doi:10.3772/j.issn.1002-0470.2010.05.006

基于 Lattice 分段的高质量混淆网络快速生成方法①

王欢良②* **　韩纪庆③*

(*哈尔滨工业大学计算机科学与技术学院　哈尔滨 150001)

(**青岛科技大学信息学院　青岛 266042)

摘　要　针对现有混淆网络生成方法难以兼顾速度和质量的问题,研究了基于横断一致性的 Lattice 分段方法和基于最大置信度的 Lattice 分段方法,研究了用这两种 Lattice 分段方法来减少对混淆网络质量的影响。提出了一种基于 Lattice 分段的高质量混淆网络快速生成方法。该方法把原始大规模 Lattice 分割成小尺寸的 Lattice,分别生成混淆网络,从而可减小计算规模,提高网络生成速度。同时通过分段数目来调节速度和质量之间的平衡。实验结果显示,与词聚类算法相比,所提方法显著提高了混淆网络的生成速度,而对混淆网络质量影响很小。从解码性能看,在相同速度下所提方法获得了比采用剪枝的词聚类算法更低的错误率。

关键词　混淆网络(CN), Lattice, 多候选, 语音识别

0　引　言

混淆网络(confusion network, CN)是一种更为紧凑的多候选结果的表示形式。它是在给定准则下对齐 Lattice 中所有路径对应的句子后形成的一种结构,具有比 Lattice 更多的优点[1]。首先,混淆网络解码可以单独考察多个相互竞争的词级候选,而 Lattice 解码只能对整个候选句子进行决策;其次,混淆网络近似于一种线性结构,比 Lattice 更易融合复杂的高级知识。此外,混淆网络是原始 Lattice 的一个超集。因此,混淆网络在语音识别相关领域已获得广泛应用。

Mangu 等人以最小化词错误率为目标提出了基于混淆网络的解码方法[1]。所提出的从 Lattice 生成混淆网络的方法,称为词聚类算法。研究表明,混淆网络解码可以获得比基于句子最大后验概率(maximum A posteriori, MAP)的解码方法更低的词错误率。混淆网络解码是分段最小贝叶斯风险解码(segmental minimum bayes risk, SMBR)的一个特例[2]。除了用于最小化错误率,Hakkani-Tür 等人把混淆网络成功应用于口语语言理解任务[3]。Hillard 等人研究了利用混淆网络进行句子边界检测的方法[4]。此外,混淆网络还被用于口语语音翻译[5]、关键词检出[6]、置信度估计[7]、手写识别和语音识别的错误检测[8,9]等任务。词聚类算法在给定准则下可保证全局最优,但其计算复杂度很高($O(N^3)$, N 是 Lattice 中转移弧的数目)[1]。在实际应用中,需要先对 Lattice 进行剪枝,这必然会带来损失。Hakkani-Tür 等人提出了一个称为轴对齐的快速生成算法[10],其时间复杂度为 $O(N \times K)$(K 为参考路径长度加上插入的空转移弧和平均的发射弧数)。它提高了生成速度,但混淆网络质量完全依赖于参考路径选择的好坏。Xue 等人提出的快速生成算法只需遍历一遍 Lattice,时间复杂度是线性的[11]。尽管该算法的复杂度很低,但在路径之间的弧数或时间点差异较大时,将会带来严重的对齐错误。以上混淆网络生成方法难以控制速度和质量之间的平衡。为此,本文提出了一种基于 Lattice 分段的混淆网络快速生成方法,它通过分段来缩减 Lattice 的尺寸,从而可提高混淆网络生成速度。本文重点研究了两种 Lattice 分段的方法,尽可能减少分段带来的对齐错误。为了更直接地评价混淆网络质量,本文定义了一种新的质量评价指标。最后,在中文连续语音数据库上,从生成速度和质量两个方面评估了本文所提出的方法。

① 863 计划(2006AA010103)和国家自然科学基金(60672163)资助项目。
② 男,1974 年生,博士,讲师;研究方向:语音识别;E-mail: huanliangwang@126.com
③ 通讯作者,E-mail: jqhan@hit.edu.cn
(收稿日期:2009-02-11)

1 基于 Lattice 分段的混淆网络生成

词聚类算法是当前生成混淆网络的最佳算法。但当弧数目较多时,其计算代价和存储需求都将急剧增加,通常需要使用剪枝来缩减 Lattice 尺寸。但剪枝可能会抛弃正确的候选结果。此外,由于空转移弧的后验概率等于 1 减去混淆集合中所有弧的后验概率之和,因此剪枝可能会导致空转移弧的概率无法准确计算。通常的情况是空转移弧概率变大,从而导致混淆网络解码时删除错误增多。

设原始 Lattice 具有 N 条转移弧,把它分段为 M 个小的 Lattice,其弧数目依次为 $k_1,\cdots,k_M$,且 $\sum_{m=1}^{M} k_m = N$,采用词聚类算法分别生成混淆网络,则

$$N^3 = \left(\sum_{m=1}^{M} k_m\right)^3 \geqslant \sum_{m=1}^{M} k_m^3 \qquad (1)$$

上式表明,Lattice 分段可以缩减计算规模,提高生成速度。当 $k_1 = k_2 = \cdots = k_M$ 时,计算规模最多可以缩减为原来的 $1/M^2$。基于 Lattice 分段的混淆网络生成方法的基本思想就是:如果可以把一个大的 Lattice 分段成多个小的相互独立的 Lattice,然后对每个小的 Lattice 采用词聚类算法生成混淆网络,那么在保证质量的前提下可以成倍地缩减计算规模。

一个 Lattice 是一个有向无环图,可用一个五元组 $L = \{Q,\Lambda,p_s,F,E\}$ 来表示,其中 Q 为节点集合,每个节点 q 都唯一对应一个时间点 $t(q)$,Λ 为转移弧上标号的集合,p_s 为 Lattice 的起始节点,F 为结束节点的集合,E 为转移弧的集合。$e = (p_e, q_e, w_e, s_e) \in E$ 为一条始节点为 $p_e \in Q$ 且末节点为 $q_e \in Q$ 的转移弧,其转移代价为 s_e,其标号为 w_e。Lattice 中的一条路径可以表示为 $h = \{e_i\}_{i=1}^{I} = \{(p_{e_i}, q_{e_i}, w_{e_i}, s_{e_i})\}_{i=1}^{I} \in H$,其中 $p_{e_1} = p_s, p_{e_{i+1}} = q_{e_i}, q_{e_I} \in F$,与该路径对应的句子为 $W = w_{e_1}\cdots w_{e_I}$,观察到该句子的对数似然度为 $-\sum_{i=1}^{I} s_{e_i}$。Q_h 表示路径 h 上所有节点的集合,H 表示 Lattice 中所有路径的集合。

定义 1: 如果 $\exists h \in H$ 使得转移弧 $e_i, e_j \in h$,则称这两条弧之间存在时序关系。

定义 2: 对于两个节点 $q, p \in Q$,如果 $\exists h \in H$ 使得 $q, p \in Q_h$,且 $t(q) < t(p)$,则称节点 q 是 p 的前缀节点。

定义 3: 设 $E_c \subset E$ 是转移弧的一个子集,满足如下三个性质:

P1: E_c 中的转移弧之间不存在时序关系;

P2: $\exists e \in E_c$,且 $p_e \neq p_s$,同样,$\exists e \in E_c$,且 $q_e \notin F$;

P3: $\forall h \in H$,$\exists e \in E_c$,且 $e \in h$,

则称 E_c 为 Lattice 的一个分段弧集合,它构成了对 Lattice 的一个时序分段。E_c 把 Lattice 按时序分段成了三部分:从 p_s 到 E_c 中所有弧的始节点的路径构成了第一部分,E_c 本身构成了第二部分,从 E_c 中所有弧的末节点到 Lattice 结束节点的所有路径构成第三部分。根据性质 P1 和 P3,E_c 直接构成一个混淆集合。通过对前后两部分添加辅助的空节点,就形成了两个小的 Lattice。因此本文称 E_c 把 Lattice 分段为两个小的 Lattice。图 1 给出了利用分段方法从音节 Lattice 生成混淆网络的一个例子。其中图(b)和(c)给出了对 Lattice 进行时序分段的一个例子。

对于 $\forall e_i \in E_c$,如果 $\exists e_j \in E_c$ 使得 e_i 的始节点是 e_j 始节点的前缀节点,则在第一个 Lattice 中 e_i 所在的位置添加一条空转移弧,设置它的转移代价为 e_i 的转移代价;同样,如果 e_j 的末节点是 e_i 末节点的前缀节点,则在第二个 Lattice 中同样添加一条空转移弧。图 2 展示了一个这样的例子。

定义 4: 分段得到的两个 Lattice 和分段弧集 E_c 中的转移弧不同时出现在原始 Lattice 生成的混淆网络的同一个混淆集合中,则称分段为 Lattice 的无损分段。对 Lattice 进行无损分段不会影响混淆网络的质量。在以上定义的基础上,基于 Lattice 分段的混淆网络生成方法的三个步骤可简单描述如下:

(1)把原始 Lattice 按照时序尽可能无损地分段成 M 个小的相互独立的 Lattice;

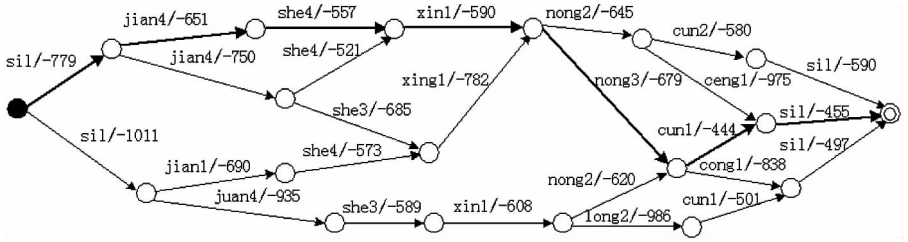
(2)采用词聚类算法(也可采用其他算法,但不保证全局最优)把每个小的 Lattice 转化为混淆网络;

(3)按照时间顺序把所有小的混淆网络串联起来,形成原始 Lattice 对应的混淆网络。

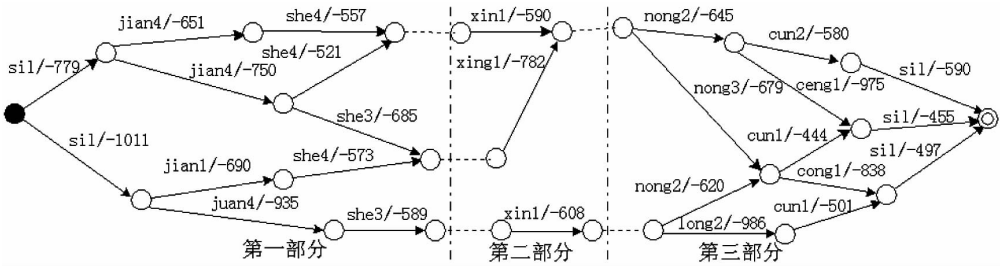
显然,通过分段数目 M 可直接控制生成速度和质量之间的平衡。图 1 为基于 Lattice 分段的混淆网络生成过程的一个完整示例。

2 Lattice 分段方法

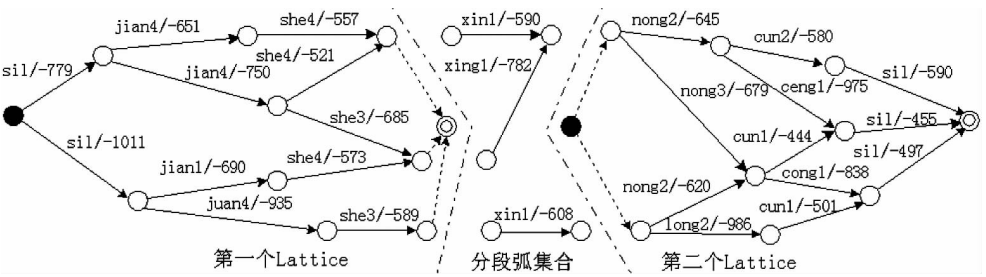
Lattice 的无损分段实质上要寻找一个合适的分段弧集 E_c,使得它本身可以构成原始 Lattice 生成的混淆网络中的一个完整的混淆集合。下面给出两种寻找 E_c 的方法,使得它对 Lattice 的分段尽可能无损,从而尽量减少分段对混淆网络质量的影响。



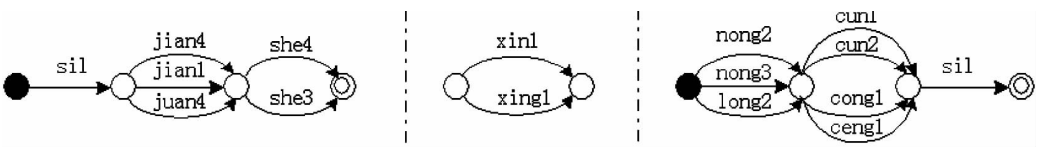
(a)原始 Lattice，其中弧上的标记为音节/对数似然度得分,粗线路径对应基于 MAP 的句子



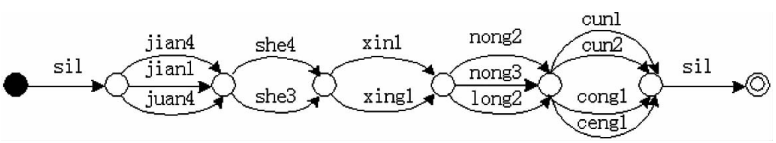
(b)分段弧集合把 Lattice 按照时序分为三部分,其中第二部分对应分段弧集合



(c)通过添加辅助节点,分段弧集合前后的两部分构成两个小的 Lattice



(d) 两个小的 Lattice 分别生成对应的混淆网络



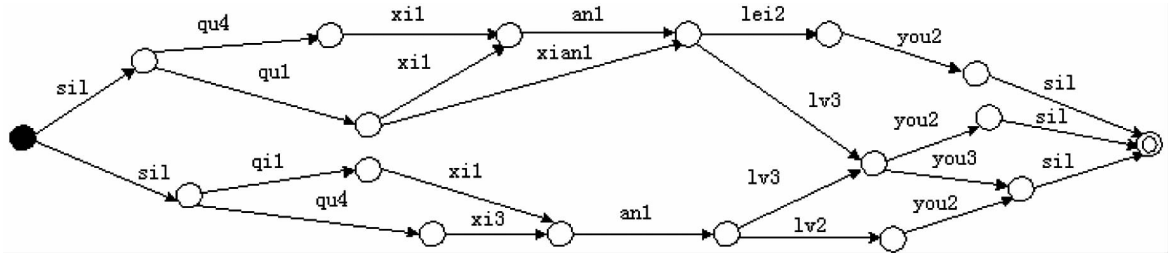
(e)串联多个小的混淆网络生成最后的混淆网络

图 1　利用分段方法从音节 Lattice 生成混淆网络的一个例子

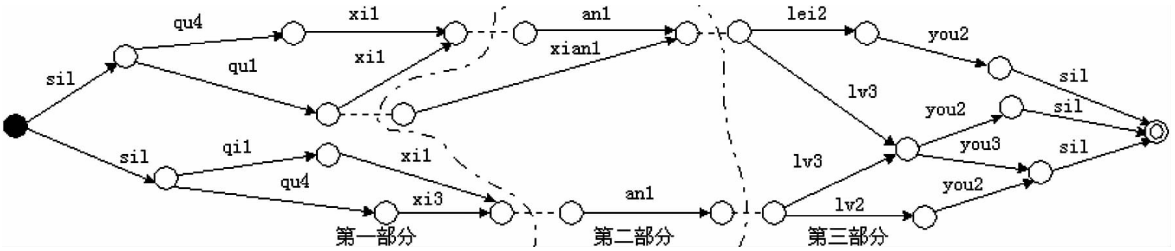
2.1　基于横断一致性的分段方法

在一段足够长的时间内,如果 Lattice 中的所有路径都对应一条转移弧,并且这些转移弧的标号有较高的相似性,则这些转移弧通常会构成一个混淆集合。基于横断一致性的分段方法首先投影 Lattice 到时间轴,每对相邻投影节点构成一个时间窗,然后考察每个时间窗内的所有转移弧。弧标号的相似性越高且窗越长,则该时间窗内所有转移弧构成的 E_c 可以产生无损分段的可能越大。详细算法描述如下:

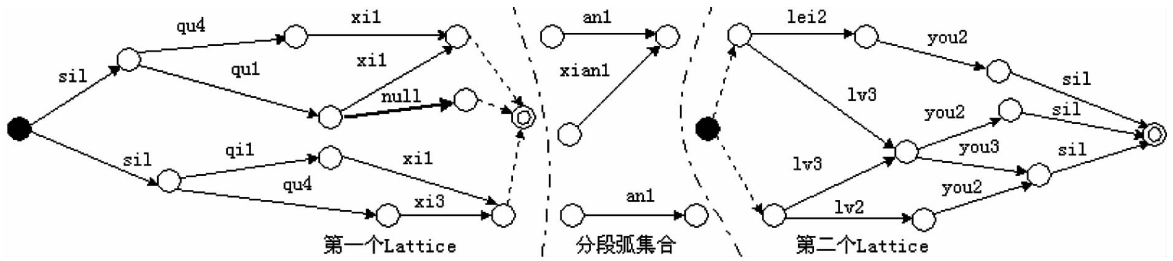
(1)投影 Lattice 到时间轴,如图 3 所示。每对相邻投影节点组成的时间窗构成了对 Lattice 的一个横向截断。



(a) 原始的音节 Lattice



(b) 利用分段弧集合把 Lattice 按照时序分为三部分,其中第二部分对应分段弧集合



(c) 把原始 Lattice 分段为两个小的 Lattice,其中粗线弧为添加的空转移弧

图 2 分段后需要添加空转移弧的一个例子

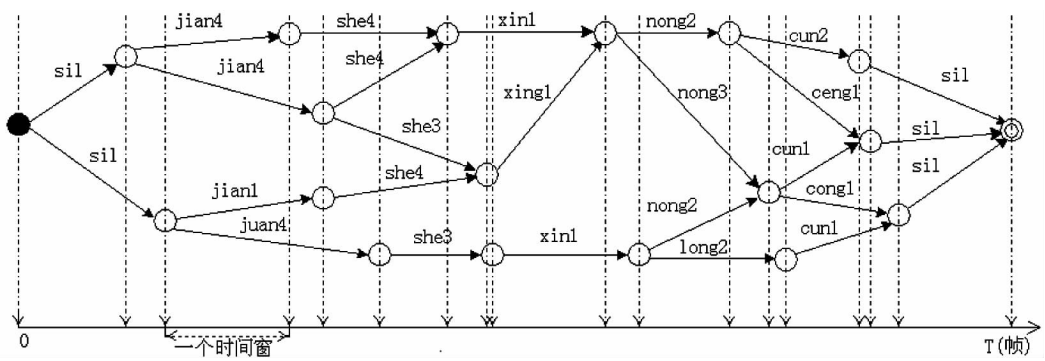


图 3 Lattice 向时间轴的投影

(2)对于那些窗长大于给定阈值(通常设置为最短弧长的一半)的时间窗,计算式

$$D(E_l)=\begin{cases}\dfrac{1}{N_l(N_l-1)}\sum\limits_{\substack{\forall e_i,e_j\in E_l,\\ i\neq j}} sim(e_i,e_j), & N_l>1\\ 1, & N_l=1\end{cases} \tag{2}$$

$$G=\frac{T_l}{T}D(E_l) \tag{3}$$

其中 T_l 为第 l 个时间窗的长度, T 为语音的总长度, E_l 为该时间窗内所有转移弧的集合, $N_l=|E_l|$。$sim(e_i,e_j)\in[0,1]$ 为转移弧 e_i 和 e_j 所对应标号的相似性,且 $sim(e_i,e_j)=sim(e_j,e_i)$。对于空转移弧 e_ε, 设置 $sim(e_i,e_\varepsilon)=1$。

(3)在开始和结束的两个时间窗之外,选择 G 值最大的时间窗,如果不存在,则该 Lattice 不能继续分段,转到第(5)步。

(4)用时间窗内的所有转移弧构造 E_c,并把 Lattice 分段为两个小的 Lattice;

(5)如果分段数未达到 M,且存在未尝试分段的 Lattice,则从中选择转移弧数最多的一个,返回第(3)步进行继续分段。

该算法的时间复杂度为 $O(L \times \hat{N})$,其中 L 为时间窗的数目,$\hat{N} = \max_{1 \leqslant l \leqslant L} N_l$。

2.2 基于最大置信度的分段方法

在 Lattice 中置信度较高的转移弧在混淆网络中对应的混淆集合通常较小,即它的竞争弧相对要少。因此,以置信度高的转移弧为参照构建一个混淆集合,然后作为分段弧集合是比较可靠和容易实现的。基于最大置信度的分段方法首先从 Lattice 中寻找置信度最高的转移弧来初始化分段弧集合,然后直接寻找可与该转移弧构成混淆集合的所有其它转移弧。在生成混淆网络时,通常利用前后向算法预先计算每条转移弧上的后验概率作为其置信度[1]。详细算法描述如下:

(1)对于当前 Lattice,从起始和结束转移弧之外的其他转移弧中,选择后验概率最大的转移弧 e。如果不存在,则该 Lattice 不能继续分段,转到第 5 步,否则 $e \rightarrow \hat{E}$。设 e 所在时间窗内所有转移弧的集合为 $\tilde{E}$。

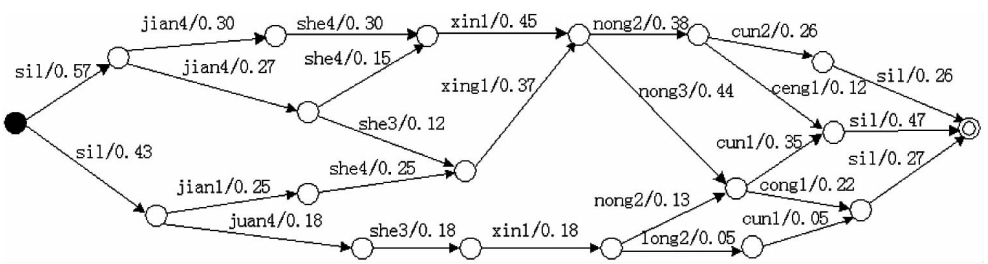
(2)根据式

$$\tilde{e} = \arg\max_{e_i \in \tilde{E}, e_i \notin \hat{E}} [P(e_i)\, overlap(e_i, e)\, sim(e_i, e)] \tag{4}$$

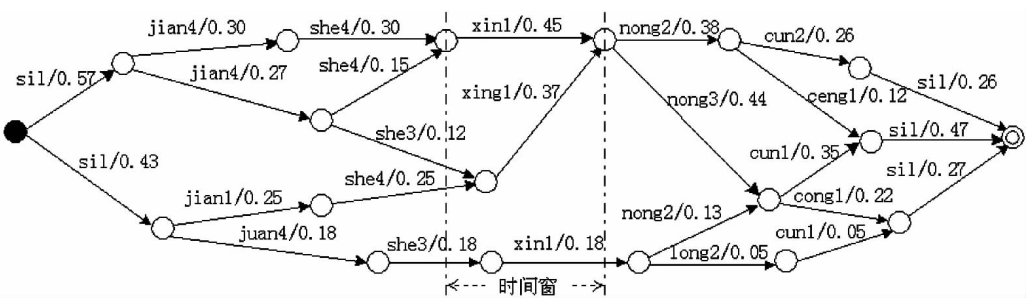
寻找转移弧 $\tilde{e}$。其中 $overlap(e_i, e) \in [0,1]$ 为转移弧 e_i 和 e 的时间重叠程度,$sim(e_i, e)$ 的定义与前面一致,$P(e_i)$ 为转移弧 e_i 的后验概率。

(3) $\tilde{E} \leftarrow \tilde{E} \setminus \tilde{e}$,如果 $\tilde{e}$ 与 $\hat{E}$ 中的弧不存在时序关系,则 $\tilde{e} \rightarrow \hat{E}$。

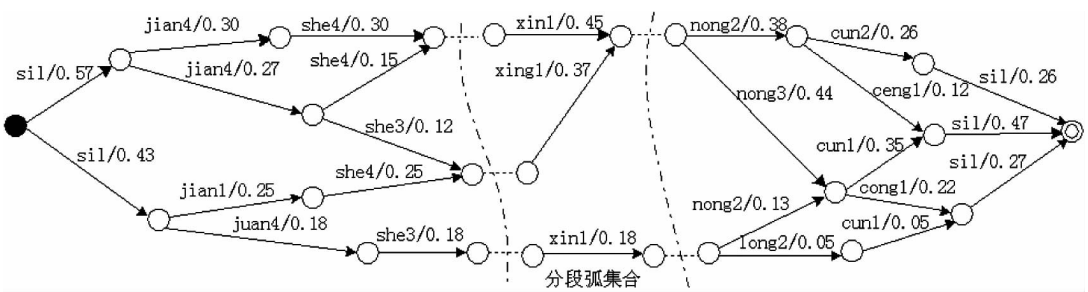
(4)如果 $|\tilde{E}| > 0$,转到第(2)步,否则 $\hat{E}$ 构成一个分段集合,从而把 Lattice 分段为两个小的 Lattice。



(a) 原始 **Lattice**,其中每条弧上的标号是音节/置信度



(b) 最大置信度转移弧(**xin1/0.45**)所对应的时间窗



(c) 根据最大置信度转移弧构造的分段弧集合

图 4　基于最大置信度的分段方法的一个示例

(5)如果分段数未达到 M，且存在未尝试分段的 Lattice，则从中选择转移弧数最多的一个，返回第(1)步进行继续分段；否则，结束。

该算法的时间复杂度为 $O(M \times \widetilde{N})$，其中 $\widetilde{N}$ 为一个时间窗内最大的转移弧数。图 4 给出了基于最大置信度的分段方法的一个示例。

3　实验与分析

3.1 实验配置

训练和测试数据集均来自微软亚洲研究院[12]。训练集由来自 250 名男性话者的 5 万句话组成。基本音素集由声母和带调韵母组成，共 187 个，采用 3 状态隐马尔可夫模型(HMM)建模。声学模型为上下文相关的跨音节的三音素(triphone)模型，共有 5000 个绑定状态，每状态 16 个混合高斯。测试集由来自 25 名男性话者的 500 句话组成，平均句子长度为 19 个字。特征采用 39 维 Mel 频率倒普系数在语音识别中(MFCC)特征，由对数能量和 12 维 MFCC 及其一阶和二阶差分组成。在测试集上，采用上下文无关的音节解码器来生成有调音节的 Lattice。这些 Lattice 的平均密度为 335 条转移弧/音节。基于最大后验概率(MAP)解码的有调音节错误率是 45.8%，而 Lattice 的 Oracle 错误率为 12.2%。后面的实验统计在这 500 个有调音节 Lattice 上进行，结果是它们的平均。

3.2　混淆网络质量评价指标

用来评价混淆网络质量的指标通常有两个：解码错误率和 Oracle 错误率。前者是从每个混淆集合中选择后验概率最大的标号，串联后所得结果的错误率，后者是混淆网络中与参考句子最相似句子的错误率。语音识别中的错误率定义为

$$\text{错误率} = \frac{\text{替换错误数} + \text{插入错误数} + \text{删除错误数}}{\text{总测试单元数}} \tag{4}$$

错误率的计算都需要利用语音的参考标注，且通常只适合于那些以最小化词错误率为目标的任务，而语音文档检索和智能语音对话等任务并不是以最小化词错误率为目标的，因此错误率并不能直接从本质上反映混淆网络质量。混淆网络是多条路径的全局对齐，对齐所产生的全局失真才直接反映了混淆网络的质量。这里定义混淆网络的全局失真作为其质量评价的客观指标。首先合并每个混淆集合中带有相同标号的转移弧，把它们的后验概率相加作为合并后转移弧的后验概率。然后，计算混淆网络的全局失真

$$D = \sum_{k=1}^{K}(1 - D(C_k)) \tag{5}$$

其中 K 为混淆集合的总数，C_k 为第 k 个混淆集合，$D(C_k)$ 的定义同式(2)。显然，全局失真 D 越小，混淆网络的质量越高。

3.3　分段数目对生成速度和质量的影响

分段数目控制着混淆网络的生成速度，也影响着它的质量。图 5 给出了在不同分段数目下，采用两种分段方法，生成速度和质量的变化情况。分段数为 1 表示不分段直接采用词聚类算法生成混淆网络。生成速度采用相对耗时，以分段数为 1 时的时间耗费为基准。

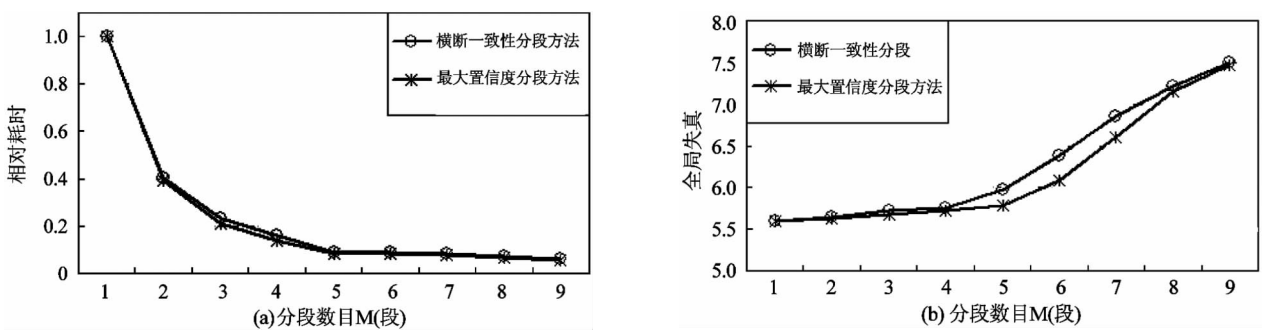


图 5　在不同分段数目下，采用两种分段方法的生成速度和质量的变化情况

从图 5(a)可以看出，随着分段数目的增加，生成混淆网络的相对耗时在不断减少，这说明分段可以提高生成速度。但随着分段数目增多，相对耗时减少的幅度趋于平缓。这主要因为分段较小的 Lattice 对计算规模的缩减是有限的。从图 5(b)可以看出，随着分段数目的增加，混淆网络的全局失真在不断增长，这说明分段会影响混淆网络质量。因此，分段数目可以控制速度和质量之间的平衡。当分段数

目超过 5 后,时间耗费趋于平缓而全局失真急剧增加。从图上也可以看到,在速度上两个算法相差不大,但在质量上,基于最大置信度的分段方法要更好一些。下面的实验中,我们只采用分段数目为 5 的最大置信度分段方法。

3.4 不同生成方法的比较

这里从速度和质量两个方面对轴对齐方法、Xue 的快速算法和分段方法进行比较。图 6 对不同生成方法在速度和质量上作了比较,从图中可以看到,分段方法在生成速度上没有其它方法快,但在生成质量上却明显要优于其它方法。实际上,尽管分段方法所耗费的时间大于快速算法(在 3GHz 奔腾 4 处理器上,$<10^{-2}$倍实时),但其速度已可以满足实际任务的需要。

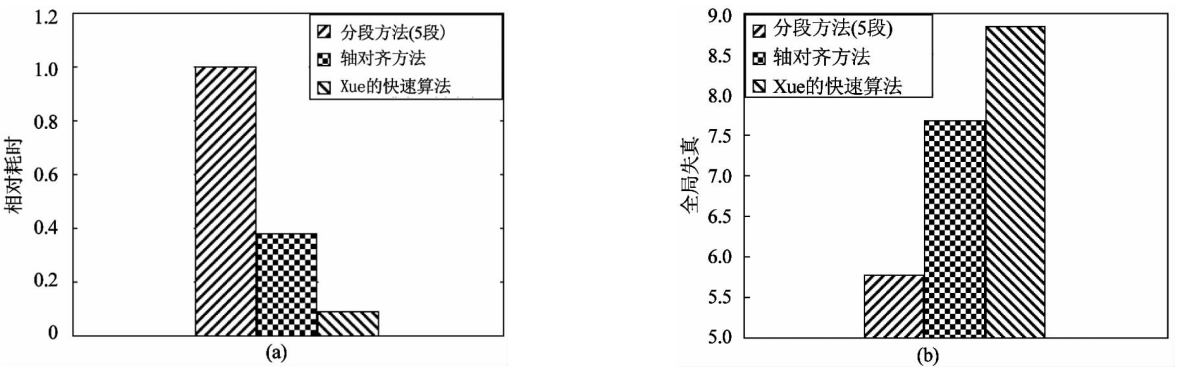


图 6 不同生成方法在速度和质量上的比较

3.5 不同方法的解码性能

本实验用解码性能来评估不同方法生成混淆网络的质量。从每个混淆集合中选择后验概率最大的转移弧,把这些弧的标号按顺序联接起来作为混淆网络解码的结果。用有调音节错误率作为解码性能指标。表 1 给出了不同方法生成混淆网络的解码性能。

表 1 不同方法生成的混淆网络的解码性能

	词聚类算法(不分段)			轴对齐方法	Xue 的快速算法	分段方法(5 段)
剪枝幅度(%)	0	70	95	0	0	0
有调音节错误率(%)	45.2	45.6	46.0	46.2	46.9	45.3

从表 1 可以看到,词聚类算法在原始 Lattice 上生成的混淆网络获得了最好的解码性能。分段数为 5 的分段方法获得的混淆网络在解码性能上要优于轴对齐方法和 Xue 的快速算法。这和 3.4 节实验中混淆网络的质量指标是一致的,这说明全局失真可以作为混淆网络质量评估的客观指标。Lattice 剪枝会导致混淆网络解码性能的下降。当剪枝幅度为 70%时,词聚类算法生成混淆网络所耗费的时间和分段数为 5 的分段方法耗费的时间相当,但后者的性能要优于前者。这说明,在相同生成速度下,分段方法带来的损失要小于剪枝方法。剪枝幅度为 95%的词聚类算法和 Xue 的快速算法生成的混淆网络的解码性能甚至低于基于句子级 MAP 的解码性能。主要原因是过度剪枝会剪掉正确的候选结果,且会增加删除错误,而 Xue 的快速算法本身会导致很多严重的对齐错误。

4 结论

本文提出了基于 Lattice 分段的高质量混淆网络快速生成方法。这种方法通过分段来缩减 Lattice 尺寸,降低计算规模,从而提高生成速度。主要研究了基于横断一致性的分段方法和基于最大置信度的分段方法。这两种方法直接从原始 Lattice 中寻找混淆集合作为分段边界,因此在提高生成速度的同时最大限度地保证了生成混淆网络的质量。实验结果显示,这种生成方法可以通过分段数目来控制生成速度和质量的平衡。分段数为 5 的分段方法生成的混淆网络质量要优于轴对齐方法和 Xue 的快速算法,而生成速度也是可以接受的。在相同速度下,分段

方法在解码性能上要优于剪枝方法。所提出的基于混淆网络全局失真的质量评价指标和基于解码性能的评价指标是一致的。

参考文献

[1] Mangu L, Brill E, Stolcke A. Finding consensus in speech recognition: word error Minimization and other applications of confusion networks. *Computer Speech and Language*, 2000, 14(4): 373-400

[2] Goel V, Kumar S, Byrne W. Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2004, 12(3): 234-249

[3] Hakkani-Tür D, Béchet F, Riccardi G, et al. Beyond ASR 1-best: using word confusion networks in spoken language understanding. *Computer Speech and Language*, 2006, 20(4): 495-514

[4] Bertoldi N, Zens R, Federico M. Speech translation by confusion network decoding. In: Proceedings of the 2007 International Conference on Acoustics, Speech, and Signal Processing, Honolulu, USA, 2007. 1297-1300

[5] Hillard D, Ostendorf M, Stolcke A, et al. Improving automatic sentence boundary detection with confusion networks. In: Proceedings of the 2004 Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics annual meeting, Boston, USA, 2004. 69-72

[6] Shao J, Zhao Q W, Zhang P Y, et al. A fast fuzzy keyword spotting algorithm based on syllable confusion network. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 2007. 2405-2408

[7] Hillard D, Ostendorf M. Compensation forward posterior estimation bias in confusion networks. In: Proceedings of the 2006 International Conference on Acoustics, Speech, and Singnal Processing, Toulouse, France, 2006. 1153-1156

[8] Quiniou S, Anquetil E. Use of a confusion network to detect and correct errors in an on-line handwritten sentence recognition system. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, 2007. 382-386

[9] Allauzen A. Error detection in confusion network. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 2007. 1749-1752

[10] Hakkani-Tür D, Riccardi G. A general algorithm for word graph matrix decomposition. In: Proceedings of the 2003 International Conference on Acoustics, Speech, and Singnal Processing, Hong Kong, China, 2003. 596-599

[11] Xue J, Zhao Y-X. Improving confusion network algorithm and shortest path search from word lattice. In: Proceedings of the 2005 International Conference on Acoustics, Speech, and Singnal Processing, Philadelphia, USA, 2005. 853-856

[12] Chang E, Shi Y, Zhou J L, et al. Speech lab in a box: a Mandarin speech toolbox to jumpstart speech related research. In: Proceedings of the 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, 2001. 2779-2782

The fast generation method based on lattice segmentation for high-quality confusion network

Wang Huanliang[*,**], Han Jiqing[*]

([*] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

([**] School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266042)

Abstract

Aimed at the problem that the existing confusion network generating methods cannot keep a tradeoff between the network generation speed and the quality of confusion network, the paper investigates two major lattice segmentation methods with the purpose of using them to reduce the impacts of segmentation to the quality of confusion networks, and based on this, presents a high-quality method for fast generating confusion networks based on lattice segmentation. The method segments the large-scale lattice from automatic speech recognition (ASR) into sequences of smaller sub-lattices and then generates the confusion networks from these sub-lattices, thus remarkably decreasing the computation scale and increasing the network generating speed. The balance between the generation speed and the network quality is controlled by the segmentation number. The experimental results show that the proposed method can significantly improve the speed of confusion network generation while hold almost the same quality compared with the traditional word-clustering method without lattice segmentation. At the same speed, the proposed method can obtain a lower tonal syllable error rate than the word-clustering method with lattice pruning.

Key words: confusion network, lattice, multi-candidates, speech recognition