ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 30,No. 2,Mar. 2015,pp. 390－398
DOI:10.16337/j.1004-9037.2015.02.018

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:＋86-025-84892742

基于加权有限状态转换器的语音查询项检索技术

陆梨花[1,2]　张连海[1]　陈　琦[1]

(1.解放军信息工程大学信息系统工程学院,郑州,450001; 2.中国人民解放军95826部队,上海,201419)

摘　要:为了提高语音查询项检索效率,提出了一种在加权有限状态转换器(Weighted finite-state transducer,WFST)框架下以混淆网络代替词格建立索引的技术。在索引建立阶段,首先将词格转化为混淆网络并用自动机形式表示,然后利用自动机构建基于时间的因子转换器,最后将所有因子转换器进行联合及优化得到索引。在查询阶段,将查询项转化为自动机形式后与索引进行合成运算得到表示查询结果的自动机。实验结果表明,在保证系统检测正确率的前提下,与直接以词格建立的WFST索引相比,以混淆网络建立的WFST索引尺寸更小,检索速度更快,因而系统性能更好。

关键词:加权有限状态转换器;语音查询项检索;混淆网络;因子转换器

中图分类号:TP391　　**文献标志码**:A

Spoken Term Detection Techniques Based on Weighted Finite-State Transducer

Lu Lihua[1,2], Zhang Lianhai[1], Chen Qi[1]

(1. Institute of Information System Engineering, PLA Information Engineering University, Zhengzhou, 450001, China; 2. 95826 Troops, PLA, Shanghai, 201419, China)

Abstract: An indexing method based on confusion network instead of Lattice is proposed in the weighted finite-state transducer framework (WFST) to improve the efficiency of the spoken term detection system. In the indexing stage, firstly confusion networks are extracted from Lattices and transformed to automatons; Then, timed factor transducers are constructed with these automatons; Finally, the index is achieved by taking the union of the factor transducers and optimizing the union. In the searching stage, the queries are transformed to automatons and then composed with the index. After optimization, the automaton representing the searching results is obtained. Experimental results show that compared with the WFST index based on Lattice, the confusion network-based index has smaller index size, faster searching speed and better performance when ensuring the retrieval accuracy.

Key words: weighted finite-state transducer; spoken term detection; confusion network; factor transducer

引　　言

随着互联网和多媒体技术的发展,语音数据大量积累,如何对其进行高效检索成为亟待解决的问题。语音查询项检索(Spoken term detection,STD)是一种关键的信息检索技术,它是指在大量的语音

基金项目:国家自然科学基金(61175017)资助项目。
收稿日期:2013-05-05;**修订日期**:2013-10-22

资料中检索用户查询项并返回相关信息的过程,处理目标是开集大词汇表的查询项检索任务。系统的实现分为两个过程:建立索引库和查询。为了实现快速检索,要求根据识别结果建立起适合查询的索引。在索引建立阶段,通常先通过自动语音识别(Automatic speech recognition,ASR)技术对语音数据进行处理,然后利用解码输出结果建立索引文件。在查询阶段,对用户提出的查询请求处理后在索引中进行检索,每次查询需要很高的实时性。最后根据门限进行判决并输出结果。

然而,非特定领域和非特定环境下的自动语音识别性能仍然不能令人满意,严重影响系统检索的准确率[1]。词格(Lattice)是一种多候选的语音识别结果。它不仅能够补偿识别错误带来的影响,而且能够提供用于置信度计算的声学得分和语言得分。因此,利用 Lattice 建立索引成为目前广泛采用的方法。文献[2]提出直接存储 Lattice 本身作为索引,然后在 Lattice 上搜索与查询项匹配的局部路径,并计算该局部路径的后验概率作为相关度。基于动态匹配词格检测(Dynamic match lattice spotting,DMLS)的检索方法[3],在查询项搜索时利用最小编辑距离来计算 Lattice 中某段局部路径与查询项之间的相似度。针对 Lattice 结构较为复杂的特点,以 Lattice 的改进形式如混淆网络(Confusion network,CN)[4]或标明位置的后验概率词格(Position specific posterior lattices, PSPL)[5]等进行语音检索能够去除大量冗余信息,提高检索效率。

由于 Lattice 为加权有向无环图结构,实现快速检索需要找到符合其结构特点的检索方法。加权有两状态转换器(Weighted finite-state transducer,WFST)可以为这种图形结构的表示和使用提供有效框架,并利用优化算法对搜索网络进行优化压缩,从而降低时间和空间复杂度,极大地提高搜索速度。因此,在 WFST 框架下建立 Lattice 索引的方法逐渐成为语音检索中的研究热点。文献[6-7]提出将 WFST 应用于语音识别领域,其提出的一系列理论算法为后续研究奠定了基础。在 WFST 框架下构建的语句检索(Spoken utterance retrieval,SUR)系统,充分利用了 WFST 在时间和空间上的优势,明显提高了检索效率[8]。为了满足 STD 的精确查询要求,建立 WFST 索引时可以将时间信息插入输出符号或权重中[9]。近来研究表明,在 WFST 索引结构中融入 N 元文法,能有效利用词间丰富信息,获得更高的集外词检测率[10]。

上述方法都是在 WFST 框架下直接以 Lattice 作为 ASR 输出结果建立索引。然而,Lattice 中存在的大量冗余信息会造成存储空间的浪费和搜索效率的下降。本文考虑去除 Lattice 中的冗余信息以建立更高效的索引。作为一种比 Lattice 更为紧凑的多候选结果表示形式,混淆网络比 Lattice 占用存储空间小,结构简单而且容易处理。本文提出一种基于混淆网络的 WFST 索引技术,以混淆网络代替 Lattice 作为 STD 的输入建立索引,从而构建高效的 STD 系统。索引建立阶段,首先利用自动语音识别的解码结果 Lattice 生成混淆网络并转化为自动机形式,然后对自动机进行预处理,并构建基于时间的因子转换器,最后对所有因子转换器进行联合并优化得到 WFST 索引。查询阶段,将查询项转化为自动机形式,与索引进行合成运算从而得到表示查询结果的自动机,最后根据设定的门限进行判决并输出结果。

1 基于 WFST 的索引

1.1 混淆网络的生成与转化

对于每个句子 $u_i, i=1,\cdots,n$,通过 ASR 解码输出 Lattice。然后由 Lattice 生成混淆网络,并将其转化为加权自动机 A_i ,同时提取状态的时间列表 t_i(各节点到初始节点的时间)。

定义 1 五元组 $Cn=(Q',V',q'_1,q'_N,E')$ 为包含 N 个节点的混淆网络,其中:$Q'=\{q'_1,\cdots,q'_N\}$ 为所有节点的集合;E' 表示所有转移弧的集合;V' 表示混淆网络中所有弧的标号;q'_1 和 q'_N 分别是混淆网络的始节点和末节点。

$e'=(q_{e'}^{s},q_{e'}^{f},w_{e'},g_{e'})\in E'$ 表示混淆网络中的一条弧，其中 $q_{e'}^{s}$ 和 $q_{e'}^{f}$ 分别表示弧的头节点和尾节点，若 $q_{e'}^{s}=q'_i$，则 $q_{e'}^{f}=q'_{i+1}$；$w_{e'}$ 表示弧上的标号；$g_{e'}\in(0,1)$ 表示弧 e 的后验概率。

定义 2 第 i 个混淆集合为 $S_i^m=\{e'\mid e'\in E',q_{e'}^{s}=q'_i\}\in E'$，且 $\sum_{e'\in S_i^m}g_{e'}=1$。具有 N 个节点的混淆网络，有 $N-1$ 个混淆集。

本文采用文献[11]提出的多路径同时对齐方法把 Lattice 转化为混淆网络，近似实现最小词错误率解码。在混淆网络中，竞争同一个声学位置的所有词假设形成一个混淆集。把所有混淆集按照时间顺序连接起来就构成了混淆网络。这样，最小化词错误率解码就简化为从每个混淆集合中选择后验概率最大的词，并将它们依次串联。

由 Lattice 生成混淆网络后，转化成自动机形式作为 STD 系统的输入。本文中，加权有限状态转换器和加权有限状态接收机都称为自动机，不将两者严格区分。

WFST 是定义在半环上的八元组 $T=(\Sigma,\Delta,Q,I,F,E,\lambda,\rho)$，其中：$\Sigma$ 为有限输入字母表；Δ 表示有限输出字母表；Q 表示有限状态集合；$I\subseteq Q$ 表示初始状态集合；$F\subseteq Q$ 表示终止状态集合；$E\subseteq Q\times(\Sigma\cup\{\varepsilon\})\times(\Delta\cup\{\varepsilon\})\times K\times Q$ 为有限弧集合，弧 $e=\{p[e],i[e],o[e],w[e],n[e]\}\in E$，$i[e]$ 为输入符号，$o[e]$ 为输出符号，$w[e]$ 为权重，$p[e]$ 为初节点，$n[e]$ 为末节点；λ 和 ρ 分别为初始和终止权重函数。加权有限状态接收机定义类似：$A=(\Sigma,Q,I,F,E,\lambda,\rho)$，仅省略了输出符号。

混淆网络转换成自动机，即把混淆网络的混淆节点作为自动机状态节点，混淆网络的始节点和末节点分别作为自动机的初始和终止节点；把混淆集包含的弧映射成自动机的弧，其中弧上的标号，即解码识别单元对应于自动机弧上的输入(输出)符号，弧的后验概率对应于自动机弧上的权重。初始和终止函数设为单位值。具体转换步骤如下：

(1)对混淆网络进行剪枝，剪去每个混淆集中概率特别小的弧；

(2)以混淆节点集合建立自动机状态集 $Q=Q'$，即 $q_i=q'_i$，其中 $q_i\in Q,i=1,\cdots,N$；

(3)设置 $I=q_1\in Q$ 为自动机唯一的初始状态，$F=q_N\in Q$ 为唯一的终止状态；

(4)为初始状态建立一条新的弧($I,sil,1,q_2$)；

(5)为自动机中相邻的两个状态建立新弧，根据相邻状态 $q_i,q_{i+1}\in Q,i=2,\cdots,N-1$ 对应混淆节点间的混淆集 $S_i^m=\{e'\mid e'\in E',q_{e'}^{s}=q'_i\}\in E'$ 中的每一条弧 e'，建立新的自动机弧 $e=(q_i,w_{e'},g_{e'},q_{i+1})\in E$，其中 $w_{e'}$ 表示弧上的 ASR 识别结果，$g_{e'}$ 表示弧的后验概率；

(6)为终止状态建立一条新的弧($q_{N-1},sil,1,F$)。

图 1 给出了由 Lattice 和混淆网络转换成的自动机。由于混淆网络中存在大量空弧，对其进行处理后能方便地建立索引。对于存在空弧的混淆集，如果空弧概率最大，则去除整个混淆集，只保留空弧(为了方便提取状态的时间信息)。否则保留后验概率比空弧大的弧，去除空弧和后验概率小的弧。提取状态的时间信息时，去除开始和结束的两个静音集，将每个混淆集近似划作等时间段。

1.2 预处理

得到加权自动机 A_i 后，在 Log 半环上对 A_i 应用权重推进算法[12]，将权重转化为后验概率的负对数形式。

STD 为精确检索，需将查询项在一句话中不重叠时间域内的所有出现次数都检索出来，本文采用聚类方法解决这一问题。将带有相同输入符号并在重叠时间域内的弧聚类，对于每个输入符号，具体算法如下：(1)以结束时间为准对(开始，结束)时间对排序；(2)为所有不重叠时间对分配聚类编号；(3)根据最大重叠域将剩余弧归入已有类中。通过聚类，将自动机转化为输出符号为聚类编号的转换器，从而解决一句话中查询项在不重叠时间域内出现多次的问题。图 2 表示实数半环 R 上时间列表为 $t_1=$

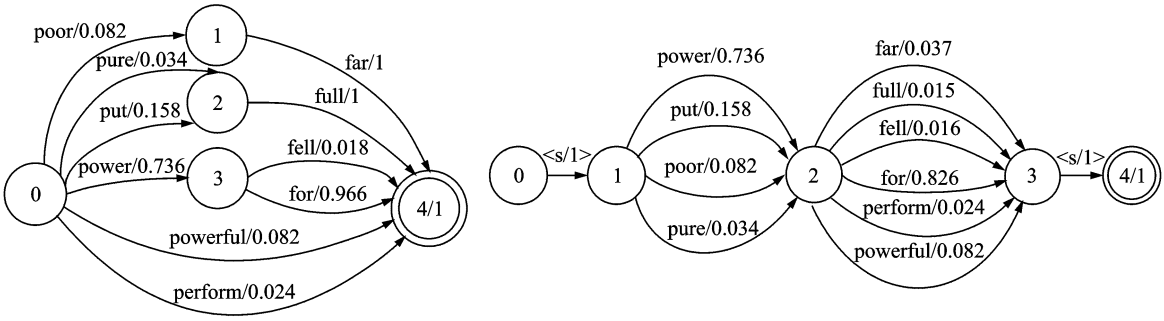


图 1 Lattice 和混淆网络转换的自动机

Fig. 1 Automata converted form lattice and confusion network

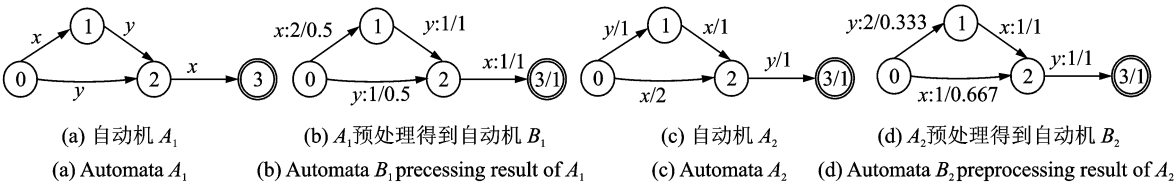


图 2 自动机预处理过程

Fig. 2 Preprocessing of automata

[0,1,2] 和 t_2=[0,1,2] 的加权自动机图 2(a,c) 和经过预处理后分别得到图 2(b,d)。

1.3 因子转换器

给定 Σ 中两个的字符串 u 和 v,如果 $u=xvy$,则称 v 是 u 的一个因子(子字符串)。字符串 u 的因子转换器 $F(u)$ 是能够识别因子 u 的最小确定有限状态转换器。因子转换器是一种字符串因子集合的倒排索引。如果因子在句子中出现,则用一条有效路径表示。路径上的输入符号表示因子,输出符号表示因子出现句子的标号,权重表示因子在句子中的得分期望。利用因子转换器可以建立一个能够直接搜索自动机接收的任何字符串中的任何因子的索引。为了满足 STD 的精确查找要求,本文构建带有时间的因子转换器[13],将时间信息插入因子转换器的权重中,从而检索出查询项的时间信息。

因子转换器的构建以半环[13]运算为基础。首先介绍半环代数的相关理论。

半环为五元组 $(K,\oplus,\otimes,\bar{0},\bar{1})$,其中 K 为数值集合,$\oplus$,$\otimes$ 为两个基本运算,$\bar{0}$ 和 $\bar{1}$ 分别为零元和单位元。定义 Log 半环 $L=(R\cup\{\infty\},\oplus_{\log},+,\infty,0)$,其中,$\forall a,b\in R\cup\{\infty\}$,$a\oplus_{\log}b=-\log(e^{-a}+e^{-b})$;Tropical 半环 $T=(R_+\cup\{\infty\},\min,+,\infty,0)$;相应地,$T'=(R_+\cup\{\infty\},\max,+,-\infty,0)$。

定义多个半环笛卡尔积上的特殊半环结构如下:对于两个半环 $A=(A,\oplus_A,\otimes_A,\bar{0}_A,\bar{1}_A)$ 和 $B=(B,\oplus_B,\otimes_B,\bar{0}_B,\bar{1}_B)$,其积半环和字典序半环分别为

$$A\times B=(A\times B,\oplus_\times,\otimes_\times,\bar{0}_A\times\bar{0}_B,\bar{1}_A\times\bar{1}_B) \tag{1}$$

$$A*B=(A\times B,\oplus_*,\otimes_*,\bar{0}_A\times\bar{0}_B,\bar{1}_A\times\bar{1}_B) \tag{2}$$

式中:$\oplus_\times$,$\otimes_\times$ 和 $\otimes_*$ 为分量对应运算,$\oplus_*$ 为字典序优先运算。例如,$\forall a_1,a_2\in A,b_1,b_2\in B$ 有

$$(a_1,b_1)\otimes_\times(a_2,b_2)=(a_1\otimes_A a_2,b_1\otimes_B b_2) \tag{3}$$

$$(a_1,b_1)\oplus_*(a_2,b_2)=\begin{cases}(a_1,b_1\oplus_B b_2) & a_1=a_2\\ (a_1,b_1) & a_1=a_1\oplus_A a_2\neq a_2\\ (a_2,b_2) & a_1\neq a_1\oplus_A a_2=a_2\end{cases}\tag{4}$$

本文基于时间的因子转换器构建在3个Tropical半环组成的字典序半环 $T*T*T$ 上。

因子转换器构建方法如下：假设 $B_i=(\Sigma_i,\Delta_i,I_i,F_i,E_i,\lambda_i,\rho_i)$ 为 A_i 经过预处理后得到的Log半环上的转换器，其权重表示每个句子 u_i 中因子的后验概率，输出符号表示聚类编号。对于每个状态 $q\in Q_i$，在Log半环上分别定义 $d[q]$ 和 $f[q]$ 为前向和后向概率

$$d[q]=\bigoplus_{\pi\in\Pi(I_i,q)}^{\log}(\lambda_i(p[\pi])+w[\pi])\tag{5}$$

$$f[q]=\bigoplus_{\pi\in\Pi(I_i,q)}^{\log}(w[\pi])+\rho_i(n[\pi]))\tag{6}$$

式中：Π 表示路径集合。以 $C_i(x,y)$ 表示因子对 $(x,y)\in\Sigma\times\Delta$ 在 u_i 中的概率，因子对 (x,y) 出现的概率期望为

$$-\log(E_{P_i}[C_i(x,y)])=\bigoplus_{i[\pi]=x,o[\pi]=y,\pi\in\Pi_i}^{\log}d[p[\pi]]+w[\pi]+f[n[\pi]]\tag{7}$$

同样地，在Tropical半环上，假设 $t_i[q]$ 表示状态 $q\in Q_i$ 的时间，$t_i^s(x,y)$ 和 $t_i^e(x,y)$ 分别表示在 u_i 中的开始时间和结束时间

$$t_i^s(x,y)=\min_{i[\pi]=x,o[\pi]=y,\pi\in\Pi_i}t_i[p[\pi]]\tag{8}$$

$$t_i^e(x,y)=\max_{i[\pi]=x,o[\pi]=y,\pi\in\Pi_i}t_i[n[\pi]]\tag{9}$$

将因子对 (x,y) 的权重映射到三元组 $\{-\log(E_{P_i}[C_i(x,y)]),t_i^s(x,y),t_i^e(x,y)\}$。首先通过因子选择建立一个转换器，对每个因子对的每次出现都进行索引，权重为相应的三元组(后验概率，开始时间，结束时间)。然后对已建立的转换器在 $L\times T\times T'$ 半环上进行优化。为了方便剪枝和最短路径计算，最后将弧的权重从 $L\times T\times T'$ 半环映射到 $T*T*T$ 半环。

根据权重在 L 半环的加权转换器 B_i 和时间列表 t_i 构建 $T*T*T$ 半环上基于时间的转换器 T_i 的具体步骤如下：

(1) 因子选择。

①将弧的权重从 L 半环映射到 $L\times T\times T'$ 半环：$w\in L\rightarrow\{w,\bar{1},\bar{1}\}\in L\times T\times T'$，即将每条弧的权重由 $\{w\}$ 转化为 $\{w,0,0\}$；

②建立状态 $s\notin Q_i$，并使 s 为唯一的初始状态；

③建立状态 $e\notin Q_i$，并使 e 为唯一的终止状态；

④为每一个状态 $q\in Q_i$ 建立一条新的弧 $(s,\varepsilon,\varepsilon,\{d[q],t_i[q],0\},q)$；

⑤为每一个状态 $q\in Q_i$ 建立一条新的弧 $(q,\varepsilon,i,\{f[q],0,t_i[q]\},e)$。

(2) 优化。

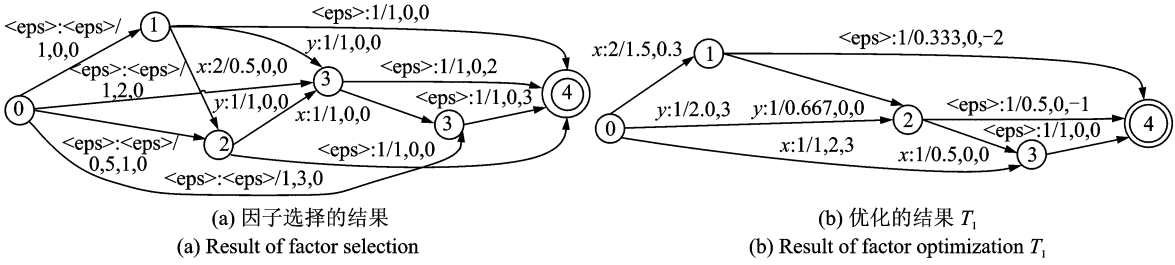
①将已建立的转换器视为接收机(把输入、输出符号共同编码成一个符号)，在 $L\times T\times T'$ 半环上运用 ε 移除、确定化、最小化等算法，得到优化后的因子转换器。在优化过程中，对带有相同因子对的路径合并：对半环 L 上的后验概率进行 $\oplus_{\log}$ 运算，对半环 T 上的开始时间进行 min 运算，对半环 T' 上的结

束时间进行 max 运算;

②每条弧的权重映射到 $T*T*T$ 半环[14]

$$\{w_1,w_2,w_3\}\in L\times T\times T'\rightarrow$$
$$\{w_1,w_2,w_3\}\in T*T*T \tag{10}$$

图 3 表示加权自动机 B_1 因子选择后的中间结果和优化后得到的转换器 T_1。



(a) 因子选择的结果
(a) Result of factor selection

(b) 优化的结果 T_1
(b) Result of factor optimization T_1

图 3 加权自动机 B_1 建立因子转换器 T_1

Fig. 3 Factor transducer T_1 from weighted automaton B_1

1.4 联合

建立每个句子的转换器 T_i 之后,整个集合基于时间的因子转换器 T 构建如下:

(1)将每个句子构建的转换器联合

$$U=\bigcup_i T_i \qquad i=1,\cdots,n \tag{11}$$

(2)通过将 U 的输入、输出符号共同编码得到接收机,对其在 $T*T*T$ 半环上运用加权 ε 移除、确定化、最小化算法;

(3)对 U 的符号解码,根据输入符号对弧排序[14];

(4)除了通向终止状态路径上弧的自动机编号,去除转换器中所有的输出符号(聚类编号),从而得到整个集合的转换器 T,即本文所要建立的索引。

由于每个 T_i 有唯一的自动机编号,在 $T_i, i=1,\cdots,n$ 之间没有相同的有效路径,因此可以对 U 在 $T*T*T$ 上进行任意优化。$T*T*T$ 半环有利于剪枝和最短路径计算。图 4 表示以图 2 中自动机 B_1, B_2 和时间列表 $t_1=[0,1,2]$ 和 $t_2=[0,1,2]$ 建立的基于时间的因子转换器,即索引 T。

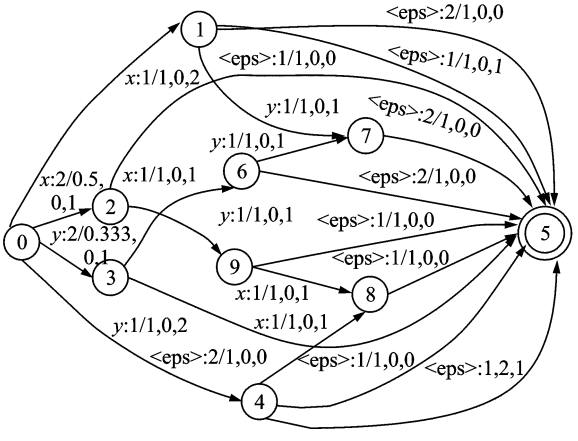


图 4 由图 2 自动机 B_1, B_2 建立的索引 T

Fig. 4 Index T constructed from the weighted automaton B_1 and B_2 in Fig. 2

2 查询项搜索

用户提出的查询项是典型的非加权字符串,也可以看成一个任意加权的自动机 X。对查询项 X 的搜索结果 R 是另一个自动机。通过以下方式得到:

(1)将 X 和 T 在输入符号处进行合成,并将得到的转换器映射输出符号[14]

$$P = \Pi_2 (X \circ T) \tag{12}$$

式中:$\circ$ 表示合成运算;Π_2 表示映射输出符号运算。

(2)进行 ε 移除等优化操作,用最短路径算法排序得到 R。

R 是一个简单的接收机,在输入状态和终止状态之间有 M 条弧。R 中的每一条有效路径的输入标签 $i[\pi]$ 为自动机编号,权重 $w[\pi] \in T * T * T$ 为三元组(后验概率的负对数值,开始时间,结束时间)。在 R 上按照弧的顺序进行简单遍历就能得到结果。

3 实验结果与分析

本文实验在 TIMIT 语料库上进行,训练语料为训练集 (不包含 SA1 和 SA2)的 3 296 个语句,测试语料为测试集(不包含 SA1 和 SA2)的 1 344 个语句。从测试语料的标注中人工选取 50 个查询项。查询项全面覆盖长词、短词、高频词和低频词等各种情况。在 ASR 实验中,声学特征选用 12 维 MFCC 和 1 维能量特征以及它们的一阶与二阶差分共 39 维,声学模型采用三音子模型,每个模型包含自左向右 5 状态 HMM,每个 HMM 状态含有 8 个混合高斯分量。语言模型采用二元文法模型。实验在 WFST 框架下分别以 Lattice 和混淆网络构建 STD 系统。

3.1 检测性能

本文采用 ATWV(Actual term weighted value)[15]作为 STD 系统检测性能的评价指标。对于 STD 系统的输入,即混淆网络和 Lattice 转化的自动机,系统运用自动机剪枝算法对其进一步剪枝。实验表明,自动机剪枝门限对系统性能存在影响。图 5 给出了不同自动机剪枝门限下两种方法的 ATWV 曲线变化情况。

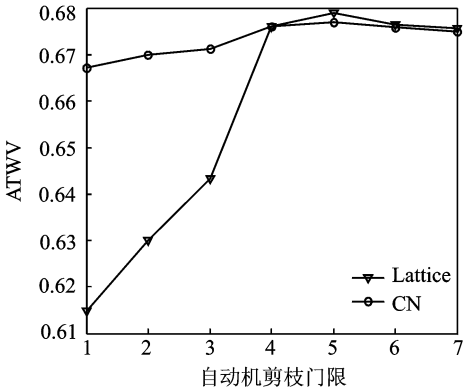


图 5 不同自动机剪枝门限下 ATWV 对比

Fig. 5 ATWV with different pruning thresholds

图 5 表明,从检测性能来看,自动机剪枝对基于混淆网络的索引影响较小,不同剪枝门限下的 AT-

WV 没有明显改变。这是由于混淆网络本身就是一种紧凑的结构,存在冗余较少。然而,自动机剪枝对于基于 Lattice 的索引的检测性能影响较大,这种影响在自动机剪枝门限值较小时比较明显。当剪枝门限值于 4 时,剪枝阈值越大,保留的节点信息就越多,因而 ATWV 随着门限值的增大而增大,而当剪枝门限值达到 4 时,ATWV 基本趋于平稳状态。这是由于剪枝阈值达到一定值时,许多冗余信息被保留下来,会增加虚警率,反而可能影响系统检测性能。

3.2 索引尺寸

在保证检测性能的情况下,即 ATWV 相近时,索引尺寸是系统评测的一项重要指标。理想情况下,索引尺寸应该越小越好,特别是随着数据量的增加不能成指数形式增长。索引尺寸很大程度上依赖于建立索引所用数据的数量和类型。由于混淆网络是 Lattice 的高效压缩形式,其建立的索引必然远远小于 Lattice 建立的索引。图 6 可以看出随着自动机剪枝门限的增加两种不同方法索引尺寸的变化情况。

从图 6 可以看出,混淆网络构建的索引尺寸远小于 Lattice 构建的索引尺寸。并且随着剪枝门限的增加,以 Lattice 建立的索引尺寸增长幅度非常大,而以混淆网络构建的索引尺寸基本不受剪枝的影响。由此可以看出本文提出的方法存在很大的优势。

3.3 搜索时间

语音检索中考虑的最重要问题之一就是搜索时间。由于 STD 是在线搜索,需要很强的实时性,搜索时间是评价系统性能的一项重要指标。本实验测试预料包含约 1.2 h 的语音测试,图 7 给出了 50 个查询项分别在两种方法中的搜索时间。

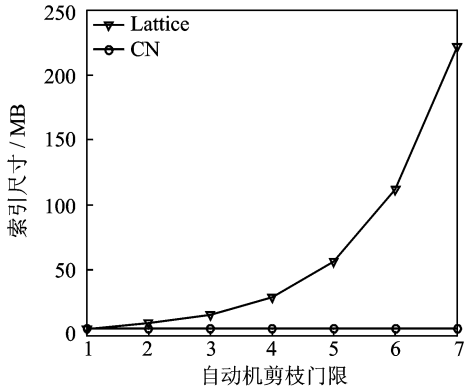


图 6 不同自动机剪枝门限下索引尺寸对比

Fig. 6 Index size with different pruning thresholds

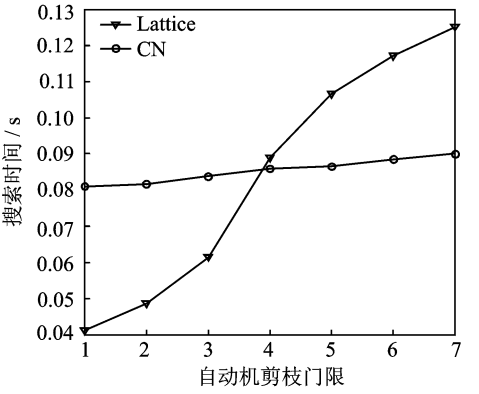


图 7 50 个查询项在不同剪枝宽度下搜索时间对比

Fig. 7 Search time of 50 queries with different pruning thresholds

从图 7 可以看出,当自动机剪枝门限不小于 4 时,以混淆网络构建的索引搜索时间更短,检索速度更快。这是由于混淆网络结构紧凑,去除了很多冗余,建立的索引较小,其搜索效率必定更高。当剪枝门限小于 4 时,以 Lattice 构建的索引搜索时间较短。但是由图 5 表明,当剪枝门限小于 4 时,由于 Lattice 剪枝幅度较大,保留下来有用信息较少,导致 ATWV 较低,其检索性能不能满足要求。因此,只要考虑 ATWV 相近的情况下两者的搜索时间。可以看出,在检测性能相近,即自动机剪枝门限不低于 4 时,以混淆网络构建的索引搜索时间更短,检索效率更高。另外,图中数据表明,两种方法搜索时间都远远低于实时,可见 WFST 的时间效率极高。

4 结束语

本文提出了一种利用混淆网络建立 WFST 索引的技术。在 WFST 框架下，以混淆网络代替 Lattice 建立索引，实现高效的 STD 系统。索引建立阶段，由 Lattice 生成混淆网络并转化为自动机形式，自动机经过预处理后，构建基于时间的因子转换器，最后进行联合优化得到 WFST 索引；查询阶段，将查询项转化为自动机形式，与索引进行合成后得到表示查询结果的自动机。实验表明，在 ATWV 相近的情况下，以混淆网络建立的索引尺寸明显小于以 Lattice 建立的索引，并且具有更快的检索速度。

参考文献：

[1] 许友亮，张连海，牛铜. 基于音位属性和边界信息的音素识别[J]. 数据采集与处理，2013，28(2)：178-183.
Xu Youliang, Zhang Lianhai, Niu Tong. Phone recognition method based on phonological attributes and phone bound-aries [J]. Journal of Data Acquisition and Processing, 2013, 28(2): 178-183.

[2] Yu P, Chen K, Ma C, et al. Vocabulary-independent indexing of spontaneous speech [J]. IEEE Transactions on Audio, Speech and Language Processing, 2005, 13(5): 635-643.

[3] Thambiratnam K, Sridharan S. Rapid yet accurate speech indexing using dynamic match lattice spotting [J]. IEEE Transactions on Audio, Speech and Language Processing, 2007, 15(1): 346-357.

[4] Mamou J, Ramabhadran B, Siohan O. Vocabulary independent spoken term detection [C]// Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. NY, USA: ACM, 2007: 615-622.

[5] Chelba C, Acero A. Position specific posterior lattices for indexing speech[C]// Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005: 443-450.

[6] Mohri M. Finite-state transducers in language and speech processing [J]. Computational Linguistics, 1997, 23(2): 269-311.

[7] Mohri M, Pereira F, Riley M. Weighted finite-state transducers in speech recognition [J]. Computer Speech & Language, 2002, 16(1): 69-88.

[8] Allauzen C, Mohri M, Saraclar M. General indexation of weighted automata: Application to spoken utterance retrieval [C]// Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004: 33-40.

[9] Arisoy E, Can D, Sak H, et al. Turkish broadcast news transcription and retrieval [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2009, 12(2): 291-301.

[10] Lee H Y, Lee L S. Improved semantic retrieval of spoken content by document/query expansion with random walk over acoustic similarity graphs[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 2014, 22(1): 80-94.

[11] Mangu L, Brill E, Stolcke A. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks[J]. Computer Speech & Language, 2000, 14(4): 373-400.

[12] Allauzen C, Riley M, Schalkwyk J, et al. OpenFst: A general and effcient weighted finite-state transducer library [J]. Lecture Notes in Computer Science CIAA, 2007, 4783: 11-23.

[13] Dogan C, Saraclar M. Lattice indexing for spoken term detection [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2011, 19(8): 2338-2347.

[14] Eilenberg S. Automata, languages, and machines [M]. [S. l.]: Access Online via Elsevier, 1974.

[15] Fiscus J G, Ajot J, Garofolo J S, et al. Results of the 2006 spoken term detection evaluation[C]// Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational. Amsterdam: [s. n.], 2007: 51-55.

作者简介：**陆梨花**(1988-)，女，硕士研究生，研究方向：语音识别与处理，E-mail：lulihua1025@163.com；**张连海**(1971-)，男，副教授，研究方向：语音识别和语音信号处理；**陈琦**(1974-)，男，讲师，研究方向：语音信号处理。