第36卷 第14期 计 算 机 工 程 2010年7月
Vol.36 No.14 Computer Engineering July 2010
·人工智能及识别技术· 文章编号：1000—3428(2010)14—0169—03 文献标识码：A 中图分类号：TP18

基于 HTK 的语音识别网络优化算法

杨善茜，黄汉明，蒋正锋，李 锐
(广西师范大学计算机科学与信息工程学院，桂林 541004)

摘 要：隐马尔可夫模型工具包(HTK)的 HParse 命令根据用户以正则表达式形式定义的任务语法来生成 HTK 可用的底层表示的语音识别网络，但不是每个语句都能用正则表达式表示出来。针对该问题，提出基于 HTK 的语音识别网络算法用于识别网络的优化问题，给出该算法的具体实现过程。实验结果表明，在保证识别率的前提下，优化后的语音识别网络在语音识别系统中所用的时间比较短，算法是有效的。

关键词：连续语音识别；自动机；隐马尔可夫模型工具包；语音识别网络

HTK-based Optimization Algorithms of Speech Recognition Network

YANG Shan-xi, HUANG Han-ming, JIANG Zheng-feng, LI Rui
(College of Computer Science and Information Technology, Guangxi Normal University, Guilin 541004)

【Abstract】 For speech recognition network of Hidden Markov Model ToolKit(HTK) bottom representation is generated by the HParse command module of the HTK according to the form of regular expressions to define the task grammar, but not every language can use regular expressions to express. Aiming at the problem, this paper presents a HTK-based speech recognition network algorithm used to identify the network optimization problem, gives the detailed realization of the algorithm. Experimental results show that the optimized speech recognition network costs less time in speech recognition than the original un-optimized one, while the recognition rate of the two recognition system configurations are almost the same, and verifies the validity of the proposed algorithm.

【Key words】 continuous speech recognition; automata; Hidden Markov Model ToolKit(HTK); speech recognition network

1 概述

在语音识别中，语音识别器可由识别网络、字典和 HMM 集构成。其中，识别网络定义了可识别的语句，这些语句也包括孤立的词，语句的集合称为语言。语言可以被多个不同的识别网络所接受，那么选择合适的识别网络对于语音识别器来说至关重要。一般的识别方法是针对每一句可识别语句建立一个 HMM[1-2]，然后使用 Viterbi[1-2]搜索算法来计算每一个 HMM 的概率值。因此，在基于隐马尔可夫模型工具包(Hidden Markov Model ToolKit, HTK)的连续语音识别系统中是将各种可能的子词序列组建成一个识别网络，利用 Viterbi 算法计算可能产生的识别语句。HTK 是一个实验平台，它的算法为了要保证通用性，网络结构设计得比较复杂。在连续语音识别中，为了能够提高 Viterbi 算法的搜索速度，进而提高识别速度，可以在 HTK 的基础上，借鉴一些好的思路，实现识别网络的优化。在优化过程中引入自动机理论，使网络优化工作取得比较好的效果，搜索算法的运算量也将大大减少，有效提高识别速度。本文算法的基本思路是：在 HTK 基础上把识别网络看成是一个自动机，以解决优化语音识别网络的问题。

本文论述了识别网络的优化算法，再搭建汉语连续数字串识别系统，分别测试优化网络与原语音识别网络用于识别 100 次所用的时间。

2 识别网络的生成

优化原始的语音识别网络，由原始的语句或语言得到转换系统，再用子集法将不确定的有穷自动机(Non-deterministic Finite Automata, NFA)确定化为确定的有穷自动机(Deterministic Finite Automata, DFA)[3-4]，然后把 DFA 极小化，再将其转换为与 DFA 对应的语音识别网络。

2.1 语料库的建立

在实验室环境下采用 Cool Edit Pro2.0 作为录制工具完成录音工作，设置采样率为 16 000 Hz，声道为单声道，采样精度为 16 bit。将 123456, 1235, 17856 每个连续数字串录制 15 个语音文件用于训练(也用于测试)，语音数据库由 45 个语音文件组成。

2.2 原始语音识别网络的生成

根据待识别的连续数字串，可以建立一个线性文法规则，执行 HParse[1]指令则生成线性识别网络为 linearnet.txt，其中，N=20, L=21，原始语音识别网络如图 1 所示。

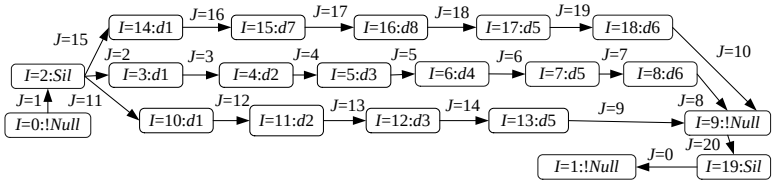

图 1 原始语音识别网络

作者简介：杨善茜(1984－)，女，硕士，主研方向：语音识别网络；黄汉明，副教授、博士；蒋正锋、李 锐，硕士

收稿日期：2010-01-10 **E-mail：**hhmmentor@gmail.com

其中，$N=20$ 代表有 20 个节点(Nodes)；$L=21$ 代表有 21 条连接(Links)；I 为节点的编号；J 为连接的编号；$I=4:d2$ 说明第 4 个节点的内容是 $d2$，$d2$ 代表语音为数字 2，其余类推。

3 优化网络的生成

在优化过程中必须遵循“简化后的网络，其所有可能的路径和原来的网络结构相同[2]”，即使用 HParse 指令生成的识别网络和优化网络的所有路径组成的集合是完全一样的。

3.1 原始语句转换系统

待识别的语句是 123456, 1235 和 17856，这 3 个语句就是自动机能够接受的语言。构造转换系统，再用子集法将其确定化为 DFA，根据 NFA 确定化为 DFA 的过程如表 1 所示，相应的状态如图 2 所示。其中，S 是开始状态；E 是终止状态，在这里除开始状态外非终结状态都用小写字母表示。

表 1 NFA 确定化为 DFA 的过程

起始状态	输入	达到状态
$ST[S]$	1	$A[a,f,j]$
$A[a,f,j]$	2	$B[b,h]$
$A[a,f,j]$	7	$C[k]$
$B[b,h]$	3	$D[c,i]$
$C[k]$	8	$E[l]$
$D[c,i]$	4	$F[d]$
$D[c,i]$	5	$END[E]$
$E[l]$	5	$G[m]$
$F[cd]$	5	$H[e]$
$H[e]$	6	$END[E]$
$G[m]$	6	$END[E]$

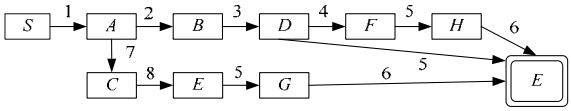

图 2 DFA 状态

3.2 DFA 极小化

表 1 根据极小化算法[3-4]，最后得出的结论为状态 F 等价状态 E，状态 H 等价状态 G。按这个极小化算法，得出等价状态后，构造出商自动机对应的状态图。

3.3 转换与 DFA 对应的语音识别网络

状态最少的商自动机，即是最优的。一个节点上的内容就是待识别的语音单元，那么该商自动机所对应的语音识别网络如图 3 所示，刚好能够识别 123456, 1235 和 17856。

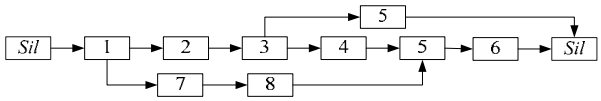

图 3 商自动机所对应的语音识别网络

根据图 3，建立文法规则：

```
$digitfirst = d2 d3 d4;
$digitsec = d7 d8;
$digitthird = $digitfirst|$digitsec;
$digittemp =  d1 $digitthird d5 d6;
 (SILENCE  digittemp SILENCE)
```

在 HTK 中，识别网络的生成过程是先写出正规表达式，然后用 HParse 指令得到 HTK 能够识别的文件格式。这就要求优化后识别网络与所生成的 HTK 能够识别的网络一致。简化后的识别网络如图 4 所示。

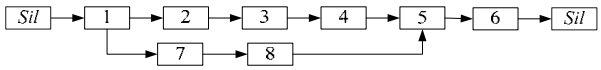

图 4 简化后的语音识别网络

生成的识别网络为 firststepnet.txt，其中，N=13, L=13，网络相应的节点和链路连接情况见表 3 和表 4，其对应的识别网络是一个不完全的语音识别网络，只能识别 123456 和 17856，所以，要增加节点 $d5$，需要修改节点编号和链路编号。当一个节点 A 的入度大于等于 2 时，就要在其前增加一个 $W=!Null$ 节点，连接到节点 A 的链路就接连到 $W=!Null$ 上，再从 $W=!Null$ 节点连接到 A 节点。

表 2 简化后的识别网络节点编号及节点

节点编号	节点内容	节点编号	节点内容
I=0	W=!*Null*	I=8	W=$d7$
I=1	W=! *Null*	I=9	W=$d8$
I=2	W=*SILENCE*	I=10	W=$d5$
I=3	W=$d1$	I=11	W=$d6$
I=4	W=$d2$	I=12	W=$d5$
I=5	W=$d3$	I=13	W=! *Null*
I=6	W=$d4$	I=14	W=*SILENCE*
I=7	W=! *Null*		

表 3 简化后的识别网络链路情况

链路编号	链路内容		链路编号	链路内容	
	头节点	尾节点		头节点	尾节点
J=0	S=14	E=1	J=8	S=3	E=8
J=1	S=0	E=2	J=9	S=8	E=9
J=2	S=2	E=3	J=10	S=7	E=10
J=3	S=3	E=4	J=11	S=10	E=11
J=4	S=4	E=5	J=12	S=11	E=13
J=5	S=5	E=6	J=13	S=5	E=12
J=6	S=6	E=7	J=14	S=12	E=13
J=7	S=9	E=7	J=15	S=13	E=14

3.4 节点插入算法

节点插入算法是识别网络优化过程中非常重要的一个模块，其主要目的是增加节点，并修改节点编号和链路编号，使优化后的语音识别网络和采用 HParse 指令生成的识别网络的所有路径组成的集合完全一样。节点插入算法可以采用以下的伪码语言来描述：

(1)节点列表的修改

在节点列表中找出最大的节点编号的节点 N，并用 $IMaxNum$ 记录节点 N，把节点 N 的节点编号改成一个较大的值 M，M 的值是原来节点个数与新插入实节点个数之和的 2 倍，即 $M=2\times(InsertNum+N+1)$。

(2)链路列表的修改

把链路列表中所有 $S=IMaxNum$ 和 $E=IMaxNum$ 分别改成 $S=M$ 和 $E=M$，用 $JMaxNum$ 记录链路列表中所有的链路。

(3)增加新节点到节点列表

给要插入的节点分配一个编号 $I=IMaxNum$，并把它按编号从小到大的顺序插入到节点列表中，修改记录原来最大节点编号的变量值 $IMaxNum=IMaxNum+1$。

(4)增加新链路到链路列表

按识别网络示意图增加以新节点为尾的链路 $J=JMaxNum$ 和以新节点为头节点的链路 $J=JMaxNum+1$，如果链路 $J=JMaxNum$ 的头节点和 $J=JMaxNum+1$ 的尾节点就是链路列表中的一条链路 $J=JTemp$ 的头节点和尾节点，那么把链路 $J=JMaxNum$ 的尾节点作为 $J=JTemp$ 的尾节点，删除链路 $J=JMaxNum$，并且把链路 $J=JMaxNum+1$ 的链路编号改为 $J=JMaxNum$。如果不存在以链路 $J=JMaxNum$ 的头节点和 $J=JMaxNum+1$ 的尾节点构成的链路，则 $JMaxNum+=2$。

(5)如果还存在没有插入的节点，则转到第(3)步，否则在识别网络示意图中还没有添加到链路列表中的链路逐一添加进入，每次 $J=JMaxNum$，$JMaxNum+=1$。

(6)入度大于等于 2 的节点的处理

检查链路列表中每一个节点是否有入度大于等于 2 的，即 2 条或 2 条以上链路的尾节点是否相同，如果存在相同的尾节点 $I = ISame$ 并且不是 $Null$ 节点的链路 $J = JSame$，则在节点表格中新增加一个 $Null$ 节点，节点编号为 $I = IMaxNum$，插入一条链路 $J = JMaxNum$，其中，$S = IMaxNum$；$E = ISame$，把链路编号表格中所有 $E = ISame$ 分别改成 $E = IMaxNum$，$IMaxNum = IMaxNum + 1$。

(7)在节点列表中将节点编号 $I = M$ 修改成 $I = IMaxNum$。把链路列表中所有 $S = M$ 和 $E = M$ 分别改成 $S = IMaxNum$ 和 $E = IMaxNum$。

在 firststepnet.txt 中，按照节点插入算法，插入节点 $W = d5$，修改后的识别网络为 finalnet.txt，语音识别网络如图 5 所示。

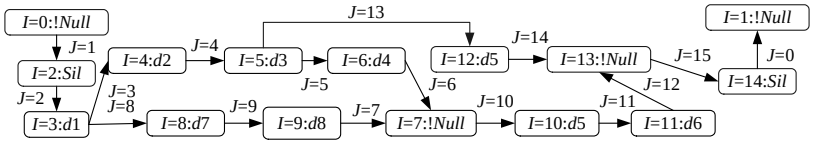

图 5　插入新节点后的语音识别网络

4　实验

实验采用的训练集和测试集都是前面所建立的语料库。特征参数为 13 维的 MFCC(MFCC_E_A_D)，包括 12 阶的频谱值加上对数能量，并取其一阶差分和二阶差分，共 39 维。选择音节作为识别单元，状态数取 6，并采用 1 个 Stream，搭建汉语连续数字串识别系统。使用原始语音识别网络和优化后的语音识别网络分别测试该识别系统 100 次，每次识别 45 句连续数字，其在测试中所花的时间如表 4 所示。

表 4　时间复杂度比较

语音识别网络类型	识别 100 次所花的时间/s	识别率/(%)
原始语音识别网络	92.609	97
优化后的语音识别网络	83.031	97

一般地，FA 是从识别语言句子的角度定义语言，原识别网络对应 NFA，优化后的识别网络是 DFA。在每一步，NFA 可能处于若干状态，而 DFA 只能处于一个状态。由于对于任意给定的 NFA，存在一个 DFA 与之等价[3]，将 NFA 化为等价的 DFA，也就是将 NFA 的“树状”计算变成 DFA 的“线状”计算[5]。因此，避免了搜索时因不可达所导致的“回溯(回过头来，重新进行选择)”所用的时间，从而提高了系统的效率。

在连续语音识别中，使用优化后的语音识别网络来减少搜索时间是可行的。将待识别的语句与自动机联系起来，应用自动机理论的知识来指导原始语音识别网络的优化，然后用实验证明了它的正确性。由于只是为了测试识别网络对语音识别系统的影响，因此测试所用到的语音与训练所用到的语音数据是一样。通过表 4 可以看出，在保证识别效果的前提下，优化后的语音识别网络识别 100 次所花的时间为 83.031 s，比原始语音识别网络少花了 9.578 s。

5　结束语

大词汇量连续语音识别系统是语音识别技术应用的一个发展方向，主要应用于基于 PC 处理平台的听写机以及与电话网或者互联网相结合的语音信息查询服务系统。这些系统都需要较大规模的运算平台实现，如果在该系统中应用优化后的语音识别网络，将会减少系统运行时 Viterbi 搜索的搜索空间和计算量，从而大大提高系统的执行速度，对于那些对系统的实时性有要求的系统，无疑也是有益的。

参考文献

[1] Young S, Evermann G, Gales M. The Hidden Markov Model Toolkit[EB/OL]. (2005-10-20). http://htk.eng.cam.ac.uk/.

[2] Jang R. Audio Signal Processing and Recognition[Z]. (2009-05-30). http://neural.cs.nthu.edu.tw/jang/books/audioSignalProcessing/.

[3] 蒋宗礼, 姜守旭. 形式语言与自动机理论[M]. 2 版. 北京: 清华大学出版社, 2007.

[4] 陈文宇, 欧　齐, 程　炼. 形式语言与自动机[M]. 北京: 人民邮电出版社, 2005.

[5] 张立昂. 可计算性与计算复杂性导引[M]. 2 版. 北京: 北京大学出版社, 2004.

编辑　索书志

(上接第 168 页)

[5] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[6] 马　良. 基于蚂蚁算法的函数优化[J]. 控制与决策, 2002, 17(Zl): 719-726.

[7] Dréo J, Siarry P. An Ant Colony Algorithm Aimed at Dynamic Continuous Optimization[J]. Applied Mathematics and Computation, 2006, 181(1): 457-467.

[8] Krzysztof S, Dorigo M. Ant Colony Optimization for Continuous Domains[J]. European Journal of Operational Research, 2006, 185(3): 1155-1173.

[9] Baskan O, Haldenbilen S, Ceylan H. A New Solution Algorithm for Improving Performance of Ant Colony Optimization[J]. Applied Mathematics and Computation, 2009, 211(1): 75-84.

[10] 谢　政. 对策论[M]. 长沙: 国防科技大学出版社, 2004.

[11] Hamzacebi C. Improving Genetic Algorithms' Performance by Local Search for Continuous Function Optimization[J]. Applied Mathematics and Computation, 2008, 196(1): 309-317.

[12] Dejong K A. Analysis of the Behavior of a Class of Genetic Adaptive Systems[D]. Ann Arbor, USA: University of Michigan, 1975.

[13] Weibull J W. Evolutionary Game Theory[M]. Cambridge, MA, USA: MIT Press, 1995.

[14] Gibbons R. A Primer in Game Theory[M]. New York, USA: Harvester Wheatsheaf, 1992.

编辑　任吉慧