基于级联重排序的汉语音字转换

李鑫鑫 [1,2] 王 轩 [1,2] 姚 霖 [1,3] 关 键 [1,3]

摘 要 N 元语言模型是解决汉字音字转换问题最常用的方法. 但在解析过程中, 每一个新词的确定只依赖于前面的邻近词, 缺乏长距离词之间的句法和语法约束. 我们引入词性标注和依存句法等子模型等来加强这种约束关系, 并采用两个重排序方法来利用这些子模型提供的信息: 1) 线性重排序方法, 采用最小错误学习方法来得到各个子模型的权重, 然后产生候选词序列的概率; 2) 采用平均感知器方法对候选词序列进行重排序, 能够利用词性、依存关系等复杂特征. 实验结果显示, 两种方法都能有效地提高词 N 元语言模型的性能. 而将这两种方法进行级联, 即首先采用线性重排序方法, 然后把产生的概率作为感知器重排序方法的初始概率时性能取得最优.

关键词 汉语音字转换, 重排序, 最小错误学习, 感知器方法

引用格式 李鑫鑫, 王轩, 姚霖, 关键. 基于级联重排序的汉语音字转换. 自动化学报, 2014, **40**(4): 624−634

DOI 10.3724/SP.J.1004.2014.00624

Chinese Pinyin-to-character Conversion Based on Cascaded Reranking

LI Xin-Xin[1,2] WANG Xuan[1,2] YAO Lin[1,3] GUAN Jian[1,3]

Abstract The word n-gram language model is the most common approach for Chinese pinyin-to-character conversion. It is simple, efficient, and widely used in practice. However, in the decoding phase of the word n-gram model, the determination of a word only depends on its previous words, which lacks long distance grammatical or syntactic constraints. In this paper, we propose two reranking approaches to solve this problem. The linear reranking approach uses minimum error learning method to combine different sub-models, which includes word and character n-gram language models, part-of-speech tagging model and dependency model. The averaged perceptron reranking approach reranks the candidates generated by word n-gram model by employing features extracted from word sequence, part-of-speech tags, and dependency tree. Experimental results on "Lancaster Corpus of Mandarin Chinese" and "People's Daily" show that both reranking approaches can efficiently utilize information of syntactic structures, and outperform the word n-gram model. The perceptron reranking approach which takes the probability output of linear reranking approach as initial weight achieves the best performance.

Key words Chinese pinyin-to-character conversion, reranking approach, minimum error learning, averaged perceptron

Citation Li Xin-Xin, Wang Xuan, Yao Lin, Guan Jian. Chinese pinyin-to-character conversion based on cascaded reranking. *Acta Automatica Sinica*, 2014, **40**(4): 624−634

汉语音字转换问题是中文信息处理中的重要问题之一, 是汉语智能输入技术、语音识别等任务的基础. 但是目前这个任务仍然具有一定的挑战性. 对于汉语来说, 通常包括 410 个无调音节 (不同的标准下数目有所不同). 而汉语通用字表包括 7 000 个字, 汉语常用字表包括 3 500 个字. 这表明每个无调音节都可能对应着多个汉字. 例如, 在汉语常用字表中就有 60 个汉字的发音为 "yi". 汉语音字转换问题就是同音字的消歧问题.

目前, 解决汉语音字转换问题最常用的方法是词 N 元语言模型. N 元语言模型理论成熟、易于训练, 可以很好地集成到解码过程中, 应用非常广泛[1]. 但同时它也存在着很多问题. 其中一个问题是 OOV (Out of vocabulary) 问题. 由于汉语词的数目比较多, 并且仍然在持续增长中, 所以语言模型无法包括所有的词. 对于 OOV 词, 语言模型赋予的概率比较低, 采用平滑技术[1−2] 可以改善但并不能真正解决这个问题. 适应学习方法可以根据用户的输入和反馈来不断的增加新词和修正词频[3−6]. 云输入法利用服务器端大规模的存储和计算能力, 弥补了传统

收稿日期 2013-04-22 录用日期 2013-09-22
Manuscript received April 22, 2013; accepted September 22, 2013
国家科技部重大科技专项 (2011ZX03002-004-01), 深圳市基础研究重点项目 (JC201104210032A, JC201005260112A) 资助
Supported by Key Science and Technology Projects of the Ministry of National Science and Technology (2011ZX03002-004-01) and Shenzhen Basic Research Key Project (JC201104210032A, JC201005260112A)
本文责任编委 党建武
Recommended by Associate Editor DANG Jian-Wu
1. 哈尔滨工业大学深圳研究生院计算机应用研究中心 深圳 518055 2. 深圳互联网多媒体应用技术工程实验室 深圳 518055 3. 移动互联网应用安全产业公共服务平台 深圳 518057
1. Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055 2. Shenzhen Applied Technology Engineering Laboratory for Internet Multimedia Application, Shenzhen 518055 3. Public Service Platform of Mobile Internet Application Security Industry, Shenzhen 518057

输入法受限于单机内存、只能使用规模较小的词库和语言模型的不足, 从而提升输入准确率[1].

词 N 元语言模型面临的第二个问题是它只针对前面的邻近词建立模型, 缺少句法和语法上的约束, 特别是长距离词之间的约束. 汉语中的一些语言现象只通过序列结构, 如词 N 元语言模型, 是无法描述清楚的. 例如在 "一只漂亮可爱的小花猫" 中, "一只" 的确定要依赖于与词 "小花猫" 的关系, 以区别于 "一支" 依赖于 "小花". 在词 N 元语言模型中, 这种约束关系很难体现. 目前已经有很多方法提出来用于解决这个问题, 包括更高阶的语言模型[7−9]、机器学习方法[10−14] 以及后处理规则等[15].

针对第二个问题, 我们引入了多个子模型到汉语音字转换问题, 包括词性标注模型、依存句法模型等. 然后在此基础上采用两种方法对汉语音字转换的候选词序列进行重排序. 在线性重排序方法中, 采用了最少错误训练方法来获得每个子模型的权重. 平均感知器重排序方法能够使用每个候选值中的词、词性、依存关系作为特征. 实验结果显示引入的多个子模型对汉语音字转换问题有重要作用, 将两种方法进行级联结合, 即首先采用线性重排序方法, 然后把产生的概率作为感知器方法的初始概率时性能取得最优. LCMC (Lancaster Corpus of Mandarin Chinese) 语料库和人民日报语料上的实验表明本文的级联重排序方法要优于目前已有的工作.

1 相关研究

词 N 元语言模型是汉语音字转换问题最常用的方法. 对于一个拼音序列, 它选择概率最大的词串或字串作为正确的识别结果. 在解码过程中, 当前词/字的确定只依赖于前面邻近的拼音和字/词串 (详情见第 2 节). 这个方法简单有效, 但是很多语言现象用词 N 元语言模型并不能很好地表示, 例如长距离词之间的约束现象和形容词、名词的嵌套结构等.

目前已经有很多工作提出来用于解决这个问题. 一种方法是采用更有效的语言模型存储和索引方法, 能够减少模型的规模, 提高训练和预测的速度, 以便在有效的存储空间上支持更高阶的语言模型[7−8]. Siu 等提出了一种可变 N 元语言模型, 用树结构来替代原始的语言模型[9]. 在这个树模型中, 可以通过调整节点的合并和结合来增加 N 元语言模型的长度. 从而可以使这个模型表示出比传统模型更长的词关系. 基于类的语言模型是另外一种引入长距离约束的模型[2]. 它把词/短语分成不同的类别, 因为类别的数目要远远小于词的数目, 这样就可以表示更长的词约束关系.

词 N 元语言模型也可以通过集成更高层次的模型来加以扩展. Ney 等提出了一个模型把两个词之间的关系引入到词 N 元语言模型, 从而能够使用长距离词之间的依存关系表示的信息[16]. 同样也可以将概率自顶向下的句法解析模型 (上下文无关文法) 融入到词 N 元语言模型, 这样就可以有效的利用句法结构信息[17−18].

基于规则的后处理方法可以引入长距离词的约束或者嵌套词关系到基于转换的方法中. 使用这些句法和语法约束规则对候选词进行处理, 去掉不符合规则的候选词, 可以有效地增强词 N 元语言模型. 这些约束关系通常表示各种不同的规则, 这些规则可以通过手动或自动的方法进行提取. 文献 [15] 提出了一种通过粗糙集理论从训练数据中自动提取汉语音字转换规则的方法.

各种机器学习方法, 例如最大熵模型[10−11]、最大熵马尔科夫模型[12]、支持向量机模型[13]、条件随机场模型[14]、机器翻译的方法[19] 等, 都可以用来解决这个问题. 这些模型使用当前词/字前后的信息和长距离的词/字信息作为特征, 有效地弥补了词 N 元语言模型只使用前面邻近的词/字的不足.

除了传统的基于准确拼音序列的音字转换问题外, 还有不少工作是在做基于连续拼音流和基于有输入错误的拼音串的音字转换研究[20−21]. 文献 [22] 提出了一种基于错误容忍的模型, 可以将音节串的错误条件概率引入到原始的音字转换概率模型中. 对于一个有错误的拼音串, 它能够选择可能所有的正确拼音作为候选集, 然后使用概率容忍模型对其重排序选出最优的结果. 本文主要基于传统的音字转换进行研究.

2 N 元语言模型

本节主要描述汉语音字转换问题和采用 N 元语言模型进行音字转换的方法. 对于一个音节序列 $S = s_1, s_2, \cdots, s_n$, 汉语音字转换的目的是找到对应的字序列 $C = c_1, c_2, \cdots, c_n$, 或词序列 $W = w_1, w_2, \cdots, w_m$ $(m \leq n)$. 一个音字转换的例子如图 1 所示. 其中, 词 w_k 是由字序列 $c_i, \cdots, c_j$ 组成, 对应着拼音序列 $(s_i, \cdots, s_j)$. 其中 i, j 随着 k 的不同而变化. 对于音节序列 S, 选择的最佳词序列 W 满足:

$$W^* = \arg\max_W P(W|S) \tag{1}$$

采用词 N 元语言模型时, 根据贝叶斯公式:

$$W^* = \arg\max_W \frac{P(S|W)P(W)}{P(S)} \tag{2}$$

[1] http://pinyin.sogou.com

其中,

$$P(S|W)=\prod_{k=1}^{m}p((s_i,\cdots,s_j)|w_k) \tag{3}$$

$$P(W)=\prod_{k=1}^{m}p(w_k|w_1,\cdots,w_{k-1}) \tag{4}$$

jian chi	gai ge	、	kai fang	,
坚 持	改 革	、	开 放	,

图 1 一个音字转换的例子

Fig. 1 An example of Chinese pinyin-to-character conversion

对于三阶语言模型, $P(W)=p(w_1)p(w_2|w_1)\times\prod_{k=3}^{m}p(w_k|w_{k-2},w_{k-1})$. 如果采用字 N 元语言模型, 最佳的字序列 C 可以通过下式来确定:

$$C*=\arg\max_{C}P(C|S)=\arg\max_{C}\frac{P(S|C)P(C)}{P(S)} \tag{5}$$

其中, $P(S|C)$ 和 $P(C)$ 同 $P(S|W)$ 和 $P(W)$ 的计算方法相似.

当使用基于词 N 元语言模型时, 可以通过一个 Beam 搜索解码方法来得到最优的词序列和最优的 k 个词序列. 在解码过程中, $P(S|W)$ 和 $P(S|C)$ 通常是省略的. 解码算法从第一个音节开始. 对于音节序列中位置 j, 解码算法会根据词典生成以音节 s_j 结尾的音节序列 $s_i,\cdots,s_j$ $(i=\max(0,j-20))$ 对应的所有可能词, 然后语言模型给出当前词基于前面邻近词序列的概率, 从而得到当前位置候选词序列的概率. 对于每个位置 j, 算法可以保存最优的 k 个词序列. 算法依次向后进行解析, 最优的词序列为句子结尾位置上概率最大的词序列.

3 子模型

单个词 N 元语言模型并不能很好地解决汉语音字转换问题, 因此引入词性、句法等信息是必要的. 为了更好地使用这些信息, 我们对每个候选词序列都进行词性标注和句法分析. 本文我们引入了多个子模型, 主要包括基于字的模型、词性标注模型、拼音–词共现模型、词–词性共现模型、词性 N 元语言模型和依存句法模型等. 这些模型都能够对候选词序列生成相应的概率.

3.1 基于字的模型

我们采用的基于字的模型是一个判别式机器学习模型, 该模型能够使用音节序列的信息作为特征. 模型采用平均感知器方法进行训练[23], 其中特征模板见表 1 所示.

表 1 基于字的模型采用的特征模板

Table 1 Feature templates for the character-based model

1	s_n $(n=-2,\cdots,2)$
2	s_ns_{n+1} $(n=-1,\cdots,0)$
3	$s_{-1}s_1$

表 1 中, s_0 表示当前音节, s_{-n},s_n 表示当前音节前面的第 n 个音节和后面的第 n 个音节. 对于音节序列 S, 它对应的字序列 C 选择为

$$C*=\arg\max_{C\in GEN(S)}P_{\text{char}}(C|S)=\arg\max_{C\in GEN(S)}\Phi(S,C)\times\bar{\alpha} \tag{6}$$

其中, $P_{\text{char}}(C|S)$ 为模型的概率输出. 特征函数 $\Phi(S,C)$ 可以在解码之前就全部生成, 从而减少解码的时间. 模型的解码算法和平均感知器的参数训练方法可以参考文献 [23−24].

3.2 拼音 – 词共现模型

对于一个音节序列 S 和其对应一个词序列 W, 我们定义拼音–词共现概率分别为

$$P_{\text{occur}}(W|S)=\prod_{k=1}^{m}p(w_k|(s_i,\cdots,s_j)) \tag{7}$$

$$P_{\text{occur}}(S|W)=\prod_{k=1}^{m}p((s_i,\cdots,s_j)|w_k) \tag{8}$$

其中, $p(w_k|(s_i,\cdots,s_j))$ 和 $p((s_i,\cdots,s_j)|w_k)$ 为单个词 w_k 和拼音 $(s_i,\cdots,s_j)$ 的共现概率, 共现在本文中表示词和拼音是严格对齐的. 一个音节序列 $s_i,\cdots,s_j$ 可能存在着多个对应的词, 一个词也可能对应着多个音节序列, 所以 $p(w_k|(s_i,\cdots,s_j))$ 和 $p((s_i,\cdots,s_j)|w_k)$ 都不恒为 1. 我们根据频率估计得到共现概率, 并采用加一平滑方法来消除稀疏性.

$$p(w_k|(s_i,\cdots,s_j))=\frac{N(w_k,(s_i,\cdots,s_j))+1}{N(s_i,\cdots,s_j)+N_{S_{j-i+1}}} \tag{9}$$

$$p((s_i,\cdots,s_j)|w_k)=\frac{N((s_i,\cdots,s_j),w_k)+1}{N(w_k)+N_{W_{j-i+1}}} \tag{10}$$

其中, $N_{S_{j-i+1}}$ 表示词典中长度为 $j-i+1$ 的音节的数目, $N_{W_{j-i+1}}$ 表示词典中长度为 $j-i+1$ 的词的数目. 音节 $N(s_i,\cdots,s_j)$ 的数目, 词 $N(s_i)$ 的数目和拼音–词对 $N(w_i,(s_i,\cdots,s_j))$ 的数目可以从已

标注好拼音的训练文本中统计. 训练文本与 N 元语言模型采用的文本相同, 可详见第 6 节说明.

3.3 词性标注模型

给定一个词序列 W, 我们可以确定它对应的词性序列 T. 文献 [25] 的研究表明词性信息能够有效地提高中文分词的准确性. 在这里我们引入词性标注模型来帮助选择最优的词序列.

我们采用平均感知器方法来训练模型. 对于词序列 W, 其对应的词性标注序列 T 选择为

$$T^* = \arg\max_{T\in GEN(W)} P(T|W) = \arg\max_{T\in GEN(W)} \Phi(T,W)\times\bar{\alpha} \tag{11}$$

针对候选值 (W,T), 词性标注模型的概率为 $P(T|W)$, 特征大部分来自于文献 [23−26], 如表 2 所示. 我们的特征只采用了当前拼音/词及其前面的信息, 可以随着词序列的解析过程生成, 可以不必预先生成整个词序列.

表 2　词性标注模型采用的特征

Table 2　Feature templates for part-of-speech tagging

1	$w_{-2}t_0$	$end(w_{-1})w_0start(w_1)t_0$
2	$w_{-1}t_0$	when $len(w_0)=1$
3	w_0t_0	$start(w_0)t_0$
4	w_1t_0	$end(w_0)t_0$
5	w_2t_0	$c_nt_0, (n=1, len(w_0)-2)$
6	$t_{-1}t_0$	$start(w_0)c_nt_0$ $(n=above)$
7	$t_{-2}t_{-1}t_0$	$end(w_0)c_nt_0$ $(n=above)$
8	$t_{-1}w_0$	$c_nc_{n+1}t_0$ $(c_n=c_{n+1})$
9	$w_0t_0end(w_{-1})$	$class(start(w_0))t_0$
10	$w_0t_0start(w_1)$	$class(end(w_0))t_0$

我们采用汉语树库 (Chinese treebank 5, CTB5) 来训练词性标注模型, 其中训练集、开发集和测试集的设置与文献 [26] 相同. 在开发集上我们的词性标注模型的 F−1 值为 95.26 %.

3.4 词性 N 元语言模型

与词 N 元语言模型相似, 我们建立一个词性 N 元语言模型. 定义词性序列 T 的概率为

$$P(T)=\prod_{i=1}^{m} p(t_i|t_1,\cdots,t_{i-1}) \tag{12}$$

词性 N 元语言模型采用的训练文本与 N 元语言模型相同. 我们可以从标注好词性的训练文本中得到 $p(t_i|t_1,\cdots,t_{i-1})$, 其计算方法与 $p(w_k|w_1,\cdots,w_{k-1})$ 相似.

3.5 词性 – 词共现模型

我们采用的词性 – 词共现模型与拼音 – 词共现模型的定义类似. 对于一个词序列 W 和它对应的词性序列 T, 它们的共现频率定义为

$$P_{\text{occur}}(W|T)=\prod_{i=1}^{m} p(w_i|t_i) \tag{13}$$

$$P_{\text{occur}}(T|W)=\prod_{i=1}^{m} p(t_i|w_i) \tag{14}$$

其中, $p(w_i|t_i)$ 和 $p(t_i|w_i)$ 都是通过频率统计来得到, 并采用加一平滑消除稀疏性, 计算方法与拼音 – 词共现频率相似.

3.6 依存句法模型

如前面所示, 仅使用相邻的音节和词来判断当前音节对应的词是不够的, 可能需要句法和语法的信息. 依存句法模型能够为一句话中长距离词之间建立依存关系. 对于一个已经分词和词性标注好的句子 (W,T), 我们可以使用基于转移的句法分析方法对其进行解析[27]. 依存句法模型的训练语料采用汉语树库 (CTB5), 训练集、开发集和测试集的设置与文献 [27] 相同, 准确率达到了 86 %.

图 2 给出了一个句法树结构的例子. 从图中可以看出, 词 “一只” 与词 “小花猫” 在序列方向上相隔三个词, 但是存在着直接的依存关系. 这也表明依存关系能为音字转换问题提供有用信息.

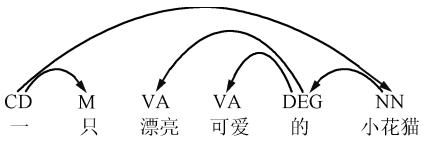

图 2　一个依存句法的例子

Fig. 2　An example of a dependency tree

依存树 D 的概率定义为

$$P(D|W,T)=\prod_i w_i\times f_i(D|W,T) \tag{15}$$

其中, $f_i(D|W,T)$ 表示建立依存树过程中的每个转移状态对应的特征.

4 线性重排序方法

研究表明融合多种信息的模型能够比单个模型取得更好的性能[28]. 本文首先采用一种线性重排序方法来结合第 3 节提出的各种子模型. 针对每个拼音序列 S, 可以根据 Beam 搜索算法产生 k 个最优的词序列 W, 然后对每个词序列进行词性标注和句法分析. 为了有效地利用各种子模型, 我们采用一种

线性方法来使用子模型对每个候选词序列产生的概率. 我们的重排序方法就是从这 k 个候选项中选择出最优的一个.

子模型的权重可以通过最小错误训练方法 (Minimum error training method, MERT) 得到[29]. 最早提出最小错误训练方法是用来解决机器翻译中的特征权重问题, 已成功应用于自然语言处理的不同领域, 如中文分词和词性标注等[30]. 对于汉语音字转换问题来说, 给定一个音节序列 S, 它对应的候选词序列 W 的概率不再只是词 N 元语言模型产生的概率 $P(W)$, 而是多个子模型的概率组合, 计算为

$$\begin{aligned} P_{\text{mert}}(W|S) = P_{\text{mert}}(W,C,T,D|S) = \\ \sum_{i=0}^{k} w_i * P_{\text{sub}}(W,C,T,D|S) = \\ w_0 \times P(W) + w_1 \times P(C) + \\ w_2 \times P_{\text{char}}(C|S) + \\ w_3 \times P_{\text{occur}}(W|S) + w_4 \times P_{\text{occur}}(S|W) + \\ w_5 \times P(T|W) + w_6 \times P(T) + \\ w_7 \times P_{\text{occur}}(W|T) + w_8 \times P_{\text{occur}}(W|T) + \\ w_9 \times P(D|W,T) \end{aligned} \tag{16}$$

其中, $\sum_{i=0}^{9} w_i = 1$. $P_{\text{sub}}(W,C,T,D|S)$ 表示各个子模型的概率输出, 包括词 N 元语言模型 $P(W)$, 字 N 元语言模型 $P(C)$, 字模型 $P_{\text{char}}(C|S)$, 拼音–词共现模型 $P_{\text{occur}}(W|S)$ 和 $P_{\text{occur}}(S|W)$, 词性标注模型 $P(T|W)$, 词性 N 元语言模型 $P(T)$, 词性–词共现模型 $P_{\text{occur}}(W|T)$ 和 $P_{\text{occur}}(W|T)$, 和依存句法模型 $P(D|W,T)$. 由于每个词序列 W 只产生唯一对应的词性序列 T 和依存树 D, 所以公式的第 1 行是成立的.

子模型的权重 w_j $(0 \le j \le k)$ 可以通过保持其他权重不变, 每次只计算一个权重 w_j 得到. 每个候选值的概率为

$$\begin{aligned} P_{\text{mert}}(W|S) = w_j \times P_j(W|S) + \\ \sum_{i \ne j} w_i \times P_i(W|S) \end{aligned} \tag{17}$$

在计算权重时, 设置最左边的概率 $w_j \times P_j(W|S)$ 为一个变量, 最右边的概率 $\sum_{i \ne j} w_i \times P_i(W|S)$ 为一个常量. 这时, 直接通过格搜索算法来确定每个子模型的权重是不适合的, 因为重新计算所有候选值的概率找到最优值是非常费时的. 最小错误训练方法对每个 jth 方向使用一个分片式线性搜索算法. 每个子模型的权重 w_j 的最优值肯定在上面公式画出的所有直线的相交点上. 因为当在相交值上变化时, 只有少数候选值需要计算, 所以这种方法很容易找到最优值. 最小错误训练方法的细节可参考文献 [29−31].

5 感知器重排序方法

在线性重排序方法的基础上, 我们提出的一种判别式的重排序方法: 感知器重排序. 对于每个候选词序列 W, 我们生成其对应的词性序列 T 和句法树 D. 感知器模型能够从词序列, 词性序列和依存树结构中提取的全局特征作为长距离约束的信息.

对于一个音节序列 S, 我们可以通过感知器模型来重排序所有可能的候选集. 拼音序列 S 对应的最优词串 W 选择为

$$\begin{aligned} W^* = \arg \max_{W \in GEN(S)} (P_{\text{init}}(W|S) + \\ P_{\text{rerank}}(W|S)) = \\ \arg \max_{W \in GEN(S)} (P_{\text{init}}(W|S) + \\ \Phi(W,S) \times \bar{\alpha}) \end{aligned} \tag{18}$$

式中, $GEN(S)$ 表示音节序列 S 产生的所有候选词序列. 第一个概率 $P_{\text{init}}(W|S)$ 表示在感知器重排序之前拼音序列 S 产生词序列 W 的概率, 它可以设置为词 N 元语言模型产生的概率 $P(W)$, 也可以是线性重排序模型产生的概率 $P_{\text{mert}}(W|S)$. 线性重排序模型 $P_{\text{mert}}(W|S)$ 产生的概率可见第 4 节.

第二个概率 $P_{\text{rerank}}(W|S)$ 表示感知器重排序模型产生的概率. 图 2 描述了词性标注模型和句法分析器产生的句法树实例. 我们的模型使用两种不同类型的特征. 第一类特征是平面序列特征, 包括词和词性的特征组合. 第二类特征是从依存树中抽取的句法特征. 这种句法特征包括父子依存特征、父子兄弟依存特征、祖父父亲孙子依存特征. 在这里只考虑词之间是否存在依存关系, 不考虑有标注的情况. 重排序模型采用的最有效特征可通过实验进行选择, 详见第 6 节.

我们采用平均感知器模型作为重排序方法. 对于从拼音序列 S 产生 k 个最优的依存树 D_i $(i = 1, \cdots, k)$, 模型的参数 $\bar{\alpha}$ 可更新为

$$\bar{\alpha} = \bar{\alpha} + \Phi(\bar{\alpha}, D_g) - \Phi(\bar{\alpha}, D_p) \tag{19}$$

其中, D_g 表示音节序列 S 对应的正确词序列的句法树 D_i. 但对于每个音节序列, 正确词序列可能不在候选词序列集中, 所以我们设置 D_g 为候选集中具有最小字错误率的词序列对应的依存树. D_p 表示候选集中感知器重排序模型产生的概率最大的依存树.

6 实验结果及分析

6.1 数据设置

本文在两个数据集上进行实验. 第一个是人民日报语料, 这是音字转换任务最常用的实验语料, 但是它的文本类型比较单一, 只包含新闻类文本. 我们采用的第二个数据集为 “The Lancaster Corpus of Mandarin Chinese (LCMC)” 语料库[2], 它包括报道、宗教、科学、小说等 15 种不同类型的文本. 因此本文的实验主要在这个语料库上进行, 只采用人民日报语料来与其他工作做比较.

目前, 我们有人工标注拼音的 50 000 句人民日报语料, 选择其中 40 000 句做为训练集, 5 000 句分别作为开发集和测试集[3]. LCMC 语料库共包括 45 735 个人工分词和词性标注后的句子. 其原始的拼音序列是通过 pinyin4j 工具[4]来进行标注的, 而 pinyin4j 工具针对每个字只选择最常用的注音, 多音字也只有一个对应的音节. 我们采用最少分词法对语料库重新进行拼音标注, 标注采用 Sogou 注音词典. 在人工标注的人民日报语料上测试我们的拼音标注方法, 其准确率达到 99.7 %.

为了评价词 N 元语言模型, 我们从每种文本类型中分别选择 200 句话作为开发集和测试集, 剩余的 39 735 句话就作为训练集. 为了更好地模拟汉字输入, 我们对训练集、开发集和测试集进行预处理. 所有句子根据中文句子结束标点进行分割[5], 然后删除包含英文单词的句子. 处理后的训练集、开发集和测试集统计信息如表 3 所示[6].

表 3 训练集、开发集和测试集的统计信息
Table 3 Statistics of training, development, and test data

数据集	训练集	开发集	测试集
句子数	112 859	8 235	8 623
总词数	875 397	62 961	63 608
汉语词数	723 424	52 039	52 263
汉语字数	1 144 559	82 527	82 788
OOV 词数	13 692	1 043	1 245
OOV 词比例 (%)	1.56	1.66	1.96

6.2 N 元语言模型的实验结果

我们的字 N 元语言模型和词 N 元语言模型采用 1998 年、2006 年、2007 年和 2009 年 ∼ 2012 年的人民日报语料进行训练. 由于汉语中词的性质, 获得全部的词是不可能的. 即使我们能够列举出所有的词, 所需的训练语料和生成的模型都是巨大的. 我们的词 N 元语言模型采用的词典共包括 130 750 个词, 来源包括: 7 000 个汉语通用汉字, 56 064 个常用词, 新华字典中的所有词, 94 412 个从 Google Chinese 5-gram corpus[7] 中抽取的高频词. 而字 N 元语言模型采用的词典只包括 7 000 个汉语通用汉字. 基于这个词典, 我们采用最少分词法来分割训练文本, 然后采用 SRILM 工具来训练语言模型[32], Knerser-Ney 方法进行平滑. 本文采用词三元语言模型和字四元语言模型来进行实验.

我们首先在开发集上验证词 N 元语言模型, 并与字 N 元语言模型, 字词 N 元语言模型混合结果的 oracle, 基于词 N 元语言模型产生的 k 个最优结果的 oracle 进行比较, 如表 4 所示.

表 4 不同的语言模型在开发集上的性能 (%)
Table 4 Performance of different LMs on development data (%)

模型	CER	IVWER	OOVWER
字 N 元语言模型	12.92	11.25	15.04
词 N 元语言模型	11.27	9.03	14.23
字词混合模型的 oracle	7.38	6.89	11.58
词模型 k 个最优结果的 oracle	2.01	4.01	5.94

这些模型的实验结果都采用字错误率 (Character error rate, CER) 进行评价:

$$\mathrm{CER} = \frac{\text{预测错误的汉字数目}}{\text{所有汉字的数目}} \times 100\,\% = \frac{\text{所有汉字的数目} - \text{预测正确的汉字数目}}{\text{所有汉字的数目}} \times 100\,\% \tag{20}$$

接着我们测试词 N 元语言模型在开发集上的 k 个最优词序列的 oracle 值, 结果见图 3 所示. 其中 k 表示最优候选集的数目. 当 $n = 1$ 时, CER 表示单个词 N 元语言模型的性能. 从图中还可以看出, 随着候选集数目 k 的增加, oracle 的 CER 值不断减少. 基于此, 我们选取 100 作为候选集的数目.

除 CER 外, 两个语言模型也都采用 IVWER (In vocabulary word error rate) 和 OOVWER (Out of vocabulary word error rate) 进行比较. 表 4 中评估的 IV 词和 OOV 词的长度都大于 1. 表 4 的结果显示词 N 元语言模型在各个评价标准上都优于字 N 元语言模型, 这表明词 N 元语言模型能

[2] http://www.lancs.ac.uk/fass/projects/corpus/LCMC/
[3] http://www.uniml.com/nlp/py2char/pddata.tar.gz
[4] http://pinyin4j.sourceforge.net
[5] 句子结束标点包括: “,”、“.”、“? ”、“!”、“:” 和 “;”.
[6] http://www.uniml.com/nlp/py2char/lcmcdata.tar.gz
[7] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2010T06

比字 N 元语言模型提供更多的约束. 并且采用词 N 元语言模型时, IV 词的性能要优于 OOV 词.

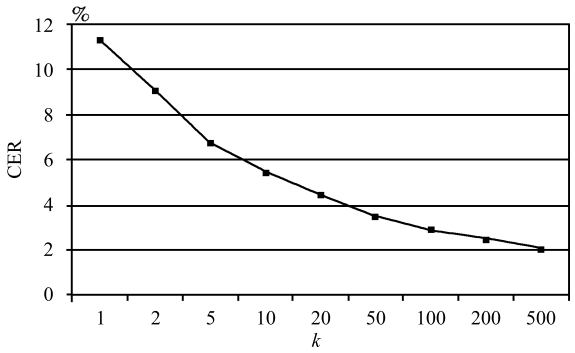

图 3 词 N 元语言模型产生的 k 个最优词序列在开发集上的 oracle

Fig. 3 Oracle of different k-best candidates on development data

我们同时比较了两个模型的混合结果的 oracle, 和基于词的语言模型的 k 个最优结果的 oracle 值. 表 4 的实验结果显示基于词的语言模型和基于字的语言模型能够引起不同的错误, 并且能够互相弥补. 词 N 元语言模型的 k 个最优候选集的 oracle 值表明词 N 元语言模型还是有很大的提高空间.

6.3 线性重排序方法的实验结果

与计算 k 个最优词序列的 oracle 一样, 我们对每个拼音序列 S 生成 100 个候选词序列和对应的依存树 (W, D). 每个子模型都可以对候选项产生一个概率值 $P_{\text{sub}}(W, C, T, D|S)$, 然后通过线性重排序方法选择出最优的一个. 系统的性能采用字错误率 (CER) 作为评价标准.

线性重排序方法采用 LCMC 训练集进行训练, 首先在训练集上得到每个子模型的概率, 然后在开发集上进行测试. 由于并不确定是否每个子模型都有利于汉语音字转换问题, 我们采用一种后向贪婪搜索算法来找到最优的子模型集. 我们的子模型集从包含所有的子模型开始, 在开发集上计算其性能. 然后依次去掉一个子模型并重新计算新子模型集的性能, 从中去掉使性能提高最小的或降低最大的子模型. 重复这个过程直到性能不再增加为止.

我们使用了后向贪婪子模型搜索策略的实验结果表明词 – 拼音共现概率 $P_{\text{occur}}(S|W)$ 对于提高系统的性能没有帮助, 因此我们不使用这个子模型. 表 5 比较了包含所有子模型的线性重排序模型与去掉其中一个子模型的线性重排序模型的结果.

表 5 的实验结果表明所有的子模型都对汉语音字转换问题都有帮助作用. 对于单个子模型来说, 词 N 元语言模型对于提高系统的性能作用最大, 字 N 元语言模型次之. 从表 5 中可以看出, 词性信息和依存句法信息也都有助于提高重排序模型的性能. 去掉词性标注模型、词 – 词性共现模型、词性 N 元语言模型和依存句法模型后系统的性能都有所降低. 不正确的候选词序列中存在着识别错误的词, 对其进行词性标注和依存句法分析后, 产生的词性序列和依存树中也存在着错误, 词性标注模型和依存分析器赋予该非法句子的概率也比较低. 从而我们的重排序模型赋予该非法句子的概率比较低, 选择该非法句子作为识别结果的可能性也比较小.

表 5 线性重排序方法在开发集上的实验结果

Table 5 Experimental results of linear reranking model on development data

编号	子模型集	CER (%)
A	全部	**9.76**
A/0	全部/词 N 元语言模型	10.47
A/1	全部/字 N 元语言模型	10.10
A/2	全部/字模型	9.83
A/3	全部/拼音 – 词共现模型	10.06
A/4	全部/词性标注模型	9.84
A/5	全部/词 – 词性共现模型	9.83
A/6	全部/词性 – 词共现模型	9.76
A/7	全部/词性 N 元语言模型	9.78
A/8	全部/依存句法模型	9.82

为了更好地评价每个子模型的重要性, 我们接着计算单个子模型在整个任务上的性能, 结果如图 4 所示. 其中数字 0 ∼ 8 分别代表各个子模型, 与表 5 中的编号一致. 图 4 中上面的曲线表示单独使用一个子模型时的实验结果. 结果与表 5 的结果是一致的, 其中词 N 元语言模型的字错误率最低, 字 N 元语言模型次之. 单独采用依存句法模型时准确率最低, 主要原因是其缺乏邻近词的相关信息.

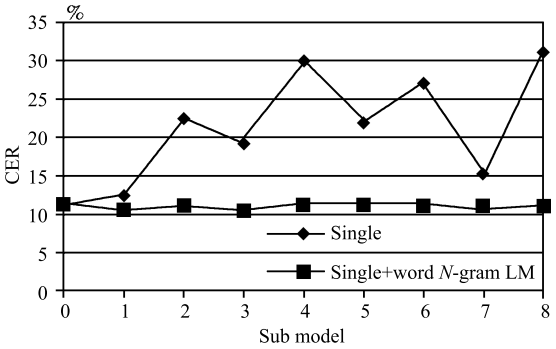

图 4 子模型在开发集的实验结果

Fig. 4 Performance of sub models on development data

下面的曲线表示了词 N 元语言模型与其他任何一个子模型进行线性组合后的实验结果. 结果表

明任意两个子模型进行线性组合的结果都要优于词 N 元语言模型. 在给定词 N 元语言模型的情况下, 添加拼音–词共现模型使性能提高最多, 这也是在词 N 元语言模型原始解码算法中省略的部分.

6.4　感知器重排序方法的实验结果

给定每个拼音序列 S 对应的候选依存树 D, 我们可以使用感知器重排序方法. 重排序方法可以使用两种类型的特征集, 详情见表 6 所示. 第一个特征集是平面序列特征 $(W, T1, T2)$, 包含了词和词性的相关信息. 其中, w_0 和 t_0 分别表示当前标示对应的词和词性, $w_{\pm i}$ 中 $-i$, $+i$ 表示当前词的前面第 i 个词和后面第 i 个词.

第二类特征集表示从依存树中抽取的依存特征. 在表 6 中, 特征集 $D1$ 包括父子结点之间的依存关系, 特征集 $D2$ 包括祖父–父亲–孙子结点之间的依存关系, 特征集 $D3$ 包括父亲–儿子–兄弟结点之间的依存关系. 在这些特征中, G、F、S、L 和 R 分别表示依存关系中的祖父、父子、儿子、左儿子和右儿子结点. 依存关系中父结点到子节点的方向表示为 D_{FS}, 可为 1 或 0, 分别表示左方向和右方向. 另外, A_{FS} 表示结点和子节点是否是相邻的. 在依存关系中, F_tS_w 表示在一个依存关系中父结点的词性和子节点的词的组合.

表 6　重排序模型使用的特征

Table 6　Features for the reranking model

W	$w_{-2}w_{-1}w_0, w_{-1}w_0, w_0$
$T1$	$t_{-2}t_{-1}t_0, t_{-1}t_0, t_0, t_0w_0$
$T2$	$t_{-1}t_0w_0, w_{-1}t_{-1}t_0w_0$
	$t_{-2}w_{-i}t_{-1}t_0w_0, w_{-2}t_{-2}w_{-1}t_{-1}t_0w_0$
$D1$	$F_tD_{FS}A_{FS}S_t, F_wD_{FS}A_{FS}S_wS_t$
	$F_wF_tD_{FS}A_{FS}S_t, F_wF_tD_{FS}A_{FS}S_wS_t$
$D2$	$G_tD_{GF}A_{GF}F_tD_{FS}A_{FS}S_t$
	$G_tD_{GF}A_{GF}F_tD_{FS}A_{FS}S_wS_t$
	$G_tD_{GF}A_{GF}F_wF_tD_{FS}A_{FS}S_t$
	$G_wG_tD_{GF}A_{GF}F_tD_{FS}A_{FS}S_t$
	$G_tD_{GF}A_{GF}F_wF_tD_{FS}A_{FS}S_wS_t$
	$G_wG_tD_{GF}A_{GF}F_wF_tD_{FS}A_{FS}S_t$
	$G_wG_tD_{GF}A_{GF}F_wF_tD_{FS}A_{FS}S_wS_t$
$D3$	$L_tD_{FL}A_{FL}R_tD_{FR}A_{FR}F_t$
	$L_wL_tD_{FL}A_{FL}R_wR_tD_{FR}A_{FR}F_t$
	$L_tD_{FL}A_{FL}R_tD_{FR}A_{FR}F_wF_t$
	$L_wL_tD_{FL}A_{FL}R_wR_tD_{FR}A_{FR}F_wF_t$

我们使用前向贪婪算法来选择重排序模型的特征. 首先考虑平面序列特征, 采用不同的初始概率 $P_{\text{init}}(W|S)$ 和特征集的实验结果见图 5 所示. 如第 5 节所述, 我们采用两种不同的初始概率 $P_{\text{init}}(W)$, 分别为词 N 元语言模型产生的概率 $P(W)$, 和线性重排序模型产生的概率 $P_{\text{mert}}(W|S)$.

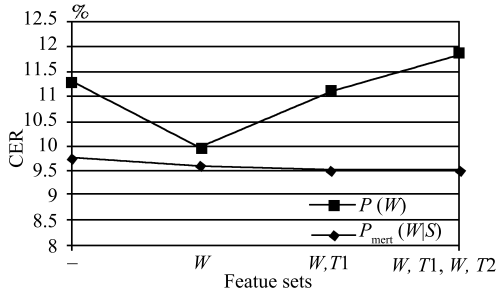

图 5　不同初始权重和平面特征的结果

Fig. 5　Results of reranking models with different initial weights and flat feature sets on development data

图 5 中特征集 "–" (坐标原点) 表示不进行感知器重排序的初始模型. 从实验结果可以看出, 对于两个初始权重, 采用平面序列特征的重排序模型的性能都优于初始模型. 但是对于采用两种不同初始权重的模型, 随着特征的增加时, 其性能的表现不同. 当使用词 N 元语言模型产生的概率 $P(W)$ 作为初始概率时, 使用的特征集 W 的模型的性能要优于使用复杂特征集 $W, D1, D3$. 而当使用线性重排序方法产生的概率 $P_{\text{mert}}(W|S)$ 作为初始概率时, 模型的性能随着特征的增加而提高. 其中采用特征集 W, $T1$ 的模型与特征集 W, $T1$, $T2$ 的性能接近, 但是模型的大小要小很多. 但是采用 $P_{\text{mert}}(W|S)$ 作为初始概率的性能的增长要小于以 $P(W)$ 作为初始概率的方法, 这是因为线性重排序方法已经使用了各个子模型的信息.

采用依存特征的重排序方法的训练和测试方式与采用平面序列特征的方法相似, 首先使用父子依存关系, 然后再分别添加父子兄弟依存关系和祖父父亲儿子依存关系, 实验结果如表 7 所示. 实验表明采用所有依存特征集的感知器重排序方法取得最优的性能, 比词 N 元语言模型高 1.88 个百分点.

6.5　测试集上的实验结果

我们接着在测试集上评估重排序方法的性能, 并与其他方法进行比较, 实验结果如表 8 所示. 从表中可以看出, 采用线性重排序方法时字错误率达到了 10.96 %, 比词 N 元语言模型减少了 1.07 个百分点, 错误率减少了 8.89 %, 显示出与开发集上相似的结果. 把重排序方法进行级联后, 模型的性能不仅优于词 N 元语言模型和单个重排序方法, 还优于现有的工作[33−34].

表 7 不同初始权重和依存特征的实验结果

Table 7 Results of reranking models with different initial weights and dependency feature sets on development data

初识概率	特征集	CER (%)
	W	**9.95**
$P(W)$	$W, D1$	11.26
	$W, D1, D2$	11.37
	$W, D1, D3$	11.46
	$W, T1$	9.50
$P_{\text{mert}}(W\|S)$	$W, T1, D1$	9.40
	$W, T1, D1, D2$	9.40
	$W, T1, D1, D3$	**9.39**

表 8 测试集上的实验结果比较

Table 8 Comparison of different approaches on test data

模型	CER (%)
词 N 元语言模型	12.03
线性重排序	10.96
级联重排序	**10.60**
基于词的隐马尔科夫模型[33]	18.49
基于分割的隐马尔科夫模型[33−34]	13.9

6.6 错误分析

接着我们在开发集上分析重排序方法引起的错误. 表 9 显示了不同长度的词的准确性. IV 表示词典内的词, OOV 表示未登录词. 对于 IV 词来说, 当词长为 1 时, 其准确率不断提高; 而当词长大于 1 时, 从词 N 元语言模型到线性重排序方法有所提高, 而从线性重排序方法到感知器级联重排序方法却有所降低. 而对于 OOV 词来说, 不同长度的词准确性都有所提高. 从词 N 元语言模型到线性重排序方法, 再到感知器重排序方法, 总的词的准确性得到了提高.

表 9 不同方法在开发集上的错误分析

Table 9 Error analysis of different approaches on development dataset

词长	词 N 元语言模型		线性重排序		感知器级联重排序	
	IVR	OOVR	IVR	OOVR	IVR	OOVR
1	81.18	0	82.62	0	**84.26**	0
2	90.34	44.93	**92.47**	46.96	92.06	**47.11**
3	96.34	56.98	**97.84**	58.91	97.05	**60.07**
≥ 4	99.59	78.72	**99.32**	78.01	99.18	**78.72**
All	86.13	52.45	87.86	54.08	**88.45**	**54.56**

图 6 比较了级联重排序模型和词 N 元语言模型在包含不同词数的句子的字错误率. 实验结果显示, 除了单个词的句子外, 对于包含其余词数的句子, 级联重排序模型的字错误率都低于词 N 元语言模型. 其原因可能是我们的级联重排序模型除了依存句法模型, 还包括拼音 – 词共现模型, 词性标准模型等, 从而使得对包含不同词数的句子的识别率都得到提高.

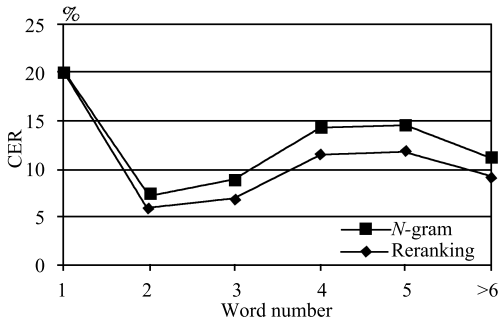

图 6 包含不同词数的句子的字错误率

Fig. 6 Results of different models on sentences with different word numbers

6.7 人民日报数据集上的实验结果

为了更好地评价重排序模型, 我们在人民日报语料上进行实验, 并与其他工作进行比较, 结果如表 10 所示. 我们首先将 LCMC 数据上训练的线性重排序模型和感知器模型应用于人民日报语料. 实验结果表明其性能要稍低于人民日报语料上训练的重排序模型, 但是仍远大于词 N 元语言模型. 这表明我们的级联重排序模型在不同的语料上具有一定的通用性.

表 10 人民日报数据集上的实验结果比较

Table 10 Comparison of different approaches on "People's Daily" dataset

模型	CER (%)
最大熵模型[11]	10.86
基于类别的最大熵马尔科夫模型[12]	5.28
支持向量机模型[13]	7.06
条件随机场模型[14]	11.46
机器翻译方法[19]	4.45
混合字词网格[35]	7.99
词 N 元语言模型	5.48
LCMC 的级联重排序模型	4.74
级联重排序模型	**4.39**
基于压缩的适应算法[3]	4.98
基于频率的在线适应 N 元模型[5]	1.52
错误驱动的适应语言模型[6]	4.44

从表 10 中可以看出, 我们的重排序模型要优于目前已有的非适应学习方法, 包括最大熵模型、条件随机场模型等[11−14, 19, 35], 其中只有机器翻译方法与我们的结果接近[19]. 但文献 [19] 采用的测试数据与我们的并不相同, 采用最大熵模型实验时, 其字错误率为 6.9 %. 而我们的测试数据上只有 10.09 %, 这表明我们的重排序方法要优于该方法. 与采用适应学习方法相比, 我们的方法优于基于压缩的适应算法[3], 但是低于其他两种方法. 这是因为适应学习方法能够根据输入者的反馈添加新词和调整词频, 通常都在测试数据上进行了预处理. 适应学习模型可以做为一个子模型加入到本文的重排序模型中, 使模型的性能得到进一步提高.

6.8 复杂度分析

重排序方法的时间和空间复杂度对于汉语音字转换问题是非常重要的. 从第 4 节的模型描述可以看到, 词性标注模型只使用当前词前面的信息作为特征, 可以随着音字转换的解析过程一起进行标注. 而依存句法分析采用基于转移的方法, 时间复杂度为 $O(l)$, 与句子长度 l 呈线性关系. 所以总的重排序方法也与句子长度 l 呈线性关系. 就空间复杂性来说, 其余子模型的大小都远小于词三元模型.

7 结论

本文针对汉语音字转换问题, 提出了两种重排序方法来改进词 N 元语言模型. 我们的重排序方法能够有效采用词性标注和句法结构中的信息. 线性重排序方法中通过给每个子模型分配相应的权重得到每个候选词序列的概率. 感知器重排序方法可以从候选词序列中选择词性和句法信息作为特征. 在 LCMC 和人民日报数据集上的实验表明, 两种重排序方法都能有效地提高系统的性能. 而将重排序方法进行级联后的性能达到最优. 我们的线性重排序模型还可以将适应学习方法作为子模型, 以进一步提高模型的性能. 本文的方法还可以应用于语音识别等任务.

References

1 Chen S F, Goodman J. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report, Computer Science Group, Harvard University, 1998

2 Brown P F, deSouza P V, Mercer R L, Della Pietra V J, Lai J C. Class-based n-gram models of natural language. *Computational Linguistics*, 1992, **18**(4): 467−479

3 Huang J H, Powers D M. Adaptive compression-based approach for Chinese pinyin input. In: Proceedings of the 3rd SIGHAN Workshop Chinese Language Learning. Barcelona, Spain: Association for Computational Linguistics, 2004. 24−27

4 Wei J, Li P X. Applying the word acquiring algorithm to the pinyin-to-character conversion. In: Proceedings of the 5th International Conference on Natural Computation. Washington, DC, USA: IEEE Computer Society, 2009. 17−21

5 Tang B Z, Wang X L, Wang X, Wang Y H. Frequency-based online adaptive n-gram models. In: Proceedings of the 2nd International Conference on Multimedia and Computational Intelligence. Wuhan, China: IEEE, 2010. 263−266

6 Huang J H, Powers D. Error-driven adaptive language modeling for Chinese pinyin-to-character conversion. In: Proceedings of the 2011 International Conference on Asian Language Processing. Penang, Malaysia: IEEE, 2011. 19−22

7 Pauls A, Klein D. Faster and smaller n-gram language models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, 2011. 258−267

8 Shan Yu-Xiang, Chen Xie, Shi Yong-Zhe, Liu Jia. Fast language model look-ahead algorithm using extended n-gram model. *Acta Automatica Sinica*, 2012, **38**(10): 1618−1626
(单煜翔, 陈谐, 史永哲, 刘加. 基于扩展N元文法模型的快速语言模型预测算法. 自动化学报, 2012, **38**(10): 1618−1626)

9 Siu M H, Ostendorf M. Variable n-grams and extensions for conversational speech language modeling. *IEEE Transactions on Speech and Audio Processing*, 2000, **8**(1): 63−75

10 Wang X, Li L, Yao L, Anwar W. A maximum entropy approach to Chinese pinyin-to-character conversion. In: Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics. Taipei, China: IEEE, 2006. 2956−2959

11 Zhao Y, Wang X L, Liu B Q, Guan Y. Research of pinyin-to-character conversion based on maximum entropy model. *Journal of Electronics*, 2006, **23**(6): 864−869

12 Xiao J H, Liu B Q, Wang X L. Exploiting pinyin constraints in pinyin-to-character conversion task: a class-based maximum entropy markov model approach. *Computational Linguistics and Chinese Language Processing*, 2007, **12**(3): 325−348

13 Jiang Wei, Guan Yi, Wang Xiao-Long, Liu Bin-Quan. Pinyin-to-character conversion model based on support vector machines. *Journal of Chinese Information Processing*, 2007, **21**(2): 100−105
(姜维, 关毅, 王晓龙, 刘秉权. 基于支持向量机的音字转换模型. 中文信息学报, 2007, **21**(2): 100−105)

14 Li L, Wang X, Wang X L, Yu Y B. A conditional random fields approach to Chinese pinyin-to-character conversion. *Journal of Communication and Computer*, 2009, **6**(4): 25−31

15 Wang X L, Chen Q C, Yeung D S. Mining pinyin-to-character conversion rules from large-scale corpus: a rough set approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2004, **34**(2): 834−844

16 Ney H, Essen U, Kneser R. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 1994, **8**(1): 1−38

17 Wang Xuan, Wang Xiao-Long, Zhang Kai. Language model for speech recognition applications. *Acta Automatica Sinica*, 1999, **25**(3): 309−315
(王轩, 王晓龙, 张凯. 语音识别中统计与规则结合的语言模型. 自动化学报, 1999, **25**(3): 309−315)

18 Roark B. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 2001, **27**(2): 249−276

19 Yang S H, Zhao H, Lu B L. A machine translation approach for chinese whole-sentence pinyin-to-character conversion. In: Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation. Bali, Indonesia: Universitas Indonesia, 2012. 333−342

20 Wen J, Wang X J, Xu W Z, Jiang H X. Ambiguity solution of pinyin segmentation in continuous pinyin-to-character conversion. In: Proceedings of the 2008 International Conference on Natural Language Processing and Knowledge Engineering. Beijing, China: IEEE, 2008. 1−7

21 Chen Z, Lee K F. A new statistical approach to Chinese pinyin input. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong: Association for Computational Linguistics, 2000. 241−247

22 Zheng Y B, Li C, Sun M S. CHIME: an efficient error-tolerant Chinese pinyin input method. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. Barcelona, Catalonia, Spain: AAAI Press, 2011. 2551 −2556

23 Collins M. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing. Philadelphia, PA, USA: Association for Computational Linguistics, 2002. 1−8

24 Li X X, Wang X, L Yao Y. Joint decoding for Chinese word segmentation and POS tagging using character-based and word-based discriminative models. In: Proceedings of the 2011 International Conference on Asian Language Processing (IALP). Washington, DC, USA: IEEE, 2011. 11−14

25 Ng H T, Low J K. Chinese part-of-speech tagging: one-at-a-time or all at once? word-based or character-based? In: Proceedings of the 2004 EMNLP. Barcelona, Spain: Association for Computational Linguistics, 2004. 277−284

26 Zhang Y, Clark S. Joint word segmentation and POS tagging using a single perceptron. In: Proceedings of ACL-08: HLT. Columbus, Ohio: Association for Computational Linguistics, 2008. 888−896

27 Zhang Y, Nivre J. Transition-based dependency parsing with rich non-local features. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, 2011. 188 −193

28 Liu Di, Sun Dong-Mei, Qiu Zheng-Ding. Feature level fusion based on speaker verification via relation measurement Fusion framework. *Acta Automatica Sinica*, 2011, **37**(12): 1503−1513
(刘镝, 孙冬梅, 裘正定. 一种基于关系度量融合框架的说话人认证特征级融合算法. 自动化学报, 2011, **37**(12): 1503−1513)

29 Och F J. Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Sapporo, Japan: Association for Computational Linguistics, 2003. 160−167

30 Jiang W B, Huang L, Liu Q, Lü Y J. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In: Proceedings of ACL-08: HLT. Columbus, Ohio: Association for Computational Linguistics, 2008. 897−904

31 Zaidan O. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 2009, **91**(1): 79−88

32 Stolcke A. SRILM — an extensible language modeling toolkit. In: Proceedings of the 2002 International Conference on Spoken Language Processing. Denver, Colorado: IEEE 2002. 901−904

33 Liu W, Guthrie L. Chinese pinyin-text conversion on segmented text. In: Proceedings of the 12th International Conference on Text, Speech and Dialogue. Berlin, Heidelberg: Springer-Verlag, 2009. 116−123

34 Zhou X H, Hu X H, Zhang X D, Shen X J. A segment-based hidden Markov model for real-setting pinyin-to-Chinese conversion. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management (CIKM 2007). New York, NY, USA: ACM Press, 2007. 1027 −1030

35 Zhang Sen. Solving the pinyin-to-Chinese-character conversion problem based on hybrid word lattice. *Chinese Journal of Computers*, 2007, **30**(7): 1145−1153
(章森. 基于混合字词网格的汉语音字转换问题的求解. 计算机学报, 2007, **30**(7): 1145−1153)

李鑫鑫 哈尔滨工业大学深圳研究生院博士研究生. 主要研究方向为自然语言处理, 网络信息处理. 本文通信作者.
E-mail: lixxin2@gmail.com
(**LI Xin-Xin** Ph. D. candidate at Harbin Institute of Technology Shenzhen Graduate School. His research interest covers natural language processing, and network information processing. Corresponding author of this paper.)

王 轩 哈尔滨工业大学深圳研究生院教授. 主要研究方向为人工智能, 网络多媒体信息处理.
E-mail: wangxuan@insun.hit.edu.cn
(**WANG Xuan** Professor at Harbin Institute of Technology Shenzhen Graduate School. His research interest covers artificial intelligence, network multimedia information processing.)

姚 霖 哈尔滨工业大学软件学院讲师. 主要研究方向为网络信息处理, 生物信息处理. E-mail: yaolin@hit.edu.cn
(**YAO Lin** Lecturer at Harbin Institute of Technology. Her research interest covers network information processing and biology information processing.)

关 键 哈尔滨工业大学深圳研究生院博士研究生. 主要研究方向为人工智能, 语音识别.
E-mail: guanjian2000@gmail.com
(**GUAN Jian** Ph. D. candidate at Harbin Institute of Technology Shenzhen Graduate School. His research interest covers artificial intelligence and speech recognition.)