基于树结构模式挖掘的非监督中文短语结构句法分析

张晓甜，赵 海
教育部-微软智能计算与智能系统重点实验室
上海交通大学 计算机科学与工程系，上海 200240
E-mail: xtian.zh@gmail.com; zhaohai@cs.sjtu.edu.cn

摘 要：本文研究了非监督的中文短语结构句法分析。我们首次精确重现了 Rens Bod 在[2]中阐述的非监督数据驱动模型 U-DOP。应用 U-DOP 方法在 CTB[14]上达到了提出该方法的原始文献所报道的结果，同时，按照已有文献的评测策略，在已知的基于词性串分析的非监督短语结构句法分析系统中，本文报道了在可比较实验条件下的最高性能。进一步地，所实现的 U-DOP 的结果与另一个基于词的非监督句法分析器 CCL [13] 的结果进行了经验性的比对和分析。

关键词：非监督；短语结构；句法分析

Unsupervised Chinese Phrase Parsing Based on Tree Pattern Mining

Zhang Xiaotian, Zhao Hai
MOE-Microsoft Key Laboratory of Intelligent Computing and Intelligent System
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240
E-mail: xtian.zh@gmail.com; zhaohai@cs.sjtu.edu.cn

Abstract: This paper investigates unsupervised phrase parsing for Chinese. It is the first time according to our best knowledge that unsupervised data oriented (U-DOP) model described in [2] is fully and exactly re-implemented. Our U-DOP implementation achieves the similar results on CTB as reported in [2]. Moreover, using the evaluation measure in [3], our system achieves the highest F1 score among all the existing POS-sequence based unsupervised phrase parsing models. We also give a detailed comparison between the performance of our U-DOP system and the unsupervised CCL parser [13] in terms of prediction accuracy on different kinds of phrases.

Keywords: unsupervised; phrase structure; syntactic parsing

1 前言

非监督句法分析是在没有句法树标注的数据中自动寻找模式并且挖掘出句法结构的学习技术。目前，研究者们主要提出了四种非监督的分析模型，CCM [9]、DMV [10]、UDOP [2]和 CCL [13]。CCM、DMV 和 UDOP 模型之间最大的差别在于对一个句法树或子句法树赋予的先验概率的不同。比如模型 CCM [9]，概率由结构中成分和非成分的词性序列以及这些词性序列的上下文定义。在模型 U-DOP [2]中，句法结构的概率定义为子树概率的乘积。而 DMV 模型[10]则将连接的两个词性的依存弧赋予概率。CCL[13]模型是启发式的，基于一种新的基于弧的表示方式，其分析依赖一系列复杂的规则、权重和阈值的集成。在这些非监督句法分析模型中，U-DOP 具有完全基于树计数来分析的优点并取得了比 CCM 和 DMV 都更高的 F1 值。不过，由于 CCL 是从词串来分析而其他模型均是基于词性串分析，所以 CCL 与其他模型在性能上严格来说不可比较。

Rens Bod 所提出的 U-DOP 模型[2]实际上是 DOP 模型[1]向非监督句法分析的拓展，同样采用 DOP 模型中基于树结构模式挖掘的方法。模型的一个主要部分是根据基于子树提取的 PCFG 规则来寻找前 n 个最佳分析树的 CYK 句法分析器。Liang Huang 和 David Chiang 于 2005 年在文献[5]中提出了三种搜索前 n 个最佳分析树（n-best parsing tree）的算法，其中具有较优性能的第三种算

法正是 Víctor M. Jiménez 和 Andrés Marzal 在[6]中所描述的算法的一般化的拓展。而文献[6]中描述的算法主要思想为首先用 CYK 算法计算最佳分析树，然后其他候选树可以按照概率顺序依次计算得到，而这只需相对于第一步计算的较少时间。因此，我们采用文献[6]中所描述的算法。

本文中，我们报告了实现 U-DOP 模型的关键技术过程及实验结果和分析对照。在本文的工作之前，由于已有文献在关键性的一些细节上的描述缺失，U-DOP 模型仅有原始作者报道的结果。据我们所知，这是 U-DOP 模型第一次全面的再现。本文报道的的系统实现已经建立了一个开源软件项目，供同行下载使用[1]。

2 U-DOP 模型的实现

U-DOP 模型的算法包含三个主要步骤。

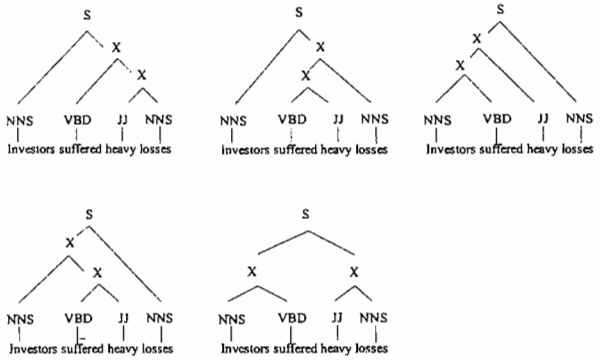

图 1　词性串 NNS VBD JJ NNS 所有可能的二叉树（摘自文献[2]）

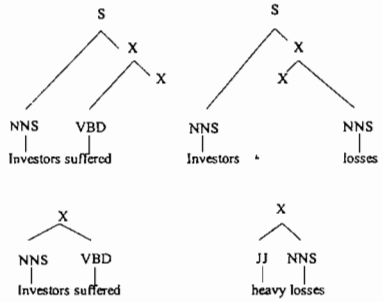

图 2　词性串 NNS VBD JJ NNS 的二叉树的部分子树（摘自文献[2]）

第一步是对训练语料（不包含树结构标记）枚举所有可能的二叉树并且从其子树中提取 PCFG 规则。如图 1，对于词性串“NNS VBD JJ NNS”总共有 5 种形状的二叉树。对于所有训练语句计算所有可能的二叉树将得到大量的二叉子树（如图 2），为了使后面的分析具有计算代价上的可行性，需要按照[2]对所得二叉树子树进行随机抽样。对于 7 个词组成的句子抽取 60%的子树，而对于长 8、9、10 个词的句子依次抽取 30%、15%和 7.5%的子树，不考虑超过 10 个词的句子。之后则按照[4]产生 PCFG 规则，以便利用这些子树进行有效的句法分析。对于每个非终结符 A_j，其子节点为 B_k 和 C_l，将会产生八条如下规则：

$$A_j \to BC \quad (1/a_j) \qquad A \to BC \quad (1/a)$$
$$A_j \to B_k C \quad (b_k/a_j) \qquad A \to B_k C \quad (b_k/a)$$

[1] 本文实现的源代码可以从 http://sourceforge.net/projects/udop/ 自由下载。

$$A_j \to BC_l \quad (c_l / a_j) \qquad A \to BC_l \quad (c_l / a)$$
$$A_j \to B_k C_l \quad (b_k c_l / a_j) \qquad A \to B_k C_l \quad (b_k c_l / a)$$

其中 A，B，C 表示标号，下标 j，k，l 表示位置，b_k 和 c_l 是以结点 B_k 和 C_l 为根的子树的数目，且有 $a_j=(b_k+1)(c_l+1)$。根据 PCFG 规则的性质，每个子树的推导都对应一个同形的具有相同概率的 PCFG 推导，因此可以采用 n-best CYK 分析算法来寻找基于所得 PCFG 规则的前 n 个最佳分析树。

文献[7]指出 DOP 的估计量是有偏的和不一致的，因此按照文献[1]通过增加修正因子 α 来修正估计量，其中 α 指训练数据中非终结符 A 出现的次数，PCFG 规则修正如下：

$$A_j \to BC \quad (1 / a_j) \qquad A \to BC \quad (1 / a\alpha)$$
$$A_j \to B_k C \quad (b_k / a_j) \qquad A \to B_k C \quad (b_k / a\alpha)$$
$$A_j \to BC_l \quad (c_l / a_j) \qquad A \to BC_l \quad (c_l / a\alpha)$$
$$A_j \to B_k C_l \quad (b_k c_l / a_j) \qquad A \to B_k C_l \quad (b_k c_l / a\alpha)$$

实验表明，修正因子使 CTB 的 F1 值提高 13%，因此，它是 U-DOP 模型实现中的一个关键要素。

Rens Bod 在[2]中并没有阐明 PCFG 规则是从带标点符号的词性串中产生，还是从不带标点符号的词性串中产生。为了求证，在 PTB[12]英文树库 WSJ10[1] 上分别用带标点符号和不带标点符号的词性串产生规则，发现用带标点符号的词性串可以产生与[2]中相同数量的 $14.8*10^6$ 条规则，而且小规模测试发现用带标点符号的词性串所产生的规则预测 F1 值更高，这与标点符号对于句法结构的重要作用有关。因此，采用带标点符号的词性串来生成 PCFG 规则。在实验中，从 CTB10 的 3.0 版本中提取了 $6.2*10^6$ 条规则。

在对测试语料上进行预测评估时，U-DOP 将会根据已产生的 PCFG 规则通过 n-best CYK 分析算法搜索对于测试词性串的前 n 个最佳分析树。这里我们实现了[6]中描述的 n-best CYK 算法。其主要思想是在用 CYK 算法得到最好分析树之后，大量的候选树可以在相对上一步较短的时间内得到。为了采用[6]中描述的算法，首先对 PCFG 规则的概率取负对数，将寻找前 n 棵概率最大树问题转换为寻找前 n 棵概率负对数之和最小的树的问题。

用 $A_{i:k}$ 来表示统治从第 i 个词性到第 k 个词性的非终结符。$T^n(A_{i:k})$表示第 n 棵最优树，$\Upsilon^n(A_{i:k})$ 是 $T^n(A_{i:k})$的候选树组成的集合，而 $T^n(A_{i:k})$就是 $\Upsilon^n(A_{i:k})$中权值最小的那棵树。用 CYK 分析算法计算出最优树 $T^1(A_{i:k})$，然后用下面的递归式通过 $T^{n-1}(A_{i:k})$计算出 $T^n(A_{i:k})$。<>三元组中第二项和第三项表示第一项的左右子树。

令

$$\Upsilon^1(A_{i:k}) = \{< A_{i:k}, T^1(B_{i:j}), T^1(C_{j+1:k}) > \ : \exists A \to BC \ \forall j \ s.t \ i \leqslant j < k\} \tag{1}$$

对于 n>1，设

$$T^{n-1}(A_{i:k}) =< A_{i:k}, T^p(B_{i:j}), T^q(C_{j+1:k}) > \tag{2}$$

如果 q=1，则

$$\begin{aligned}\Upsilon^n(A_{i:k}) = &(\Upsilon^{n-1}(A_{i:k}) - T^{n-1}(A_{i:k})) \\ &\cup \{< A_{i:k}, T^p(B_{i:j}), T^{q+1}(C_{j+1:k}) >\} \\ &\cup \{< A_{i:k}, T^{p+1}(B_{i:j}), T^q(C_{j+1:k}) >\}\end{aligned} \tag{3}$$

否则

[1] WSJ10 与 CTB10 指 WSJ 和 CTB 中长度不超过 10 个词的语料。

$$\Upsilon^n(A_{i:k}) = (\Upsilon^{n-1}(A_{i:k}) - T^{n-1}(A_{i:k})) \cup \{< A_{i:k}, T^p(B_{i:j}), T^{q+1}(C_{j+1:k}) >\} \tag{4}$$

之后有

$$T^n(A_{i:k}) = argmin_{T \in \Upsilon^n(A_{i:k})} W(T)$$

由于子树的不同推演有可能导致相同的树形，因此在计算出最优的前 100 棵树之后，U-DOP 的最后一步是将具有相同树形的树的概率求和，而具有最大概率的树形就是最后的分析结果。

3 实验

我们继续按照[2, 8]中所述方法评测所实现的 U-DOP 系统的效果。

标准语料 $G=[G_i]$，分析结果 $P=[P_i]$，则定义无标号准确率和召回率为

$$UP(P,G) \equiv \frac{\sum_i |brackets(P_i) \cap brackets(G_i)|}{\sum_i |brackets(P_i)|} \tag{5}$$

$$UR(P,G) \equiv \frac{\sum_i |brackets(P_i) \cap brackets(G_i)|}{\sum_i |brackets(G_i)|} \tag{6}$$

$$UF_1(P,G) = \frac{2}{UP(P,G)^{-1} + UR(P,G)^{-1}} \tag{7}$$

另外，如 Rens Bod 在[3]中所述，“在评测 U-DOP 时，要将测试语料转换为二叉树，否则 F1 值无意义[8, 9, 10]”。因为 U-DOP 模型的分析结果是二叉树，而标准树则更平。因此，如果按照式（5）和（6）评测打分，则 U-DOP 更倾向于具有较低的准确率和相对高的召回率。而在二叉树化后的测试集上测试明显会增加预测正确的括弧数量。然而，一方面，据我们所掌握的知识，当前 CCM 和 DMV 模型的开源实现[1]在未二叉树化的测试集上就已成功重现了[8, 9, 10]的结果，[11]中的图 6 也暗示 CCM 并不是在二叉树化的测试集上进行评测的；另一方面，由于对于每个句子，其所有二叉树所产生的括弧数目是一样的，若 UR 定义为二叉树化测试语料后预测正确的括弧数目除以二叉树化后的标准括弧数目，则会导致在二叉树化的测试集上进行评测产生相等的准确率和召回率，而这与[2, 8]中结果矛盾。或者，若 UR 等于二叉树化测试语料后预测正确的括弧数目除以二叉树化前的标准括弧数目，则同样是无意义的，因为在二叉树化过程当中产生了标准括弧中不存在的新的括弧。因此，一个有意义的使用二叉树化测试集的评价准则是：UR 仍然按照上式计算，而 UP 则按下式计算，

$$UP(P,G) \equiv \frac{\sum_i |brackets(P_i) \cap brackets(B_i)|}{\sum_i |brackets(P_i)|} \tag{8}$$

其中 $B=[B_i]$ 是二叉树化后的标准集[2]。我们采用右二叉树化测试语料。

[9, 10]中使用了 CTB 的 3.0 版本[3]进行测试，为了与之对照，本文同样使用 CTB 的 3.0 版本来测试所实现的系统。表 1 比较了不同模型的 F1 值。

CTB3.0 中不超过 10 个词的句子约 2137 句，实验花费 52 个小时（英特尔处理器 X5560 @ 2.80GHz）。CTB5.0 版本上同样进行了测试，结果 F1 值为 46.5。表格中 $UDOP_{non\text{-}binarized}$ 表示按照式（6）计算 UP，$UDOP_{binarized}$ 表示按照式（7）计算 UP。

---

[1] 可从 http://www.cs.famaf.unc.edu.ar/francolq/en/proyectos/dmvccm 下载。

[2] 如何对于测试语料进行二叉树化在文献[2]中并未提及。

[3] 文献[2]中采用的 CTB 版本并未在论文中说明。

表 1　CTBv3.0 上非监督句法分析模型测试结果

模型	UP	UR	F1
CCM	34.6	64.3	45.0
DMV	35.9	66.7	46.7
DMV+CCM	33.3	62.0	43.3
UDOP(Bod)	36.3	64.9	46.6
UDOP*			42.8
$UDOP_{non\text{-}binarized}$(ours)	34.4	53.2	41.8
$UDOP_{binarized}$(ours)	42.3	53.2	47.1
CCL	50.1	51.1	50.6

表 2　CTBv5.0 上的 U-DOP 系统的测试结果

模型	UP	UR	F1
$UDOP_{non\text{-}binarized}$(ours)	40.0	55.7	46.5

表 3　CTB10v3.0 上 U-DOP 和 CCL 对于各主要短语类型的分析准确率比照

短语类型	百分比	U-DOP	CCL	共有
CP	4.0%	0.146	0.099	0.012
DNP	2.7%	0.246	0.234	0
NP	13.2%	0.396	0.209	0.125
NP-OBJ	9.2%	0.319	0.271	0.114
NP-PN	2.5%	0.198	0.154	0.037
NP-SBJ	10.9%	0.372	0.321	0.156
PP	1.4%	0.244	0.100	0.022
QP	2.9%	0.750	0.190	0.168
VP	37.2%	0.228	0.316	0.113

进一步，我们统计了所实现的 U-DOP 系统和 CCL 系统在预测不同类型短语结构时的准确率。表 3 的第五列是在预测各类型的短语时 U-DOP 系统和 CCL 系统结果中所共有的同时预测正确的括弧所占百分比。从表中可以看出由于 U-DOP 在预测 VP 短语时准确率较 CCL 相差较大，而由于 VP 在语料中所占的比例较大，因此尽管 U-DOP 在其他类型短语的预测上均比 CCL 要好，而整体上准确率却比 CCL 要低。

4　结语

本文报道了 U-DOP 模型的实现的关键细节过程。已有文献[2]中缺少的必要的技术细节，比如标点符号的处理，分析算法以及评测标准，对于这些问题，我们通过实验以及理论分析一一在本文中给予了完整解释。在中文树库上的测试结果表明如果采用[3]中所指的二叉树化测试集方法进行评测，本文所报道的结果比其他基于词性的非监督系统的结果更高。所实现的 U-DOP 模型已经建立了一个开源软件项目，供同行自由下载使用。另外，通过对 U-DOP 和 CCL 对于不同类型短语的分析结果的比较，发现 U-DOP 对于除 VP 以外的短语类型都有相对较高的准确率，而在占有相当比例的 VP 短语上准确率较低正是导致 U-DOP 整体性能低于 CCL 的主要原因。这一结果指明了改进不同的非监督句法分析器的努力方向。

致 谢

本文的研究工作受国家自然科学基金项目（编号 60903119，60773090 以及 90820018）资助。本文的工作同时得益于和 Mark Johnson 教授的富有启发性的讨论。

参 考 文 献

[1] Rens Bod. An efficient implementation of a new dop model. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics Volume 1*, EACL'03, pages 19-26, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[2] Rens Bod. Unsupervised parsing with u-dop. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X'06, pages 85-92, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[3] Rens Bod. Linguistic relevance of unsupervised data-oriented parsing. In *Machine Learning and Cognitive Science of Language Acquisition*, June 2007.

[4] Joshua Goodman. Efficient algorithms for parsing the dop model, 1996.

[5] Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing'05, pages 53-64, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[6] Víctor M. Jiménez and Andrés Marzal. Computation of the n best parse trees for weighted and stochastic context-free grammars. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 183-192, London, UK, 2000. Springer-Verlag.

[7] Mark Johnson. Squibs and discussions: the dop estimation method is biased and inconsistent. *Comput. Linguist.*, 28: 71-76, March 2002.

[8] Dan Klein. *The unsupervised learning of natural language structure*. PhD thesis, Stanford, CA, USA, 2005. AAI3162386.

[9] Dan Klein and Christopher D. Manning. A generative constituent context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL'02, pages 128-135, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[10] Dan Klein and Christopher D. Manning. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL'04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[11] Dan Klein and Christopher D. Manning. Natural language grammar induction with a generative constituent-context model. *Pattern Recogn.*, 38: 1407-1419, September 2005.

[12] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.*, 19: 313-330, June 1993.

[13] Yoav Seginer. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384-391, 2007.

[14] Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. Building a largescale annotated Chinese corpus. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING'02, pages 1-8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.