一种自适应词性标注方法*

王 挺 陈火旺 杨 谊 史晓东

(国防科技大学计算机系 长沙 410073)

摘要 本文针对词性标注的问题,修改了经典的隐马尔可夫模型 HMM(hidden Markov model)参数估算方法,使得模型参数能够随着新语料的增加而动态地进行调整,从已标注或未标注的语料中获取知识以提高模型的准确性.

关键词 语料,词性标注,隐马尔可夫模型.

中图法分类号 TP391

所谓词性标注就是根据句子的上下文的信息给句中的每个词一个正确的标记,这些标记都具有特定的语言学意义.一般来说,词性标住有两种方法可供考虑:一种是基于规则的方法,另一种是基于统计的方法.作为后一种方法的代表性工作,Merialdo 使用隐马尔可夫模型 HMM(hidden Markov model)进行词性标注并得到了较好的结果.[1]我们注意到 Merialdo完成的只是用语料对模型参数进行一次性训练,并未考虑当新的训练语料不断引入时如何动态修改模型,使之与新的语料相适应.在实际情况下,我们很难一次得到一个巨大的、包括所有知识的语料库,并以此通过训练得到一个永远不变的模型.通常训练语料的收集是一个逐步积累的过程,新的语料会不断出现,我们必须随着训练语料的扩充,逐步地调整模型的参数使其适应性不断提高.因此,如何根据新的训练语料动态地修改模型的参数是一个重要问题.根据 HMM 理论,我们可以使用消去内插方法(Deleted Interpolated),通过合并原模型和从新语料中训练得到模型来获得一个新的模型,但是这种方法不仅非常复杂,而且新模型的好坏极大地依赖使用者的经验,因此在实际情况下并不十分可靠,难以令人满意.[2]

本文修改了 HMM 及其估算方法,使得模型的参数能方便地随着新训练语料的不断引入而得到逐步的调整.使用改进后的方法,当我们面对新的语料时,可以根据语料的加工情况分别采用两种手段来修改模型的参数:一种是使用频率统计方法(如果语料是已标注的),一种是利用著名的 Baum-Welch 估算方法(如果语料是未标注的).实验结果显示,本文提出的方法具有良好的自适应性.

* 本文研究得到国家 863 高科技项目和校预研项目基金资助.作者**王挺**,1970 年生,博士生,主要研究领域为计算语言学,机器翻译.**陈火旺**,1936 年生,教授,博士导师,主要研究领域为计算机软件和人工智能.**杨谊**,女,1973 年生,硕士生,主要研究领域为计算语言学.**史晓东**,1966 年生,博士,主要研究领域为机器翻译和计算语言学.

本文通讯联系人:王挺,长沙 410073,国防科技大学计算机系

本文 1997-01-13 收到修改稿

1 使用 HMM 的词性标注方法

一个隐马尔可夫模型 HMM 可以描述为：

(1) N，模型的状态的数目. 模型的状态及它们所组成的状态集表示为 $S=\{S_1,S_2,\ldots,S_N\}$，在 t 时刻的状态用 q_t 表示. 在词性标注这一具体问题中，状态就是词性标记，如 *Ns*，*Ving*，*ADJer* 等.

(2) M，每个状态上对应的可能的观察值的数目. 我们记这些观察为 $W=\{w_1,w_2,\ldots,w_M\}$. 在词性标注问题中，观察值就是语料中的单词，W 即为语料生成的词典.

(3) 状态转移概率分布矩阵 $A=\{a_{ij}\}$，其中

$$a_{ij}=P(q_{t+1}=S_j|q_t=S_i),1\leqslant i,j\leqslant N \tag{1}$$

在词性标注问题中，A 是标记之间的转移概率矩阵.

(4) 观察值概率分布矩阵 $B=\{b_j(k)\}$，其中

$$b_j(k)=P(\text{在 } t \text{ 时刻出现 } w_k|q_t=S_j),1\leqslant j\leqslant N,1\leqslant k\leqslant M \tag{2}$$

在词性标注问题中，B 是单词概率分布矩阵.

(5) 初始状态分布矢量 $\pi=\{\pi_i\}$，其中

$$\pi_i=P(q_1=S_i),1\leqslant i\leqslant N \tag{3}$$

在 HMM 的框架下，语料的词性标注可以视为：给定观察值序列(即单词序列，亦即句子)，试图找到它的状态序列(即该句的标记序列). 著名的 Viterbi 算法可以用来找到每个句子的最为可能的标记序列.[1]

如何训练参数以获得一个较好的模型是 HMM 研究的重要问题之一，也是提高词性标注准确率的关键. 经典的 Baum-Welch 重估算方法能够使用观察值序列来训练模型，因此使用这种方法，可以用未标注的原始语料(原始句子)来训练模型，这为我们节约了大量的手工标注语料的工作，提供了非常有效的训练手段. 该算法可以简单地描述如下：

给定观察值序列 $O=O_1,O_2,\ldots,O_T$ 以及一个 *HMM* $\lambda=(\pi,A,B)$，考虑前向变量(forward variable)$\alpha_t(i)$，其定义为

$$\alpha_t(i)=P(O_1,O_2,\ldots,O_T,q_t=S_i|\lambda) \tag{4}$$

即它是给定模型 λ，出现部分观察值序列 $O_1,O_2,\ldots,O_t$ 并且在 t 时刻处于状态 S_i 的概率.

类似地，我们可以定义后向变量(backward variable)$\beta_t(i)$如下

$$\beta_t(i)=P(O_{t+1},O_{t+2},\ldots,O_T|q_t=S_i,\lambda) \tag{5}$$

即它是给定模型 λ，在 t 时刻处于状态 S_i，并且出现部分观察值序列 $O_{t+1},O_{t+2},\ldots,O_T$ 的概率. $\alpha_t(i)$和 $\beta_t(i)$可以采用 Forward-Backward 算法递归地计算出来. 在计算过程中，我们采用比例因子(Scaling)方法以克服参数下溢的问题.[3]

在计算前向和后向变量的基础上，我们对 HMM 的参数进行重估算. 我们首先定义 $\xi_t(i,j)$为给定观察值序列 O 和模型 λ，在 t 时刻处于状态 S_i，在 $t+1$ 时刻处于状态 S_j 的概率，即

$$\xi_t(i,j)=P(q_t=S_i,q_{t+1}=S_j|O,\lambda). \tag{6}$$

它可以用下面的公式计算

$$\xi_t(i,j)=\frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}=\frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^{N}\sum_{m=1}^{N}\alpha_t(l)a_{lm}b_m(O_{t+1})\beta_{t+1}(m)} \tag{7}$$

还定义 $\gamma_t(i)$ 为给定观察值序列 O 和模型 λ,在 t 时刻处于状态 S_i 的概率,计算如下

$$\gamma_t(i)=\begin{cases}\sum_{j=1}^{N}\xi_t(i,j)=\dfrac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N}\alpha_t(k)\beta_t(k)} & 1\leqslant t\leqslant T-1\\ \dfrac{\alpha_T(i)}{\sum_{k=1}^{N}\alpha_T(k)} & t=T\end{cases} \tag{8}$$

在处理多个训练观察值序列时(在词性标注中即为用来训练模型的多个句子),例如给定 L 个观察值序列 $O(1),O(2),\ldots,O(L)$,为了方便表述本算法和下一节的修改方法,我们特定义 4 个期望值变量(Expected Variables)以记录有关的中间计算结果:$startstate(i)$ 为在 L 个训练观察值序列中,在时刻 $t=1$,处于状态 S_i 的次数的期望值;$transition(i,j)$ 为在 L 个训练观察值序列中,从状态 S_i 转移到状态 S_j 的次数的期望值;$transfrom(i)$ 为在 L 个训练观察值序列中,从状态 S_i 转移出去的次数的期望值;$observation(j,k)$ 为在 L 个训练观察值序列中,处于状态 S_j 并且输出观察值 w_k 的次数的期望值;$state(i)$ 为在 L 个训练观察值序列中,处于状态 S_i 的次数的期望值. 这些变量可以计算如下

$$startstate(i)=\sum_{l=1}^{L}\gamma_1^{(l)}(i) \tag{9}$$

$$transition(i,j)=\sum_{l=1}^{L}\sum_{t=1}^{T^{(l)}-1}\xi_t^{(l)}(i,j) \tag{10}$$

$$transfrom(i)=\sum_{l=1}^{L}\sum_{t=1}^{T^{(l)}-1}\gamma_t^{(l)}(i) \tag{11}$$

$$observation(j,k)=\sum_{l=1}^{L}\sum_{\substack{t=1\\ \text{and } O_t=w_k}}^{T^{(l)}}\gamma_t^{(l)}(j) \tag{12}$$

$$state(i)=\sum_{l=1}^{L}\sum_{t=1}^{T^{(l)}}\gamma_t^{(l)}(i) \tag{13}$$

那么,HMM λ 的参数可以重新估算为

$$\bar{\pi}_i=\frac{\text{在时刻}(t=1)\text{处于状态}S_i\text{的次数的期望值}}{\sum_{j=1}^{N}\text{在时刻}(t=1)\text{处于状态}S_j\text{的次数的期望值}}=\frac{startstate(i)}{\sum_{j=1}^{N}startstate(j)} \tag{14}$$

$$\bar{a}_{ij}=\frac{\text{从状态}S_i\text{转移到状态}S_j\text{的次数的期望值}}{\text{从状态}S_i\text{转移出去的次数的期望值}}=\frac{transition(i,j)}{transfrom(i)} \tag{15}$$

$$\bar{b}_j(k)\frac{\text{处于状态}S_j\text{并且输出观察值}w_k\text{的次数的期望值}}{\text{处于状态}S_j\text{的次数的期望值}}=\frac{observation(j,k)}{state(j)} \tag{16}$$

使用这些公式,我们可以按下面的方法对模型 $\lambda=(\pi,A,B)$ 进行重新估算:给定 L 个观察值序列,如一个包含 L 个句子的训练语料,我们用公式(14)~(16)来估算参数,得到一个新的模型 $\bar{\lambda}=(\bar{\pi},\bar{A},\bar{B})$. 这就是著名的 Baum-Welch 重估算算法. 可以证明:在最大相似

(Maximum Likelihood)原则的意义上,$\bar{\lambda}$ 比 λ 要好.[3,4]因此,我们可以找到一个更适合训练语料的模型.

2 对经典 HMM 方法的改进

因为不同领域的语料的概率分布不同[5],HMM 的参数也应随着领域的变化而变化,因此,估算模型的参数是使用 HMM 方法来标注语料的关键问题.如果我们仔细研究经典的 HMM 和它的参数估算算法,可以发现其中存在着两个问题,限制了该方法在实际中得到有效应用.

第 1 个问题是,当由于引入新的语料而需要重新调整模型的参数时,Baum-Welch 算法不能方便地修改模型.例如,给定训练语料 $C1$,我们能用 Baum-Welch 重估算算法得到一个训练好的模型 $\lambda=(\pi,A,B)$,因此 λ 反映了语料 $C1$ 的信息.如果在得到 λ 之后,新的训练语料 $C2$ 又被引入了,我们如何建立一个模型既能反映 $C1$ 又能反映 $C2$ 呢?若使用经典的 Baum-Welch 方法,我们则有两种方法可以考虑:① 使用消去内插方法(Deleted Interpolated),通过合并原模型 λ 和从新语料 $C2$ 中训练得到模型来获得一个新的模型.但是,如前言所述,这种方法不仅非常复杂,而且新模型的好坏极大地依赖使用者的经验,在实际应用中缺乏可靠性和通用性,难以令人满意.[2]② 将 $C1$ 和 $C2$ 合并在一起,用它们重新训练一个新的模型以反映 $C1$ 和 $C2$ 中的信息,采用这种方法,原模型 λ 并未在训练过程中得到利用,为得到 λ 所作的工作被全部抛弃,非常浪费.特别是,这种方法要求我们保存所有的训练语料以便后续的训练工作,使得模型与训练语料不能分离.然而在某些情况下,并未保留过去的训练语料而只保存模型的参数,这时这种方法就无法发挥作用了,譬如,若 $C1$ 未被保存,我们应该怎么办呢?也许会想到:以 λ 为初始模型,对语料 $C2$ 使用 Baum-Welch 方法,得到一个新的模型 λ'.然而显然,λ' 只反映了 $C2$ 而没有反映 $C1$.λ 中的大量的信息还是被丢失了,并未达到我们的要求.

经典 HMM 及其 Baum-Welch 估算算法存在的另一个问题是,它只能使用未加工的原始语料来训练模型的参数,而不能充分、有效地利用已标注的语料来训练模型.Merialdo 在文献[1]中显示:使用标注好的语料提供的频率统计信息作为初始模型,能大大提高估算的准确率.由此可以想到,将标注过的语料提供的频率信息结合到模型中,可以有效地提高模型的准确性(这一点也为我们的实验所证实).但是,在经典 HMM 理论框架下,我们只能在建立初始模型时使用已经标注过的语料,一旦模型经过训练而确定之后,标注过的语料就难以发挥作用了.那么,如何用标注过的语料训练已经存在的模型呢?经典的 Baum-Welch 方法不能解决这一问题,而我们通过修改该方法,能将这一问题与上面第 1 个问题一起解决.

为了解决 HMM 中存在的这两个问题,从语音处理领域的研究结果中得到启示[6],我们对原方法作如下修改:

给定两个训练语料 $C1$ 和 $C2$,根据式(14)~(16)可以得到

$$\bar{\pi}_i=\frac{\text{在语料 } C1 \text{ 和 } C2 \text{ 中,在时刻}(t=1)\text{处于状态 } S_i \text{ 的次数的期望值}}{\sum_{j=1}^{N}\text{在语料 } C1 \text{ 和 } C2 \text{ 中,在时刻}(t=1)\text{处于状态 } S_j \text{ 的次数的期望值}}$$

$$=\frac{startstate^{(C1)}(i)+startstate^{(C2)}(i)}{\sum_{j=1}^{N}startstate^{(C1)}(j)+\sum_{j=1}^{N}startstate^{(C2)}(j)} \tag{17}$$

$$\overline{a_{ij}}=\frac{\text{在语料 } C1 \text{ 和 } C2 \text{ 中,从状态 } S_i \text{ 转移到状态 } S_j \text{ 的次数的期望值}}{\text{在语料 } C2 \text{ 和 } C2 \text{ 中,从状态 } S_i \text{ 转移出去的次数的期望值}}$$

$$=\frac{transition^{(C1)}(i,j)+transition^{(C2)}(i,j)}{transfrom^{(C1)}(i)+transfrom^{(C2)}(i)} \tag{18}$$

$$\overline{b_j(k)}=\frac{\text{在语料 } C1 \text{ 和 } C2 \text{ 中,处于状态 } S_j \text{ 并且输出观察值 } w_k \text{ 的次数的期望值}}{\text{在语料 } C_1 \text{ 和 } C_2 \text{ 中,处于状态 } S_j \text{ 的次数的期望值}}$$

$$=\frac{observation^{(C1)}(j,k)+observation^{(C2)}(j,k)}{state^{(C1)}(j)+state^{(C2)}(j)} \tag{19}$$

根据上面的推导,我们可以按下列方法解决 HMM 的两个问题. 首先,给定训练语料 $C1$,我们用 Baum-Welch 方法从该语料中训练得到模型 $\lambda=(\pi,A,B)$,然而在保存模型时,我们并未直接保存 π,A 和 B 的值,而是在 λ 中保存所有的 $C1$ 的期望值变量:$startstate^{(C1)}(i)$,$transition^{(C1)}(i,j)$,$transfrom^{(C1)}(i)$,$observation^{(C1)}(j,k)$和 $state^{(C1)}(i)$(对于所有的 i,j,k). 由这些变量,我们可以非常容易地使用公式(14)～(16)计算出 π,A 和 B 的值. 现在假设有新的语料 $C2$ 引入,我们希望建立一个既能反映 $C1$ 又能反映 $C2$ 的模型. 我们使用 λ 作为初始模型,通过 Baum-Welch 重估算方法(公式(9)～(13)),我们将得到 $C2$ 的期望值变量:$startstate^{(C2)}(i)$,$transition^{(C2)}(i,j)$,$transfrom^{(C2)}(i)$,$observation^{(C2)}(j,k)$和 $state^{(C2)}(i)$(对于所有的 i,j,k). 然后我们将 λ 中保存的 $C1$ 的期望值变量与 $C2$ 的期望值变量相加,得到了反映 $C1$ 和 $C2$ 的期望值变量的值,将这些值保存下来就得到了新的模型 λ^*. 显然,λ^* 的 π,A 和 B 的值也可以方便地由公式(14)～(16)计算得到. 这样,根据公式(17)～(19),模型 λ^* 既反映了 $C1$ 的信息又反映了 $C2$ 的信息. 另外,由于保存了期望值变量,我们不需要再为了后续的训练而保存用过的训练语料了,因此模型和训练模型的语料能够分离开来,具有良好的灵活性.

根据我们的修改,第 2 个问题也能方便地得到解决. 注意到我们定义的期望值变量保存的是用 Baum-Welch 算法计算的期望值,如果我们给定的训练数据是标注过的语料,那么这些期望值变量的值就是已标注语料的相应的频率统计值. 以上面的例子,假设 $C2$ 是被手工标注过的语料,我们能够通过频率统计得到它的期望值变量的值,方法如下:

$startstate^{(C2)}(i)$=在语料 $C2$ 中以状态(标记)S_i 开头的句子的数目　(20)

$transition^{(C2)}(i,j)$=在语料 $C2$ 中从状态(标记)S_i 转移到状态(标记)S_j 的数目　(21)

$transfrom^{(C2)}(i)$=在语料 $C2$ 中从状态(标记)S_i 转移出去的数目　(22)

$observation^{(C2)}(j,k)$=在语料 $C2$ 中处于状态(标记)S_j 且对应观察值单词为 w_k 的数目　(23)

$state^{(C2)}(i)$=在语料 $C2$ 中状态(标记)S_i 的数目　(24)

类似地,我们可以将 λ 中保存的 $C1$ 的期望值变量与用公式(20)～(24)计算的 $C2$ 的期望值变量相加,从而得到新的模型 λ^*. 显然,模型 λ^* 既反映了 $C1$ 也反映了 $C2$. 因此,根据我们的方法,标注过的语料包含的信息被很好地结合进了模型,并保持了模型的概率学意义. 我们可以借助这种方法用标注过的语料来训练 HMM.

下节介绍采用这种方法逐步训练 HMM 的一个实验,在实验中,已标注过的语料和未标注过的原始语料都被用来充当训练数据.

3 实验结果及讨论

本文的主要目的是通过对 HMM 及其估算方法进行修改,使得我们在引入新的训练语料时(不论这种语料是标注过的,还是未标注过的),能够动态地修改模型,使之既能反映新的语料的信息,又能保持过去的训练语料的信息.我们用实验检验了上节中提出的方法.从上面的讨论中可以看出,我们提出的方法与经典的 HMM 方法一样,并不依赖某种特定的语言,它可以适用于多种语言,如中文、英文等.在实验中,我们用英语作为研究的对象.

在实验中,我们使用北京大学俞士汶教授提供的语料作为训练和测试语料.我们从这些语料中随机挑选了 1 714 个英语句子(14 944 个单词),并进行了手工标注.标记集由作者设计,包括 50 个标记,如 *Ns*,*Ving*,*ADJer*,*Ved* 等.这些句子被划分为 3 个子语料库:$C1$(635 句,5 464 词),$C2$(504 句,4 515 词)和 $C3$(575 句,4 965 词).实验中,我们用这 3 个子语料库分别进行训练和测试.另外,我们还建立了一个词典,其中包含了语料中出现的所有单词及其所有可能的标记.

我们以均匀分布的模型作为实验的初始模型 M_0.首先,对子语料库 $C1$ 的原始英文句子使用 Baum-Welch 算法,得到 $C1$ 的期望值变量,并将这些期望值变量加入到 M_0 中以形成新的模型 M_1.然后,将子语料库 $C2$ 的手工标注结果作为训练数据,使用公式(20)~(24)从中计算出 $C2$ 的期望值变量的值,并将这些值与 M_1 中的相应的期望值变量相加,从而得到新的模型 M_2.因此,M_2 包含了 $C1$ 和 $C2$ 中的信息.最后,我们用 $C3$ 的原始句子来训练模型 M_2,又得到一个新的模型 M_3.通过使用 Viterbi 算法,所有这些模型都被用来标注 3 个子语料库:$C1$,$C2$ 和 $C3$.下面的表格列出了实验结果.

表 1 各个模型的测试结果

	语料库 $C1$ 5 464 个单词		语料库 $C2$ 4 515 个单词		语料库 $C3$ 4 965 个单词		总语料($C1+C2+C3$) 14 944 个单词	
模型	正确的单词数	标注正确率	正确的单词数	标注正确率	正确的单词数	标注正确率	正确的单词数	标注正确率
M_0	5 046	92.3%	4 146	91.8%	4 504	90.7%	13 696	91.6%
M_1	5 117	93.6%	4 247	94.1%	4 636	93.4%	14 000	93.7%
M_2	5 136	94.0%	4 383	97.1%	4 714	94.9%	14 233	95.2%
M_3	5 184	94.9%	4 347	96.3%	4 728	95.2%	14 259	95.4%

从实验结果可以看出:

(1) 初始均匀模型 M_0 的测试结果的准确率最低.这是由于均匀模型没有提供任何概率信息以指导标注的选择,因此,对于有多个可能标记的单词来说,标记是随机选择的.但由于语料中的大多数单词只有一种可能的标记,所以标注的准确率仍然超过 90%;

(2) 通过 Baum-Welch 算法,用 $C1$ 的原始语料训练 M_0 后得到模型 M_1,M_1 的标注准确率较 M_0 有较大的提高.另外,模型 M_2 标注的准确率也提高得比较明显,这说明标注过的语料确实能提高模型的准确性;

(3) 模型 M_2 对于 $C2$ 的标注准确率的提高幅度比 $C1$ 和 $C3$ 都要大.我们认为这种封闭测试结果较开放测试结果更好的现象说明了标注过的语料提供的信息更准确,也更加特殊,所以对于训练语料之外的语料的标注准确率提高的幅度有所下降.从另一个角度看,这一现

象也告诉我们,如果标注的语料越多,我们训练得到的模型也会越好;

(4) 值得注意的是,模型 M_3 标注 $C2$ 的准确率比模型 M_2 要低.由刚才的讨论可知,M_2 是用 $C2$ 的标注结果训练的,它包含了 $C2$ 中的非常准确的信息,所以对于 $C2$ 的标注准确率很高.但通过用 $C3$ 的原始句子训练 M_2 之后,M_2 中包含的关于 $C2$ 的信息与 $C3$ 的信息进行了合并,这实际上是一个平滑的过程,模型 M_3 不仅包含了 $C2$ 的信息,而且还包含了 $C1$ 和 $C3$ 的信息,所以尽管 M_3 降低了 $C2$ 的准确率,但从整个语料库来看,总的准确率还是提高了.这说明我们的方法是有效的.

实验结果显示,第 2 节所介绍的方法能够在新的语料(不论语料是否经过标注)引入时方便地修改模型的参数,使之能同时反映新的和旧的训练语料的信息,提高模型的准确性.

4　结束语

将隐马尔可夫模型 HMM 用于语料的词性标注被实验证明是有效的.该方法的关键问题是如何训练模型的参数.使用经典的参数估算方法,模型的参数一旦确定下来,不仅难以随着新的训练语料的增加而动态地作出调整,而且也不能利用已经标注好的语料来训练模型.我们修改了经典方法,使得模型能够随着新语料(不论该语料是否经过标注)的增加而动态地调整参数,从新的语料中获取知识以提高模型的准确性.初步实验结果显示,我们的方法是有效的.为了进一步检验和完善该方法,我们将对更大规模的语料进行标注和研究.

参考文献

1　Meriaido B. Tagging English text with a probabilistic model. Computational Linguistics, 1994,**20(2)**:155～171.

2　Jelinek F, Mercer R L. Interpolated estimation of Markov source parameters from sparse data. In: Gelsema E S, Kanal L N eds. Pattern Recognition in Practice, Amsterdam: North Holl and Publishing Co., 1980. 381～402.

3　Rabiner L A. Tutorial on hidden Markov models and selected applications in speech recognition. Proc. of IEEE, 1989,**77(2)**:257～286.

4　Baum L E, Sell G R. Growth functions for transformation on manifolds. Pac. J. Math., 1967,**27(2)**:211～227.

5　Biber D. Using register-diversified corpora for general language studies. Computational Linguistics, 1993,**19(2)**: 219～240.

6　Xie J H, Gao Y Q, Tu J H. Speaker adaptation methods for speech recognition systems based on linear predictive hidden Markov models. In: Proc. of Inter. Conf. on Computer Processing of Chinese and Oriental Language, Changsha China, 1990.

AN ADAPTIVE METHOD FOR PART-OF-SPEECH TAGGING

WANG Ting　CHEN Huowang　YANG Yi　SHI Xiaodong

(*Department of Computer Science　National University of Defense Technology　Changsha*　410073)

Abstract　In this paper, the authors modified the classical HMM (hidden Markov model) method so as to train the model whenever the new corpus is introduced and no matter the corpus is tagged or not. Therefore, the information can be learned from the corpus to improve the accuracy of the model.

Key words　Corpus, part-of-speech tagging, hidden Markov model.

Class number　TP391