基于增益的隐马尔科夫模型的文本组块分析*)

李 珩 杨 峰 朱靖波 姚天顺
(东北大学计算机软件与理论研究所 沈阳110004)

摘 要 本文提出一种基于增益的隐马尔科夫模型(Transductive HMM)的方法,用于文本组块(Text Chunking)分析的研究。该方法将一些上下文信息导入隐马尔科夫模型(HMM),构造增益的隐马尔科夫模型(Transductive HMM)。该模型不需要修改标准的隐马尔科夫模型的训练和标注过程,只需要对训练语料根据导入的上下文信息进行相应的转换。实验结果显示,该方法在文本组块分析方面是有效的。

关键词 文本组块,隐马尔科夫模型,增益的隐马尔科夫模型

Transductive HMM Based Text Chunking

LI Heng YANG Feng ZHU Jing-Bo YAO Tian-Shun
(Institute of Computer Software and Theory, Northeastern University, Shenyang 110004)

Abstract In this paper, we present a technique to solve text chunking task as a tagging problem using a Hidden Markov Model-based approach (HMM). We incorporate the relevant information into the models to construct a transductive HMM which gives more complete contextual models. In this way, we don't need to change either the training or the tagging process, just transform training corpus to take into account the information. The experiment results show that it is an effective approach.

Keywords Text chunking, HMM, Transductive HMM

1. 引言

文本组块分析作为句法分析的预处理阶段,通过将文本划分成一组互不重叠的片断,来达到降低句法分析的难度,这些片断是非递归的,即片断不能嵌套,这些片断定义为Chunk[1]。请看一个文本组块的例子:

[PP Under][NP the existing contract],[NP Rockwell][VP said],[NP it][VP has already delivered][NP 793][PP of][the shipsets][PP to][NP Boeing].

当然,也可以通过为Chunk加标记来表示文本组块。本文采用IOB2[6]的标注集合,该标注集合包含三种类型的标记:B-X表示Chunk类型为X,并且是该Chunk的起始词,I-X表示Chunk类型为X,并且是该Chunk的非起始词,O表示不在任何Chunk内的词。于是,上述的例子也可以表示如下:

Under/B-PP the/B-NP existing/I-NP contract/I-NP,/O Rockwell/B-NP said/B-VP,/O it/B-NP has/B-VP already/I-VP delivered/I-VP 793/B-NP of/ B-PP the/B-NP shipsets/I-NP to/B-PP Boeing/B_NP./ O

这样,文本组块分析过程也可以看成对文本进行Chunk标注过程。目前,应用于文本组块的方法包括,基于转换的学习(Transformation based learning)算法[5],基于记忆的学习(Memory based learning)算法[9,10],基于隐马尔科夫模型(Hidden Markov Model)算法[15],最大熵模型(Maximum Entropy Model)算法[11],支持向量机(Support Vector Machine)算法[12,13]等。

本文通过构造一个增益的隐马尔科夫模型来进行文本组块分析,介绍了通过导入各种上下文信息构造增益的隐马尔科夫模型的过程,给出了实验结果,最后是结论及未来的工作。

2. 增益的隐马尔科夫模型

任何一种随机语言模型都需要在下面两方面进行折衷:一方面模型需要应用尽可能多的上下文信息进行更为精确的预测;另一方面又要考虑到模型参数的紧凑性和可实现性而不得不作出必要的独立性假设。即使通过独立性假设对上下文信息做了大量简化之后仍然无法解决数据稀疏问题而不得不求助于各种参数平滑技术。

所谓的隐马尔科夫模型是一个四元组,$\langle s^1, S, W, E\rangle$,$s^1 \in S$是初始状态,$S=\{s^1, s^2, \cdots, s^e\}$是状态集,也称为隐藏层,$W=\{w_1, w_2, \cdots, w_w\}$为输出符号表,称为观察层。

E为状态转移函数$\{p_{s,t}\}$,$s,t \in S$,满足$\sum_{t\in S} p_{s,t}=1, \forall\ s\in S$。

另外还有两个概率函数。状态概率函数$\pi: s\to[0,1]$,满足$\sum_{s\in S}\pi(s)=1$,输出符号函数概率函数$\{b_s: W\to[0,1]\}$,$s\in S$,满足$\sum_{w\in W} b_s(w)=1, \forall\ s\in S$。

HMM模型主要有两个缺点:1)HMM的相关算法得到都是局部的最优值;2)按照HMM的假设,HMM模型是无记

*)本工作得到国家自然科学基金资助项目(60083006),国家重点基础研究发展规划973资助项目(G19980305011),国家自然科学基金和微软亚洲研究院联合资助项目(60203019)的资助.李 珩 博士生,主要研究方向为统计语言处理;朱靖波 副教授,主要研究方向为中文信息处理;姚天顺 教授,博士生导师,主要研究方向为计算语言学和知识发现。

忆性的，不能利用上下文的信息。因为它只与其前一个状态有关，如果想利用更多的已知信息，必须建立高阶的 HMM 模型。

2.1 基于 HMM 的标注

令 O 和 I 分别代表输出序列和输入序列，则标注问题可视为计算以下条件概率的极大值：

$$\hat{O}=\arg\max_{O} P(O|I)=\arg\max_{O}\frac{P(I|O)P(O)}{P(I)} \quad (1)$$

式中 $P(O|I)$ 是已知输入序列 I 的情况下，出现输出序列 O 的条件概率。式中符号 $\arg\max_{O}$ 表示通过考察不同的候选输出序列 O，来寻找使条件概率取最大值的那个输出序列 $\hat{O}$。

考虑到分母 $P(I)$ 对给定的 I 是一个常数，不影响极大值的计算，可以从公式中删除。接着根据二阶马尔科夫假设(Second order Markov assumption)，公式(1)可以转成公式(2)：

$$\arg\max_{O}\prod_{t:1\cdots n} p(i_t|o_t)p(o_t|o_{t-1},o_{t-2}) \quad (2)$$

据文[4,7,8]报道，上述公式成功应用到词性标注中，标注准确率达到了96%左右。在引言里，本文提到文本组块分析过程也可以看成是 Chunk 标注过程，这样，就可以采用类似基于隐马尔科夫模型的词性标注方法来标注 Chunk。当然，两者又有一些不同之处。首先，作为模型的输入部分，词性标注模型的输入是词序列，而文本组块模型的输入是词序列和词性序列；作为模型的输出部分，词性标注模型的输出是词性序列，而文本组块模型的输出是 Chunk 标记序列。一方面，考虑到词序列和词性序列的各种可能组合，使得模型的输入集合非常庞大；而另一方面，由于 Chunk 标记过少，使得模型的输出集合过于一般。这样，训练出的模型不会很准确。可以适当的降低模型的输入规模，并导入一些上下文信息到模型的输出部分，来增加输出的标记集合。这就是增益的隐马尔科夫模型。

2.2 增益的隐马尔科夫模型(Transductive HMM)

这里的"增益"可以理解为通过导入上下文信息，获得更加准确的训练模型。考虑到模型可能导入的上下文信息，本文采用了简单的方法，设计了一组转换函数 f_i，该组转换函数作用于训练语料，来获取训练模型新的输入和输出信息。其好处是不需要修改模型的训练过程和标注过程，而只是将训练语料转换成需要的输入和输出信息格式。例如，如果考虑训练模型的输入部分是训练语料的词性序列，输出部分是训练语料的词性和 Chunk 标记序列，则该专用函数 f_i 表示为 $f_i(\langle w_t\_pos_t, c_t\rangle)=\langle pos_t, pos_t\_c_t\rangle$，其中 w_t 为词标记，pos_t 为词性标记，c_t 为 Chunk 标记。在实验中，本文设计了以下七种转换函数，分别是：

1) $f_{i1}(\langle w_t\_pos_t, c_t\rangle)=\langle pos_t, pos_t\_c_t\rangle$
2) $f_{i2}(\langle w_t\_pos_t, c_t\rangle)=\langle w_t, pos_t\_c_t\rangle$
3) $f_{i3}(\langle w_t\_pos_t, c_t\rangle)=\langle w_t\_pos_t, pos_t\_c_t\rangle$
4) $f_{i4}(\langle w_t\_pos_t, c_t\rangle)=\langle pos_t, c_t\rangle$
5) $f_{i5}(\langle w_t\_pos_t, c_t\rangle)=\langle w_t, c_t\rangle$
6) $f_{i6}(\langle w_t\_pos_t, c_t\rangle)=\langle w_t\_pos_t, c_t\rangle$
7) $f_{i7}(w_t\_pos_t, c_t)=\begin{cases}(w_t\_pos_t, pos_t\_c_t) w_t\in W_t\\(pos_t, pos_t\_c_t) w_t\notin W_t\end{cases}$

其中，W_t 集合的选取将在下节论述(见3.1.2节)。于是，将上述七个转换函数代入到公式(2)中，就得到了七个增益的隐马尔科夫模型，分别是：

模型1：

$$\arg\max_{O}\prod_{t:1\cdots n} p(pos_t \mid pos_{t-1}c_{t-1})\, p(pos_{t-2}c_{t-2} \mid pos_{t-1}c_{t-1},$$

$pos_{t-2}c_{t-2}$

模型2：

$$\arg\max_{O}\prod_{t:1\cdots n} p(w_t \mid pos_t\_c_t)\, p(pos_t\_c_t \mid pos_{t-1}\_c_{t-1}, pos_{t-2}\_c_{t-2})$$

模型3：

$$\arg\max_{O}\prod_{t:1\cdots n} p(w_t\_pos_t \mid pos_t\_c_t)\, p(pos_t\_c_t \mid pos_{t-1}c_{t-1}, pos_{t-2}c_{t-2})$$

模型4：

$$\arg\max_{O}\prod_{t:1\cdots n} p(pos_t|c_t)\, p(c_t|c_{t-1},c_{t-2})$$

模型5：

$$\arg\max_{O}\prod_{t:1\cdots n} p(w_t|c_t)\, p(c_t|c_{t-1},c_{t-2})$$

模型6：

$$\arg\max_{O}\prod_{t:1\cdots n} p(w_t\_pos_t|c_t)\, p(c_t|c_{t-1},c_{t-2})$$

模型7：

$$\begin{cases}\arg\max_{O}\prod_{t:1\cdots n} p(w_t\_pos_t \mid pos_t\_c_t)\, p(pos_t\_c_t \mid pos_{t-1}\_c_{t-1}, \\ \quad pos_{t-2}\_c_{t-2}) w_t\in W_t\\ \arg\max_{O}\prod_{t:1\cdots n} p(pos_t \mid pos_t\_c_t)\, p(pos_t\_c_t \mid pos_{t-1}\_c_{t-1}, \\ \quad pos_{t-2}\_c_{t-2}) w_t\notin W_t\end{cases}$$

本文采用的标注器是基于二阶的隐马尔科夫模型的标注器[4]，在整个训练和标注过程中，并没有修改任何部分，而只是将训练语料进行了相应的转换。

3 实验结果及分析

3.1 汉语组块识别

我们采用哈工大公开的中文树库语料：Chinese Treebank(可访问 http://mtlab.hit.edu.cn/download/4.TXT)作为我们的训练语料和测试语料。它包含2000个句子，21498个词。目前可以识别的组块类型包括：BDP，BAP，BMP，BNT，BNS，BNP，BVP。

考虑到汉语的训练集规模很小，我们使用一种称为"k-fold 交叉验证"(k-fold cross-validation)的方法进行实验结果的评估，这种方法进行 k 次不同的交叉验证，每次使用数据的不同分割作为训练集合和验证集合，然后对结果进行平均。把可供使用的 m 个实例分割成 k 个不相交的子集，每个子集有 m/k 个实例。然后，运行 k 次交叉验证过程，每一次使用不同的子集作为验证集合，并合并其他的子集作为训练集合。于是，每一个样例会在一次实验中被用作验证集合的成员，在 k－1次实验中用作训练集合的成员。本文取1500个句子(16,012个词)作为训练集，500个句子(5,485个词)作为验证集，并取 k＝4。实验中的性能指标定义如下：

组块查准率：

$$(\text{Precision})=\frac{\text{正确标注的组块的个数}}{\text{标注的组块的总个数}}*100\%$$

组块查全率：

$$(\text{Recall})=\frac{\text{正确标注的线块的个数}}{\text{正确的组块的总个数}}*100\%$$

$$F_\beta=\frac{(\beta^2+1)*Recall*Precision}{\beta^2*Recall+Precision}$$ 其中取值 $\beta=1$

下面我们作了两组实验，第一组实验考察了模型1-6的各组块识别结果，第二组实验考察了模型7的组块识别结果。

3.1.1 **模型1-6汉语组块识别结果** 实验1的目的，一是验证通过导入上下文信息而构造增益的隐马尔科夫模型是否

有效，二是希望通过实验结果，发现导入哪些上下文信息到模型中，有助于组块识别效果的提高。

表1 模型1-6汉语组块识别结果，最好的结果用粗体表

Chunk type	Precision Recall FB					
	1	2	3	4	5	6
ALL	**80.69**	63.68	71.69	76.23	62.84	70.46
	75.36	64.20	73.04	69.29	61.61	70.71
	77.93	63.94	72.36	72.59	62.22	70.59
BAP	**73.04**	56.14	63.11	65.91	58.82	64.46
	70.00	53.33	64.17	**72.50**	50.00	65.00
	71.49	54.70	63.64	69.05	54.05	64.73
BDP	80.00	28.57	66.67	**100.0**	40.00	70.00
	66.67	16.67	66.67	**75.00**	16.67	58.33
	72.73	21.05	66.67	**85.71**	23.53	63.64
BMP	83.11	76.83	80.58	81.30	79.25	**85.04**
	81.98	85.14	87.84	84.23	86.04	**89.64**
	82.54	80.77	84.05	82.74	82.51	**87.28**
BNP	**86.63**	59.95	69.23	86.21	61.92	70.46
	79.55	58.86	69.55	68.18	57.27	66.14
	82.94	59.40	69.39	76.14	59.50	68.23
BNS	33.33	0.00	66.67	**100.0**	25.00	80.00
	11.11	0.00	22.22	**55.56**	11.11	44.44
	16.67	0.00	33.33	**71.43**	15.38	57.14
BNT	20.00	43.90	**48.65**	33.33	34.62	38.30
	27.59	62.07	**62.07**	27.59	62.07	62.07
	23.19	51.43	**54.55**	30.19	44.44	47.37
BVP	**82.75**	65.16	74.91	66.67	57.84	66.33
	73.26	64.93	73.61	62.50	57.64	67.71
	77.72	65.04	74.26	64.52	57.74	67.01

由实验1结果可以发现：

1）由于训练语料的规模很小（只有1500句），因此输入集合为 w_i 的模型2，5的预测能力有限，结果不是很好，而输入集合为 pos_i 的模型1，4的预测能力较强，结果较好。

2）模型4比模型3的总体结果稍好，这也主要是由于训练语料规模较小的原因。在较小语料下，输入集合 pos_i 比 $w_i\_pos_i$ 的预测能力强，但随着训练语料规模的扩大，模型3的总体效果是优于模型4的，这在后面的英语组块识别中得到了证明。

3）由于模型1的输入集合为 pos_i，而输出集合为 $pos_i\_c_i$，模型的预测能力是最强的，因此总体上表现最好，这也与我们当初的预测是吻合的。

4）对于不同类型的组块，不同转换函数训练出来的模型得到了不同的结果。其中，最好的结果主要集中在模型1和模型3上。

5）如果融合上述各种模型的优势之处，则结果还会提高。于是，又作了实验2。

3.1.2 *模型7的汉语组块识别结果* 由实验1发现，最好的结果集中在模型1和3。值得注意的是，它们的输出部分是相同的，即 $pos_i\_c_i$，而不同的是它们的输入部分，一个是 $w_i\_pos_i$，一个是 pos_i，因此，我们考虑将这两种模型进行融合。于是设计了模型7，其中 W_t 集合的选取过程是这样的：首先，将实验1中的训练语料分成两部分，一部分（约为90%）用作对 W_t 进行选取的训练语料，另一部分（约为10%）用作对 W_t 进行选取的测试语料。考虑到语料选取的分布均匀性，在每十个连续句子里，取前九个句子作为训练部分，后一个句子作为测试部分。在选取 W_t 的过程中，本文重点考虑两类词汇集合 W_c 和 W_e，W_e 是 chunk 标注错误率比较高的词汇的集合，W_c 是训练集中属于几种特定的 chunk 类型的高频词汇，包括 BNP，BVP，BMP，BAP 等，则 $W_t=W_c\cup W_e$。在本实验中，错误率阈值取2，高频词汇的阈值取10，并采用模型1作为 W_t 选取的训练模型。实验结果表明，模型7是表现最好的模型，其中 $|W_t|$ 为增益词的个数。

表2 模型7的汉语组块识别测试结果

W_t	$\|W_t\|$	Precision	Recall	$F_{\beta=1}$
$W_t=W_e$	28	84.91%	79.38%	82.05
$W_t=W_c$	73	83.64%	80.36%	81.97
$W_t=W_e\cup W_c$	93	84.51%	80.36%	82.38

3.2 英语组块识别

由于中文组块识别没有统一的标准语料，无法进行客观的比较，所以我们实验了英语的组块识别。目的是通过比较来确定我们的方法是否正确有效，是否语言无关。本文采用 CoNLL-2000共享任务的训练语料和测试语料作为实验用语料。该语料来源于 Penn Treebank II，是经过人工标注的带句法信息的熟语料，将其转换成带 Chunk 标记的语料。目前，识别的组块类型包括：ADJP，ADVP，CONJP，INTJ，LST，NP，PP，PRT，SBAR，VP，它们的定义参考（Ramshaw and Marcus，1995；Erik F，and Sabine Buchholz，2000）。本文将实验分为两组来进行，第一组实验考察了模型1-6的组块识别结果，第二组实验考察了模型7的组块识别结果。

3.2.1 *模型1-6英语组块识别结果* 见表3。

表3 模型1-6英语组块识别结果（$F_{\beta=1}$），最好的结果用粗体表示

组块类型	模型1	模型2	模型3	模型4	模型5	模型6
All	**89.57**	88.12	89.35	84.33	83.09	85.79
ADJP	**66.35**	59.03	60.21	55.91	50.53	54.31
ADVP	73.51	74.45	**76.02**	69.55	69.49	71.85
CONJP	42.86	64.00	**66.67**	0.00	48.28	50.00
INTJ	33.33	33.33	28.57	50.00	**80.00**	80.00
NP	**91.56**	87.77	88.83	86.09	82.47	85.04
PP	91.90	95.89	**96.02**	88.37	93.72	94.16
PRT	32.93	72.03	**72.34**	30.14	66.67	69.71
SBAR	45.53	80.42	**82.23**	11.46	75.89	78.23
VP	**91.58**	87.75	90.46	85.66	80.61	86.01

由实验3结果可以发现：

1）模型4，5，6的输出集合均为 c_i，过于一般化，结果不是很好，这也与当初的预想是吻合的。

2）对于不同类型的组块，不同转换函数训练出来的模型得到了不同的结果。其中，最好的结果主要集中在模型1和模型3上。

3）如果融合上述各种模型的优势之处，则结果还会提高。于是，又作了实验4。

3.2.2 *模型7的英语组块识别结果* 由实验3发现，最好的结果也集中在模型1和3。于是，本文又测试了模型7，其中 W_t 集合的选取过程与上述大致一样，除了 W_e 是训练集中属于几种特定的 chunk 类型的高频词汇，包括 ADVP，CONJP，PP，PRT，SBAR 等。实验结果表明，模型7也是表现最好的模型。

（下转第192页）

Ont.; Software Reliability Engineering. 1995. In: Proc. Sixth Intl. Symposium on 10/24/1995 -10/27/1995. Oct. 1995

3 Savor T. Seviora R E. An architectural overview of a software supervisor. Bell Canada Software Reliability Lab. Waterloo Univ. Ont.; Real-Time Systems, 1996. In: Proc. of the Eighth Euromicro Workshop on 06/12/1996 -06/14/1996, Jun 1996

4 梁冰. 李磊. 动作推导引擎及其在通信软件设计中的应用计算机工程与应用(录用)

5 Savor T. Seviora R E. An approach to automatic detection of software failures in real-time systems. Bell Canada Software Reliability Lab. Waterloo Univ. Ont.; Real-Time Technology and Applications Symposium. 1997. In: Proc. Third IEEE 06/09/1997 -06/11/1997. Jun 1997

6 Alagar V S, et al. Specification-based Testing for Real-Time Reactive Systems. Dept. of Comput. Sci., Concordia Univ., Montreal. Que.; Technology of Object-Oriented Languages and Systems, 2000. TOOLS 34. In: Proc. 34th Intl. Conf. on 07/30/2000 -08/04/2000. 2000

7 Brown D B, et al. An automated oracle for software test-ing. Reliability. IEEE Transactions, 1992. 41(2)

8 吴今培, 肖健华. 智能故障诊断与专家系统. 科学出版社

(上接第154页)

表4给出了一组相关文献报道的实验结果,与已有的最好的一批系统相比,本文组块识别的性能极为接近目前的最优水平。值得一提的事,比本文稍优的系统,使用了更为复杂的多种机器学习组合的算法,结果和本文的结果没有明显的差别。而本文通过导入各种上下文信息到隐马尔科夫模型,达到了所有使用单一方法的系统的最佳结果。

表4 CoNLL-2000共享任务的测试结果

实验	Precision	Recall	$F_{\beta=1}$
文[12]	93.45%	93.51%	93.48
文[3]	94.04%	91.00%	92.50
实验4	92.05%	92.46%	92.25
文[14]	91.99%	92.25%	92.12
文[11]	92.08%	91.86%	91.97
文[16]	91.05%	92.03%	91.54
文[15]	90.63%	89.65%	90.14
实验3	89.58%	89.55%	89.57
文[18]	86.24%	88.25%	87.23
文[17]	88.82%	82.91%	85.76

结论 本文提出了一种基于增益的隐马尔科夫模型的文本组块分析技术,该方法通过不断增益的转换函数对训练语料进行不断优化,从而达到训练模型的不断增益的过程。该方法不需要修改原来模型的训练过程和标注过程,从而保留了原有模型的训练和标注的良好的性能,并增加了输出的标记集合,给出了更加完善的上下文模型;另一方面,实验的训练过程和标注过程都是在秒级单位内完成,这也是该模型优于其他模型的地方,比如最大熵模型、支持向量机模型等。从实验结果中发现,中文比英文的结果低10个百分点左右。主要原因可能是中文的语法结构比较灵活,存在大量的结构歧义。其次,从上面的实验也可以看出,更多的训练数据也可提高识别效果。最后,使用更多的知识能提高实验效果。下一步的工作通过导入更多的上下文信息,包括短语边界,语义,搭配和共现等信息,以提高识别的准确率和召回率。

参考文献

1 Abney S. Parsing by chunk. In Berwick, A. and Tenny, editors, Principle-Based Parsing. Kluwer, 1991

2 Erik F. Tjong Kim Sang and Sabine Buchholz Introduction to the CoNLL-2000 Shared Task: Chunking. CoNLL-2000 and LLL-2000. Lisbon, Portugal, pp. 127～132

3 Erik F, Sang T K. Text chunking by system combination. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, 2000

4 Brants T. TnT -a statistical part-of-speech tagger. In: Proc. of the Sixth Applied Natural Language Processing (ANLP-2000), Seattle, WA, 2000

5 Ramshaw L, Marcus M. Text Chunking Using Transformation-Based Learning. In: Proc. of third Workshop on Very Large Corpora, June 1995. 82～94

6 Ratnaparkhi A. Maximum Entropy Models for Natural Language Ambiguity Resolution: [Phd. Thesis]. University of Pennsylvania, 1998

7 Merialdo B. Tagging English Text with a Probabilistic Mod-el. Computational Linguistics, 1994, 20(2): 155～171

8 Church K W. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proc. of the 1st Conf. on Applied Natural Language Processing, ANLP, ACL, 1988. 136～143

9 Daelemans W, Buchholz S, Veenstra J. Memory-Based Shallow Parsing. In: Proc. of EMNLP/VLC-99, University of Maryland, USA, June 1999. 239～246

10 Argamon S, Dagan I, Krymolowski Y. A Memory-based Approach to Learning Shallow Natural Language Patterns. In: Proc. of the joint 17th Intl. Conf. on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL, Montreal, Canada, 1998. 67～73

11 Koeling R. Chunking with Maximum Entropy Models. In: Proc. of CoNLL-2000 and LLL-2000, Lisbon, Portugal, Sep. 2000

12 Kudo T, Matsumoto Y. Chunking with Support Vector Machines. In: Proc. of NAACL 2001, Pittsburgh, USA, Morgan Kaufman Publishers, 2001

13 Kudo T, Matsumoto Y. Use of Support Vector Learning for Chunk Identification. In: Proc. of CoNLL-2000 and LLL-2000, Lisbon, Portugal, Sep. 2000

14 Zhou G D, Su J, Tey T G. Hybrid Text Chunking. In: Proc. of CoNLL-2000 and LLL-2000, Lisbon, Portugal, Sep. 2000

15 Pla F, Molina A, Prieto N. Improving chunking by means of lexical-contextual information in statistical language models. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, 2000

16 Veenstra J, van den Bosch A. Single-classifier memory-based phrase chunking. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, 2000

17 Vilain M, Day D. Phrase parsing with rule sequence processors: an application to the share conll task. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, 2000

18 Johansson C. A context sensitive maximum likelihood approach to chunking. In: Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, 2000