中文分词和词性标注的在线重排序方法

孟凡东　　谢军　　刘群
中国科学院 计算技术研究所
智能信息处理重点实验室，北京 100190
{mengfandong,xiejun,liuqun}@ict.ac.cn

摘要

当前主流的中文分词与词性标注方法将分词和词性标注问题看成是序列标注问题，通常利用局部特征训练判别式模型。该方法取得了很好的效果，但是与词、词性相关的全局特征并没有被充分的利用。为了更好的处理分词和词性标注的歧义，传统的重排序方法在第一次解码的 n-best 候选结果集上，利用全局特征进行二次解码，重新选择一个更好的结果。该方法往往需要保留较大的候选结果集，并需要两次解码。本文提出了一种在线重排序方法，将重排序过程融合到一次解码的过程中，充分利用局部和全局特征，在一次解码时利用更多信息以减少搜索错误，选择一个更好的结果。本文在中文宾州树库(CTB5.0)和微软亚洲研究院语料(MSR)上做实验，结果表明,本方法相对于只用局部特征训练的基线系统以及传统的重排序方法都有明显的效果提升。

1　引言

Xue and Shen (2003) 首先提出将分词问题转化为基于字的序列标注问题，当前主流的中文分词方法基本上采用这个思想，利用最大熵 (Ratnaparkhi and Adwait, 1996)、 条件随机场 (Lafferty et al., 2001)、感知机算法 (Collins, 2002) 等训练判别式模型。相对于生成式的方法 (Rabiner, 1989; Fine et al., 1998)，判别式方法处理未登录词的能力更强。Ng and Low (2004) 进一步提出联合分词与词性标注的方法，将分词和词性标注融合在统一的框架下，以词性作为特征，增大解码空间，结果比单独分词、词性标注的基线系统效果都好。

基于字标注的分词方法，通常使用的是局部特征。局部特征是在一定长度的窗口范围内抽取字的上下文信息，距离该字较远的信息难以得到充分的利用。虽然只利用局部特征已经可以取得很好的结果，但是引入全局特征可以进一步增强处理歧义的能力，对于分词与词性标注来说是有帮助的。

通常使用全局特征的方法是重排序方法。即第一次利用局部特征训练分类器进行解码，保留 n-best 候选结果表；然后利用重排序技术进行第二次解码，在这 n-best 候选结果列表里重新选择出最好的结果。这种方法在一定程度上提高了分词、词性标注的效果，但是往往需要在第一次解码时保留较大的 n-best 列表，才能找到真正的最优解。

本文提出中文分词与词性标注的在线重排序方法，将分词解码过程与重排序过程融合在一个框架下，在充分利用传统的局部特征的基础上，补充利用全局特征。利用堆栈搜索算法解码。我们为每个字保留一个堆栈，存储从第一个字到当前字为止的最好的候选结果集，以供在线重排序使用。相对于传统的重排序方法，本方法旨在一次解码过程中利用更多的信息尽量避免错误，以便搜索到更好的结果。本方法只需要为每个字保留一个很小的堆栈，效果就有明显的提升。

本文在CTB5.0和MSR语料上做实验，实验结果表明，本方法相对于仅用局部特征训练的基线系统分词和词性标注错误率均有明显下降。相对于只用局部特征的基线系统， CTB和MSR语料上的的分词错误率分别下降11.57%和10.86%。CTB的联合分词与词性标注错误率下降为5.65%。

本文与 Jiang et al. (2008)进行对比，我们使

用同样的语料和特征模板。在分词结果上，本方法可以达到传统重排序方法在nbest-100上做重排序的效果，略低于在压缩词图上做重排序的效果。在联合分词与词性标注结果上，本方法超过传统的重排序方法，相对于在nbest-100上做重排序的结果，本方法再次提高0.3个百分点，错误率再次下降4.32%，并且与在压缩词图上做重排序的方法性能相当。

接下来，我们在第2节介绍字标注分词和词性标注方法，第3节详细阐述在线重排序方法，第4节列举相关工作，第5节是本文实验及结果分析，第6节是对本文的总结与展望。

2 字标注的中文分词与词性标注方法

本文采用字标注的判别式的分词与词性标注方法。Xue and Shen (2003) 将分词问题转化为字符 (汉字) 分类问题。根据 Ng and Low (2004)，采用四种位置标记表示汉字在词中的相对位置，“b”表示词首，“m”表示词中，“e”表示词尾，“s”表示单字词。即一个词只可以被标记成“s” (单字词) 或“bm*e”(多字词)。分词和词性标注任务可以用统一的框架表示 (Ng and Low, 2004)。联合分词与词性标注就是对于每个字，有位置标记和词性标记，例如“e_v”，表示一个动词的词尾。

2.1 分词与词性标注特征模板

根据 Ng and Low (2004)，用C_0表示当前的汉字，C_{-i}表示C_0左边第i个汉字，C_i表示 C_0右边第i个汉字。$Pu(C_i)$用于判断当前汉字C_i是否为分隔符(是就返回 1，否则返回 0)。$T(C_i)$用于判断当前汉字C_i的类别：数字，日期，英文字母，和其它（分别返回 1, 2, 3 和 4）。

序号	特征模板
1	$C_i (i = -2...2)$
2	$C_i C_{i+1} (i = -2...1)$
3	$C_{-1} C_1$
4	$Pu(C_0)$
5	$T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$

表 1 分词和词性标注的局部特征模板

表 1 描述了分词和词性标注的局部特征模板，假设当前分析的是“450 公里”中的“0”字，特征模板生成的特征如下：

1) $C_{-2} = 4$， $C_{-1} = 5$， $C_0 = 0$， $C_1 =$公， $C_2 =$里；
2) $C_{-2}C_{-1} = 45$， $C_{-1}C_0 = 50$， $C_0C_1 = 0$公， $C_1C_2 =$公里；
3) $C_{-1}C_1 = 5$公；
4) $Pu(C_0) = 0$；
5) $T(C_{-2})\,T(C_{-1})\,T(C_0)\,T(C_1)\,T(C_2) = 11144$。

算法 1 感知机训练算法

Input: Training examples (x_i, y_i)
$\vec{\alpha} \leftarrow 0$
for $t \leftarrow 1 \cdots T$ **do**
  for $i \leftarrow 1 \cdots N$ **do**
    $z_i \leftarrow \arg\max_{z \in GEN(x_i)} \Phi(x_i, z) \cdot \vec{\alpha}$
    if $z_i \neq y_i$ **then**
      $\vec{\alpha} \leftarrow \vec{\alpha} + \Phi(x_i, y_i) - \Phi(x_i, z_i)$
Output: Parameters $\vec{\alpha}$

2.2 训练算法

本文采用Collins (2002) 的平均感知机训练算法，训练分词与词性标注分类器。算法1 描述了感知机训练算法。我们采用“平均参数”技术来避免过拟合。训练的过程就是学习一个从输入$x \in X$ 映射到输出$y \in Y$ 的判别模型，X是训练语料中的句子集合，Y是相应的标记结果。Jiang et al. (2009) 中使用了$GEN(x)$函数列举输入x的所有候选结果，表示每个训练实例$(x, y) \in X \times Y$ 映射到特征向量 $\Phi(x, y) \in R^d$，对于一个特征向量，$\vec{\alpha} \in R^d$是与其对应的参数向量。对于一个输入的汉字串x,目的是找到一个满足下式的输出结果$F(x)$：

$$F(x) = \arg\max_{y \in GEN(x)} \Phi(x, y) \cdot \vec{\alpha} \qquad (1)$$

其中$\Phi(x, y) \cdot \vec{\alpha}$表示特征向量$\Phi(x, y)$和参数向量的内积。本文沿用此方法。

3 在线重排序方法

在线重排序方法利用的特征包括两部分，一部分是局部的字特征，另一部分是全局的词、词性特征。全局特征分数的计算方式与传统的重排序方法类似，因此，本节首先介绍传统的重排序方法，再介绍在线重排序方法。

3.1 传统的重排序方法

对于句子s的 n-best 候选结果$cand(s)$，重排序是从$cand(s)$中选择最好的结果$\hat{y}$:

$$\hat{y} = \arg\max_{y \in cand(s)} w \cdot f(y) \qquad (2)$$

$w \cdot f(y)$ 是特征向量 f 和权重向量 w 的点积，点积的结果用于对候选结果 $cand(s)$ 重排序。重排序特征由两部分组成，第一部分是基线分类器输出的结果分数，是实数值；第二部分是与词、词性相关的全局特征。

方法			全局特征模板	说明
联合分词与词性标注	分词	1	W_0	当前词
		2	$W_{-1}W_0$	前一个词与当前词
		3	$S(W_0)$	当前词是单字节词
		4	W_0T_0	当前词--词性对
		5	W_{-1}	当前词的前一个词
		6	T_{-1}	当前词的前一个词性
		7	$T_{-2}T_{-1}$	当前词的前两个词性
		8	$T_{-3}T_{-2}T_{-1}$	当前词的前三个词性

表 2 分词与词性标注的全局特征模板

3.2 在线重排序方法

本文提出的在线重排序方法的基本思想是利用局部字特征和全局特征共同作用，完成分词与词性标注的解码过程。解码时，为每个字维护一个堆栈，用来存储从第一个字到当前字为止的候选结果集，利用这些结果计算局部特征和全局特征分数，根据“局部特征+全局特征”的总分数进行重排序。解码的过程实际上就是为每个字构建从第一个字到当前字为止字序列的*候选结果表* 的过程。算法2 是在线重排序解码算法，详细描述了联合分词与词性标注的在线重排序解码过程。

3-16行考虑到了字序列 $C_{1:n}$ 中的每个汉字 C_i，$cands$ 用于存储从第一个字到当前汉字为止的字序列的分词与词性标注的候选结果集，例如 $cands[i]$ 表示从第一个汉字到第 i 个汉字为止的字序列的分词与词性标注的候选结果集。第5行枚举了所有以字 C_i 结尾的候选词 w，w 的长度不超过 K，本文的实验中 K 为15，即最长词的长度是15个字。第7行枚举了词语 w 所有的词性标记 t，POS 表示词性标记集。第8行中的 p 表示词-词性标记对 $< w,t >$。第10行枚举了从第1个字到第 $i-l$ 个字 C_{i-l} 为止的候选分词与词性标注候选结果 $d \in Cand[i-l]$，d 与 p 组合成新的结果 p_{new}，p_{new} 就是从第1个字到第 i 字为止的一个候选分词与词性标注结果。构建 C_i 对应的候选结果表 $cands[i]$ 时，由 $cands[i-1]$ 与 C_i 组合生成一定数量的 $C_{1:i}$ 的候选结果，再由 $cands[i-2]$ 与 $C_{i-1}C_i$ 组合生成一定数量的 $C_{1:i}$ 的候选结果，按此依次生成所有 $C_{1:i}$ 的候选结果存入 $cands[i]$ 中。第12和13行计算 p_{new} 的“局部特征”得分和“局部特征+全局特征”的总得分。第15行是将这个新结果存储到字 C_i 对应的存储候选结果表 $cands[i]$ 中。第16行将 $cands[i]$ 中的结果根据特征总得分 s 从大到小排序。第17行得到最后的结果，即最后一个字的候选结果表 $cands[n]$ 中得分最高的结果 $cands[n][0]$。

算法 2 在线重排序解码算法

1: **Input:** Character Sequence $C_{1:n}$
2: $cands[1...n] \leftarrow \varnothing$
3: **for** $i \leftarrow 1...n$ **do**
4: $cands[i] \leftarrow \varnothing$
5: **for** $l \leftarrow 1 \cdots \min(i, K)$ **do**
6: $w = C_{i-l+1:i}$
7: **for** $t \in POS$ **do**
8: $p \leftarrow < w, t >$
9: $p \cdot score_{local} \leftarrow Eval_{local}(p)$
10: **for** $d \in cands[i-l]$ **do**
11: $p_{new} \leftarrow d + p$
12: $p_{new} \cdot score_{local} \leftarrow p \cdot score_{local} + d \cdot score_{local}$
13: $p_{new} \cdot score \leftarrow Eval_{global}(p_{new}) + p_{new} \cdot score_{local}$
14: $s \leftarrow p_{new} \cdot score$
15: $cands[i] \leftarrow cands[i] \cup (p_{new}, s)$
16: **sort** $cands[i]$ **according to** s
17: $r^* \leftarrow cands[n][0]$
18: **Output:** Best Character Sequence r^*

对于只做分词任务的在线重排序，稍微修改一下算法 2 即可实现。即，不必枚举词 w 的词性标记 (第 7 行)，$cands[i]$ 中存储的也只是分词的结果。

我们采用平均感知机算法训练特征的权重。只用局部特征训练分类器时，我们用感知机训练，更新权重时步长一般设为 1。对于在线重排序方法，我们利用了局部特征和全局特征，我们更新权重时，局部特征的更新步长设置为 1，全局特征权重的更新步长设为 0.5 可以取得更好的结果。

3.3 剪枝策略

在线重排序的解码过程中，我们采用了阈值剪枝与直方图剪枝策略。构造每个字C_i的候选结果表$cands[i]$时，$cands[i]$中结果与最好结果相差大于一个阈值T时，这种结果会被去掉。$cands[i]$的容量设置为B，$cands[i]$中的结果排序后只保留特征分数排在前B的结果，排在后面的结果会被去掉。采用剪枝策略以提前排除较差的结果，在不损失结果性能（或者损失较少）的前提下，提高解码效率。实验中我们使用的阈值T为 1000，并通过实验验证，选择B为 20 比较合适。

3.4 全局特征模板

我们使用的全局特征模板如表 2 所示。表 2 中 1-3 特征模板是只做分词任务所使用的特征模板，1-8 特征模板在联合分词与词性标注的方法中使用，其中 4-8 特征模板是 Jiang et al. (2008)中使用的，本文沿用此模板。这里面所有的特征模板与表 1 中的不同，它们都是与词、词性相关的全局特征。为了方便比较，所有的全局特征模板都不包含“将来”的词或词性。我们考虑当前的词-词性对时，只使用当前的或者历史的词、词性信息 (Jiang et al., 2008)。

4 相关工作

本文提出的中文分词与词性标注的在线重排序方法是建立在传统的字标注方法基础上的。Xue and Shen (2003) 首先提出将分词问题转化为序列标注问题，Ng and Low (2004) 提出了联合分词与词性标注的框架，并验证了这种联合的方法相比分开做分词和词性标注的方法可以获得更好的精度。

近几年，分词任务中又出现了很多新的工作。 Zhang and Clark (2007) 的基于词模型的感知机训练算法，在改善分词歧义方面取得了很好的效果。Jiang et al. (2008) 提出了在压缩词图上做分词与词性标注重排序的方法，验证了在压缩词图上做重排序可以在多项式的时间内搜索到指数级的候选结果，并且在精度上比传统的重排序方法有较大的提高。 Li and Sun (2009) 利用分隔符作为暗示，并利用大量网络语料增强未登录词的识别能力。Wang et al. (2010) 的联合生成式与判别式模型，Li (2011) 分析词语内部结构，Sun (2011) 的堆栈子词模型以及 Sun and Xu (2011) 利用未标注数据信息在改进分词效果方面做了很好的工作。标注迁移方法(Jiang et al., 2009, Jiang et al., 2012)，学习不同标注标准的语料知识，帮助改进分词性能。Sun et al. (2012) 的联合分词与新词发现，二者互相帮助，改进分词精度。

本文提出的中文分词与词性标注在线重排序方法与 Zhang and Clark (2007)的词模型框架相似，但是解码时栈中元素的生成方式是不同的。Zhang and Clark (2007) 的解码过程是对于当前要处理的字，判断这个字与前面的结果应该合并为一个词，还是作为一个新词的开始，利用其独有的特征模板，囊括了字特征与词特征，训练模型。而本方法，是显式地利用已有的基于字标注的模型，并扩展该模型，在解码过程中引入全局特征，对既有结果做重排序。我们将重排序在线地融合在一次解码的过程中，在传统局部特征的基础上补充利用全局特征，在一次解码过程中利用更多信息减少搜索错误，以进一步提高分词、词性标注的精度。

5 实验及结果分析

5.1 设置

我们采用中文宾州树库语料(CTB5.0)和微软亚洲研究院语料(MSR)做分词和联合分词与词性标注实验，验证在线重排序方法的效果。

对于 CTB5.0 (以下简称 CTB)，我们用 1-270 章作为训练集(18074 句)，271－300 章作为测试集(348 句)，301-325 章作为开发集(350 句)。对于 MSR 语料，训练集有 86924 句，测试集有 3985 句。这两种语料具有不同的特点，CTB 语料中词的粒度适中， MSR 语料中词语

普遍偏长。本文在不同词语粒度的语料上验证本方法的有效性。

本文采用 F-measure 来评价分词和词性标注结果。$F = 2PR/(P+R)$，其中 P 是分词准确率，R 是召回率。

5.2 字标注分类器的性能

我们利用平均感知机算法，按照表 1 所示的字局部特征模板训练分词分类器和联合分词与词性标注的分类器，构造基线系统，探测字局部特征训练出的分类器的性能。

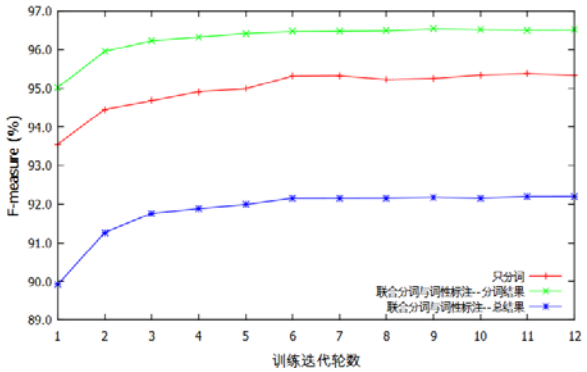

图 1 平均感知机迭代训练的学习曲线

首先，我们用 CTB 的训练集和开发集探测平均感知机迭代训练的学习曲线。学习曲线如图 1 所示，无论是分词还是联合分词与词性标注，训练迭代至第 7 轮就已经收敛。因此后面的实验，我们都采用第 7 轮训练出的模型。

语料	分词	联合分词和词性标注 (F_1 %)	
		不考虑词性	考虑词性
CTB	97.30	97.77	93.10
MSR	96.12	--	--

表 3 根据局部字特征训练的分类器的结果，“--”表示没有相应的测试结果

从表 3 的结果可以看出，仅利用局部字特征，就可以取得较好的分词、词性标注结果，其中 CTB 上的分词 F_1 值在 97%以上。在 CTB 上，利用联合分词与词性标注的方法，分词的 F_1 值比单独分词的结果要高，提高了 0.47 个百分点，错误率下降 17.4%。由于 MSR 语料没有词性标记信息，因此没有在这种语料上做联合分词与词性标注实验。

5.3 在线重排序实验

在线重排序的解码过程中，我们为每个字 C_i 维护一个从第一个字到 C_i 为止的候选结果列表 $cands[i]$，为了提早将较差的结果剪枝而提高解码效率，$cands[i]$ 的大小 B 要做限制。我们用 CTB 做实验，验证不同的 B 在 CTB 测试集上的效果。表 4 列出了不同的 B 的对解码结果的影响。

B	分词 (F_1 %)	联合分词与词性标注 (F_1 %)	
		不考虑词性	考虑词性
5	97.37	97.58	93.27
10	97.39	97.60	93.39
20	97.58	97.77	93.49
50	97.58	97.78	93.50
100	97.58	97.78	93.50

表 4 B 从 5 到 100，在线重排序算法在 CTB 测试集上的结果

从表 4 的结果可以看出，无论是分词还是联合分词与词性标注，不同的 B 对结果是有影响的。对于分词，当 B 较小时 (5，10，20)，随着 B 的增大，分词精度不断提高。B 为 20 时，相对于只用局部特征的基线系统 (97.30%)，结果 (97.58%) 已经有明显提高，F_1 值提高 0.28 个百分点，错误率下降 10.37%。当 B 超过 20 (50，100)时，分词结果变化不大。对于联合分词与词性标注，当 B 达到 20 时，虽然分词的结果相对于基线系统并没有增长，但是联合分词与词性标注的结果提高了 0.39 个百分点，错误率下降 5.65%，当 B 超过 20 时，分词和词性标注结果已经变化不大。

语料	分词 (F_1 %)	联合分词与词性标注 (F_1 %)	
		不考虑词性	考虑词性
CTB	97.58/**0.28** ↑	97.77	93.49/**0.39** ↑
MSR	96.50/**0.38** ↑	--	--

表 5 本方法与只用局部特征的基线系统的结果对比

表 5 是本方法与只用局部特征的基线系统的结果对比，其中 B 设为 20，“/”前面的是本方法的结果，后面的是本方法相对于基线系统 F_1 值提高。从表 5 的结果可以看出，相对于

基线系统，本方法在 CTB 和 MSR 语料上的 F_1 值都有明显提高。对于分词，本方法在 CTB 和 MSR 语料上的 F_1 值分别提高 0.28 和 0.38 个百分点，错误率下降分别为 11.57%和 10.86%。对于联合分词与词性标注，虽然本方法在 CTB 语料上的分词 F_1 值没有提高，但是在联合分词与词性标注的结果上提高了 0.39 百分点，错误率下降 5.65%。在联合分词与词性标注方法上，词性也作为特征帮助解码，基线系统的分词 F_1 值已经很高 (97.77%), 在不利用其他信息的情况下，分词效果很难在此基础上有明显提升。无论在分词还是联合分词与词性标注的实验结果上，F_1 值的提升，验证了本方法的有效性。

方法		联合分词与词性标注 (F_1 %)	
		不考虑词性	考虑词性
本方法		97.55	93.35
Jiang et al. (2008)	nbest-20	97.49	92.80
	nbest-50	97.51	93.02
	nbest-100	97.55	93.05
	lattice-2	97.53	92.96
	lattice-5	97.74	93.36
	lattice-10	97.74	93.37

表 6 本方法与 Jiang et al. (2008) 的结果对比

表 6 是本方法与 Jiang et al. (2008) 的结果对比，其中 B 设为 20，“nbest-”表示传统的在重新 nbest 结果候选结果上做重排序排序方法，“lattice-”表示在压缩词图上做重排序的方法。例如，“nbest-20”表示在 20 个最好的候选结果上做重排序，“lattice-5”表示压缩词图的入度剪枝数为 5，即对于每个词图上的节点保留最多 5 个入度的分支。为了方便比较结果，表 6 中本方法(在线重排序方法)采用与 Jiang et al. (2008) 中相同的局部以及全局特征模板，即不利用外部信息，如数词、时间词、字符串等。从结果可以看出，本方法在分词结果上可以达到传统重排序方法在 nbest-100 上做重排序的效果，比在压缩词图上做重排序的方法略低。在联合分词与词性标注结果上，本方法超过传统的重排序方法，相对于在 nbest-100 上做重排序的结果，本方法再次提高 0.3 个百分点 (从 93.05 提升至 93.35)，错误率再次下降 4.32%，并且与在压缩词图上做重排序 (Jiang et al., 2008) 的方法性能相当。

6 总结与展望

本文提出了一种中文分词与词性标注的在线重排序方法，将解码过程与重排序融合在一个框架下，充分利用局部特征和全局特征，在更大的解码空间下搜索最优解。相对于传统的重排序方法，本方法旨在一次解码的过程中利用更多信息，尽量避免错误，以提升效果。本文在CTB5.0 和MSR语料上做实验，实验结果表明，本方法相对于仅用局部特征训练的基线系统分词和词性标注错误率均有明显下降。当 B 为 20时，相对于只用局部特征的基线系统，CTB和MSR的分词错误率分别下降11.57%和10.86%。CTB的联合分词与词性标注错误率下降为5.65%。

本文与 Jiang et al. (2008) 的方法进行了对比，我们使用同样的语料和特征模板。在分词结果上，本方法可以达到传统重排序方法在nbest-100上做重排序的水平，略低于在压缩词图上做重排序的效果。在联合分词与词性标注结果上，本方法超过传统的重排序方法，相对于在nbest-100上做重排序的结果，本方法再次提高0.3个百分点，错误率再次下降4.32%，并且与在压缩词图上做重排序 (Jiang et al., 2008) 的方法性能相当。

将来，我们会继续研究减少分词与词性标注搜索错误的方法，以期进一步降低分词、词性标注错误率。

参考文献

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 18, Philadelphia, USA.

Fine, Shai, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden markov model: Analysis and applications. In *Machine Learning*, pages 32–41.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese partofspeech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of*

the Empirical Methods in Natural Language Processing Conference.

Rabiner, Lawrence. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE,* pages 257–286.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics*, 35:505512, December.

Zhongguo Li. 2011. Parsing the internal structure of words: A new paradigm for chinese word segmentation. In *Proceedings of the 49th Annual Meeting of the Association for Computational* Linguistics.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging: a case study. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics.*

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008. Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 385–392.

Wenbin Jiang, Fandong Meng, Qun Liu, and Yajuan Lü. 2012. Iterative annotation transformation with predict-self reestimation for chinese word segmentation. In *Proceedings of EMNLP*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th ICML,* pages 282289, Massachusetts, USA.

Ratnaparkhi and Adwait. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference.*

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of EMNLP*.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part of speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.*

Xu Sun, Houfeng Wang, Wenjie Li. 2012. Fast Online Training with Frequency-Adaptive Learning Rates for Chinese Word Segmentation and NewWord Detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.*

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character based joint model for Chinese word segmentation. In *Proceedings of COLING.*

Nianwen Xue and Libin Shen. 2003. Chinese wordsegmentation as lmr tagging. In *Proceedings of SIGHAN Workshop*.

Nianwen Xue, Fei Xia, FuDong Chiou, and Martha Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. In *Natural Language Engineering.*

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a wordbased perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics.*