In Proceedings of the Sixth Workshop on ComputationalLanguage Learning (CoNLL). 2002.

[2] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In Proceedings of the 8th InternationalWorkshop on Parsing Technologies (IWPT). 2003.

[3] Joakim Nivre and Mario Scholz. Deterministic dependency parsing of English text. In Proceedings of the 20th International Conference on Computational Linguistics (COLING). 2004.

[4] Kenji Sagae and Alon Lavie. A classifier-based parser with linear run-time complexity. In Proceedings of the 9th International Workshop on Parsing Technologies (IWPT). 2005.

[5] Mengqiu *Wang*, Kenhi Sagae, and Teruko Mitamura. A fast, accurate deterministic parser for Chinese. In Proceedings of the 44th AnnualMeeting of the Association for ComputationalLinguistics (ACL). 2006.

[6] Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. Natural LanguageEngineering. 2005.

[7] Meixun Jin, Mi-Young Kim, and Jong-Hyeok Lee. Two-phase shift-reduce deterministic dependency parser of Chinese. In Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP). 2005.

[8] Honglin Sun and Daniel Jurafsky. Shallow semantic parsing of Chinese. In Proceedings of the HLT/NAACL. 2004.

[9] Daniel M. Bikel. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. thesis of University of Pennsylvania. 2004.

[10] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers.In Proc. of the 43rd AnnualMeeting of the Association for ComputationalLinguistics (ACL). 2005.

[11] Michael Collins. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis of University of Pennsylvania. 1999.

[12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. 2005.