标签同步解码算法及其在语音识别中的应用

陈哲怀[1),2)] 郑文露[3)] 游永彬[4)] 钱彦旻[1),2)] 俞 凯[1),2)]

[1)](上海交通大学智能交互与认知工程上海高校重点实验室 上海 200240)

[2)](上海交通大学计算机科学与工程系智能语音实验室 上海 200240)

[3)](上海交通大学苏州人工智能研究院 江苏 苏州 215000)

[4)](苏州思必驰信息科技有限公司 江苏 苏州 215000)

摘 要 自动语音识别(Automatic Speech Recognition,ASR)等序列标注任务的一个显著特点是其对相邻帧的时序序列关联性建模.用于对相邻帧进行时序建模的主流序列模型包括隐马尔可夫模型(Hidden Markov Model,HMM)和连接时序模型(Connectionist Temporal Classification,CTC).针对这些模型,当前主流的推理方法是帧层面的维特比束搜索算法,该算法复杂度很高,限制了语音识别的广泛应用.深度学习的发展使得更强的上下文和历史建模成为可能.通过引入 blank 单元,端到端建模系统能够直接预测标签在给定特征下的后验概率.该文系统地提出了一系列方法,通过使用高效的 blank 结构和后处理方法,使得搜索解码过程从逐帧同步变为标签同步.该系列通用方法在隐马尔可夫模型和连接时序模型上均得到了验证.结果表明,在 Switchboard 数据集上,不损失性能的前提下,实验取得了 2~4 倍的加速.该文同时研究了搜索空间、候选序列剪枝、转移模型、降帧率等对加速比的影响,并在所有情况下取得一致性加速.

关键词 自动语音识别;隐马尔可夫模型;连接时序模型;逐帧同步解码;标签同步解码;可变帧率;剪枝

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2019.01511

Label Synchronous Decoding for Speech Recognition

CHEN Zhe-Huai[1),2)] ZHENG Wen-Lu[3)] YOU Yong-Bin[4)] QIAN Yan-Min[1),2)] YU Kai[1),2)]

[1)](*Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai* 200240)

[2)](*SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

[3)](*Suzhou Institute of Artificial Intelligence, Shanghai Jiao Tong University, Suzhou, Jiangsu* 215000)

[4)](*AISpeech Ltd., Suzhou, Jiangsu* 215000)

Abstract A unique phenomenon in human speech is the variable lengths in acoustic waves and linguistic words. Hence automatic speech recognition (ASR) requires both pattern classification and state alignment modeling between input and output sequences, called sequence prediction problem. In the inference stage, a speech recognizer is to find a sequence of labels whose corresponding acoustic and language models best match the input feature, called decoding, which determines the recognition speed and precision in real application. The most recent milestone of ASR is the application of deep neural networks (DNN) in acoustic and language modeling. However, those successful applications are still based on the traditional formulation of speech recognition and the inference stage is unchanged. In this paper, we aim to improve the decoding algorithm in the inference stage. The dominant decoding method nowadays is frame synchronous

收稿日期:2018-09-16;在线出版日期:2019-03-21.本课题得到国家重点研发计划"智能机器人"重点专项(2017YFB1302400)、国家自然科学基金项目(U1736202)、江苏省基础研究计划(BE2016078)资助.**陈哲怀**,博士研究生,主要研究方向为语音识别、语音合成和深度学习等.E-mail:chenzhehuai@sjtu.edu.cn.**郑文露**,博士,研究助理,主要研究方向为语音识别.**游永彬**,硕士,研究助理,主要研究方向为语音识别.**钱彦旻**(通信作者),博士,副教授,中国计算机学会(CCF)会员,主要研究方向为语音识别、语音理解及机器学习等.E-mail:yanminqian@sjtu.edu.cn.**俞 凯**(通信作者),博士,教授,中国计算机学会(CCF)会员,主要研究领域为认知型对话系统、语音合成、识别、理解及机器学习等.E-mail:kai.yu@sjtu.edu.cn.

Viterbi beam search whose algorithm complexity is linear with the length of the acoustic waves. Despite the wide adoption, the approach has several weakness. (1) It is an equal interval search algorithm and inefficient to deal with the variable length in the feature sequence. (2) As the sequence is decomposed to frame level as the feature sequence, the model granularity is small and the search space is large, e. g. , Hidden Markov Model states of different histories. (3) Greedy beam pruning is conducted at each frame, which is usually hard to balance search efficiency and search errors. In this paper, based on deep learning based confusion blank symbol modeling, we systematically propose label synchronous decoding (LSD) to transform the search process from frame level to label level and obtain significant speedups. We propose to transform the search process above from frame level to label level whose complexity is linear with the length of linguistic words. Namely, we utilize effective blank structure and apply efficient post-processing of blank during inference before doing Viterbi search. The post-processing is applied on the frame level acoustic model outputs: (1) Decide whether there is a label output at the current frame. (2) If so, conduct the search process. If not, discard the label output. Thus the post-processing can be viewed as the approximation of the probability calculation of each output label. The advantage of this method is the smaller search space versus the traditional methods. The search process can be greatly speed up. Moreover, the proposed framework can be applied to both generative and discriminative sequence models. In contrast to phone synchronous decoding we previously proposed, the major contributions are: (1) Propose a general decoding framework and corresponding algorithms for sequence labeling using various sequence models. (2) Role of blank is investigated and the general principle to make use of it in acceleration is discussed. The proposed framework can be uniformly applied in both Hidden Markov Model (HMM) and Connectionist Temporal Classification (CTC) based acoustic models. Experiments in the switchboard corpus show 2—4 times speed-ups for all above models without performance deterioration. Systematic investigations of the search space, hypothesis pruning, transition model and frame rate reduction in the proposed framework are conducted.

Keywords automatic speech recognition; hidden Markov model; connectionist temporal classification; frame synchronous decoding; label synchronous decoding; variable frame rate; hypothesis pruning

1 引 言

序列标注问题是指一类将给定的数据序列转化为标签序列的任务[1],如自动语音识别(Automatic Speech Recognition,ASR)和手写体识别等.区别于传统模式识别问题的是,序列标注任务中,给定样本的各数据点不符合独立同分布(independent and identically distributed,i.i.d)假设.该类问题的一个显著特点在于,特征向量序列具有可变长性,如 ASR 中,由说话人语速变化所导致的语音信号时长的不同.

为了对上述时序特征进行建模,人们提出了序列模型.根据其建模过程,序列模型可以分为以下两类:(1)生成式序列模型(Generative Sequence Models,GSM),如隐马尔可夫模型(Hidden Markov Model,HMM);(2)判别式序列模型(Discriminative Sequence Models,DSM),如连接时序模型(Connectionist Temporal Classification,CTC)等.对于 GSM,在序列鉴别性训练时,需要在序列层面使用贝叶斯定理,从条件似然度推导出序列后验概率;而 DSM 则可以直接推导和优化序列后验概率.

通常来说,出于以下原因,GSM 和 DSM 被分解为帧层面的训练准则:(1)为了更加高效地发挥帧层面分类器的建模效果,如混合高斯模型(Gaussian Mixture Model,GMM)[2] 和深度神经网络(Deep Neural Network,DNN)[3];(2)为了减轻模型的稀疏性,以及通过将简单模型分解为多个组分来增强模型的泛化能力,例如 ASR 中将模型分解为声学模型、字典和语言模型等;(3)未经序列分解的模型需

要在推理前得到整个序列信息再进行后续处理，这将给解码过程造成严重的运行延时. 本文提出的序列标注方法即是基于这样的模型[4-6]①.

在推理阶段，为了找到与输入特征最为匹配的标签序列，搜索过程需要将声学模型，语言模型和字典等结合起来. 这一过程是通过在每帧使用基于束剪枝的维特比算法来实现的[7]，称为帧同步解码(Frame Synchronous Decoding，FSD). 在该框架中，我们将特征帧的数量和语句长度的比值定义为特征速率，将标签输出数量与语句长度的比值定义为标注速率，将解码的帧数与语句长度的比值定义为解码速率. 那么，在帧同步解码中，上述三个速率均相等.

帧同步解码虽然已被广泛使用，但仍存在一些缺点：(1) 这是一个等间隔搜索算法，在处理可变长序列时较为低效；(2) 由于序列被分解为帧来作为特征序列，模型的粒度变小，导致搜索空间很大. 如ASR 中，词语历史、音素序列以及 HMM 状态之间的关联性通常以加权有限状态机(Weighted Finite-State Transducer，WFST) 进行表示(通常称为 HCLG[8] 搜索空间). 由于由多个庞大知识源共同组成，因此组成该搜索空间的状态机最终将达到百亿条边；(3) 在每帧进行贪心束剪枝通常很难兼顾搜索效率和搜索误差.

近来，神经网络的发展使得更强的上下文和历史建模效果成为可能[9-10]. 同时，更多的标注数据也进一步缓解了模型的稀疏性和泛化问题. 这些进展使得研究人员们有可能在更大的模型粒度上从帧到整个序列层面上[5,11-14]进行序列分解，如 Soltau 等人报道的一个基于单词粒度深度学习的声学模型[12]，在 125 K 小时标注数据上的表现优于较小粒度的模型. 在这些研究中，标注速率小于特征速率，但解码速率仍然等于特征速率.

本文提出将特征层面的搜索过程改变为标签层面，即搜索空间是由不同历史的标签组成的，使得解码速率等于标注速率，从而小于特征速率. 具体来说，在标签推理阶段，对帧层面声学模型的输出增加一步后处理过程：(1) 判断当前帧是否存在标签输出；(2) 若有，执行搜索过程；若无，则丢弃标签输出. 因此该后处理过程可被看作是每个输出标签概率计算的近似. 与传统方法相比，该方法的优势是搜索空间更小，且搜索过程被大大加速.

在之前的工作中，本文作者曾提出了音素同步解码[15]，与之相比，本文的主要贡献和创新点是：(1) 提出了一个可被用于不同序列模型中序列标注任务的通用解码框架和相应算法；(2) 研究并讨论了 blank 单元的作用以及在该加速框架中 blank 的设计原则；(3) 同时研究了搜索空间、候选序列剪枝、转移模型、降帧率等对加速比的影响，并在所有情况下取得一致性加速.

本文第 2 节将首先对语音识别解码算法的研究现状进行简要介绍和分析，其中 2.1 节，作者将对序列标注问题进行简要综述，并对比两种序列模型——GSM 和 DSM；2.2 节中将介绍传统逐帧同步解码的推理框架. 接着，在第 3 节和第 4 节，本文将提出标签同步解码算法并对其应用进行介绍；第 5 节将给出实验和分析结果的描述；最后第 6 节为本文结论.

2　语音识别解码算法研究现状分析

2.1　序列标注与序列模型

2.1.1　序列标注

序列标注包括所有将数据特征序列转化为标签序列的任务[1]，本节以 ASR 为例进行简要介绍. 在训练阶段，一组带有已知标签的输入特征被提供给系统进行模型构建；而测试阶段则基于特征序列和其他知识源，如语言模型和字典，进行模型推理.

序列标注问题与传统模式识别的区别在于以下两个方面：

(1) 序列内数据的相关性. 无论是特征序列，还是标签序列，序列中各数据点均不符合独立同分布(i.i.d.) 假设. ASR 中，特征序列是由声道的连续运动而产生的. 而标签序列则受到句法和语法规则、字典以及语言模型的约束. 因此，特征和标签均为强相关序列.

(2) 标签与特征序列之间的相关性. ASR 中，特征和标签之间的对齐方式是未知的，标签序列总是短于特征序列，即其主要问题在于由语速变化等导致的特征序列的可变长性. 这就要求序列模型能够同时确定输出标签的位置和内容.

2.1.2　序列模型：GSM 与 DSM

为了对上述序列相关性这一特征进行建模，人们提出了序列模型. 根据其建模过程，序列模型可被分为生成式序列模型(GSM) 和判别式序列模型(DSM).

① 最近提出的编码器-解码器模型(encoder-decoder)[4-5]则是直接在序列层面进行处理，而不进行序列分解，因此不在本文讨论之列. 本文的一些初步扩展工作可参见文献[6].

生成式序列模型是通过计算给定标签序列时特征序列的概率 $p(\boldsymbol{x}|\boldsymbol{l})$来定义的.该模型通过贝叶斯方法引入人类发声物理过程的先验知识,来提供时序和长度约束.HMM 因其作为生成式序列模型来表征人类语音声学特征的能力,而成为 ASR 的流行建模方法.在神经网络-隐马尔可夫模型(Neural Network-Hidden Markov Model,NN-HMM)混合系统中,HMMs 用来对语音信号的动态变化进行建模,而观测概率则通过神经网络来进行估计.

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{l}) &= \sum_{\pi\in\mathcal{A}(\boldsymbol{l})} p(\boldsymbol{x}|\pi) \\ &= \sum_{\pi}\prod_{t=1}^{T} p(\boldsymbol{x}|\pi_t)P(\pi_t|\pi_{t-1}) \\ &= \sum_{\pi}\prod_{t=1}^{T} p(\pi_t|\boldsymbol{x})\frac{P(\pi_t|\pi_{t-1})}{P(\pi_t)}p(\boldsymbol{x}) \\ &\simeq \sum_{\pi}\prod_{t=1}^{T} p(\pi_t|\boldsymbol{x})\frac{P(\pi_t|\pi_{t-1})}{P(\pi_t)} \end{aligned} \tag{1}$$

其中 $\boldsymbol{l}$ 是生成式序列模型的标签序列,如上下文相关(Context Dependent,CD)的音素序列.π 是 HMM 状态序列,π_t 是第 t 帧对应的 HMM 状态,$\pi_t^{(l)}$ 是指第 l 个 HMM 模型的第 s 个 HMM 状态.$P(\pi_t|\pi_{t-1})$ 是 HMM 状态转移概率,$P(\pi_t)$ 是 π_t 的状态先验概率.$\mathcal{A}$ 是指从标签序列 $\boldsymbol{l}$ 到其相应 HMM 状态序列 π 的映射函数,如下所示.

$$\mathcal{A}: L \mapsto \{\pi_1^{(1)},\cdots,\pi_5^{(1)},\cdots,\pi_5^{(|L|)}\} \tag{2}$$

L 是标签序列 $\boldsymbol{l}$ 的各个单元的集合.其中每个标签序列单元对应一个 HMM 模型,而每个 HMM 模型对应五个 HMM 状态,如图 1(a)中所示.状态后验概率 $p(\pi_t|\boldsymbol{x})$可通过神经网络进行估计得出.

而判别式序列模型则是直接计算给定特征序列 $\boldsymbol{x}$ 时输出标签序列 $\boldsymbol{l}$ 的后验概率 $p(\boldsymbol{l}|\boldsymbol{x})$.其中,连接时序模型(CTC)用于解决未分割序列数据的标注问题,它通过引入 blank 标签单元,实现对输入序列任意一点的一对一输出标签预测.

$$p(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\pi\in\mathcal{B}(\boldsymbol{l})} p(\pi|\boldsymbol{x}) = \sum_{\pi}\prod_{t=1}^{T} p(\pi_t|\boldsymbol{x}) \tag{3}$$

其中$\mathcal{B}$为如下所定义的一对多映射:

$$\mathcal{B}: L \mapsto L \cup \{\text{blank}\} \tag{4}$$

$\mathcal{B}$决定了标签序列 $\boldsymbol{l}$ 以及 $\boldsymbol{l}$ 对应的模型单元序列 π 的集合.如图 1(b)所示,通过在序列 $\boldsymbol{l}$ 的每个标签单元 l 之间插入一个可选的自循环 blank 单元进行映射.$p(\pi_t|\boldsymbol{x})$则可使用以特征序列 $\boldsymbol{x}$ 为输入的循环神经网络(Recurrent Neural Network,RNN)或长短时记忆神经网络(Long Short Term Memory,LSTM)[16]等估计得到.

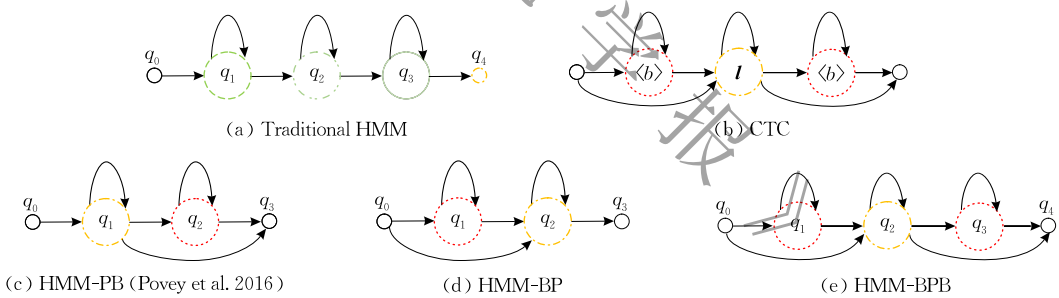

图 1 HMM、CTC 和本文提出的方法中隐藏状态的拓扑结构示意图(在后三种结构中,其名称中 B 指 blank HMM 状态,P 指标签输出 HMM 状态.每个大圆圈代表一个由神经网络建模发射概率的 HMM 状态.其中,点划线圆圈表示输出标签建模,每个均分配一个特定的模型单元,如(b)CTC 中的 $\boldsymbol{l}$.虚线圆圈表示 blank 建模,但它们并不完全相同,如(b)CTC 中的〈b〉是使用公共的 blank 建模;但(c)中的 q_2,每个输出标签有独立的 blank 建模,本文 5.2 节(3)中详细比较了不同 blank 的粒度和拓扑结构所带来的区别.其他实线小圆圈,如(c)中 q_0、q_3,(d)中 q_0、q_3,(e)中 q_0、q_4,代表非发射状态.自循环状态转移表示该状态接受当前状态的重复输出.本文 5.2 节(3)中对这些拓扑结构进行了详细比较)

通常,如本文引言中所述,为了有效利用帧级分类器如 GMM[2] 和神经网络[3] 的建模效果,减轻建模的稀疏性和增强泛化能力,避免未经分解的模型因处理整个序列而导致的运行延时等问题,GSM 和 DSM 都被分解为帧层面上的训练,本文接下来便对传统的帧同步解码进行介绍.

2.2 帧同步解码

在模型推理阶段,为了找到与输入特征最为匹配的标签序列,搜索过程需要将前述序列模型与其它知识源,即字典、语言模型等融合起来.即解码标签序列是由前述各分解序列所共同决定的.该搜索过程是通过在每帧上使用基于束剪枝的维特比算法进行的[7],即帧同步解码(FSD).FSD 框架中,解码速率等于标注速率,标注速率等于特征速率.

2.2.1 公式推导

在大词汇量连续语音识别(Large Vocabulary

Conversational Speech Recognition，LVCSR）中，解码算法的目标是找到最佳的词序列. 通过应用字典和语言模型将词序列映射到标签序列，解码公式可推导如下：

$$\begin{aligned}\boldsymbol{w}^{*}&=\arg\max_{\boldsymbol{w}}\{P(\boldsymbol{w})p(\boldsymbol{x}|\boldsymbol{w})\}\\&=\arg\max_{\boldsymbol{w}}\{P(\boldsymbol{w})p(\boldsymbol{x}|\boldsymbol{l}_{\boldsymbol{w}})\}\end{aligned}\tag{5}$$

其中，$\boldsymbol{w}$ 是词序列，$\boldsymbol{w}^{*}$ 是最佳词序列. $\boldsymbol{l}_{\boldsymbol{w}}$ 表示 $\boldsymbol{w}$ 通过映射得到的标签序列，如 NN-HMM 系统中的上下文相关音素.

以 CTC 为例：

$$\boldsymbol{w}^{*}=\arg\max_{\boldsymbol{w}}\left\{\frac{P(\boldsymbol{l}_{\boldsymbol{w}}|\boldsymbol{x})P(\boldsymbol{w})}{P(\boldsymbol{l}_{\boldsymbol{w}})}\right\}\tag{6}$$

$$=\arg\max_{\boldsymbol{w}}\left\{P(\boldsymbol{w})\max_{\boldsymbol{l}_{\boldsymbol{w}}}\frac{P(\boldsymbol{l}_{\boldsymbol{w}}|\boldsymbol{x})}{P(\boldsymbol{l}_{\boldsymbol{w}})}\right\}\tag{7}$$

这里以单音素的 CTC 为例（CTC 标签集合包括音素标签和 blank 符号）. $P(\boldsymbol{l}_{\boldsymbol{w}})$ 是音素序列的先验概率.

对于某个特定的 CTC 标签序列，其前向概率可定义并近似为[17]

$$P(\boldsymbol{l}|\boldsymbol{x})=\sum_{\pi\in\mathcal{B}(\boldsymbol{l})}\prod_{t=1}^{T}y_{\pi_t}^{t}\cong\max_{\pi\in\mathcal{B}(\boldsymbol{l})}\prod_{t=1}^{T}y_{\pi_t}^{t}\tag{8}$$

其中，$\mathcal{B}$ 的定义见式(4).

因此，式(7)可进一步被推导为如下的帧同步维特比束搜索（frame synchronous Viterbi beam search）. 这里，整体优化搜索空间——WFST，在每一帧都需要被遍历.

$$\boldsymbol{w}^{*}\cong\arg\max_{\boldsymbol{w}}\left\{P(\boldsymbol{w})\max_{\pi\in\mathcal{B}(\boldsymbol{l})}\frac{1}{P(\boldsymbol{l}_{\boldsymbol{w}})}\prod_{t=1}^{T}y_{\pi_t}^{t}\right\}\tag{9}$$

2.2.2　解码复杂度分析

在 FSD 框架中，特征速率定义为特征帧的数量除以语句的长度，标注速率定义为标签输出数量除以语句的长度，而解码速率定义为 WFST 解码的帧数除以语句的长度. 在帧同步解码框架中，这三个速率均相等. 也就是说，$\prod_{t=1}^{T}y_{\pi_t}^{t}$ 与帧 t 有关，而最大迭代次数则与序列可能的对齐方式和词汇量的大小有关. 因此，解码复杂度 $\mathbb{C}$ 可表示为

$$\mathbb{C}\propto T\cdot|L'|\cdot|W|\tag{10}$$

其中 T 是语句中帧的数量，L' 是模型单元的集合，W 为词汇量.

尽管被广泛使用，FSD 方法仍有一些缺点：(1) 它是一个等间隔搜索算法，处理变长特征序列较为低效；(2) 当序列被分解为帧层面作为特征序列时，模型粒度较小，导致搜索空间很大；(3) 在每帧均进行贪心束剪枝，很难平衡搜索效率和搜索误差. 因此，本文通过将特征层面的搜索过程改变为标签层面，提出了基于端到端建模的标签同步推理方法，接下来本文将对该框架及其应用进行详细介绍.

3　基于端到端建模的标签同步推理

本部分，作者提出将搜索过程从特征层面改为标签层面，称为标签同步解码（Label Synchronous Decoding，LSD）. 接下来该部分将分别对 DSM 和 GSM 中的 LSD 进行公式推导，具体实现方案及一些解码加速的经验方案将在下一节中进行讨论.

在测试阶段，根据上文式(5)给出的 ASR 解码，下面分别对 DSM 和 GSM 中的 LSD 给出公式推导过程.

3.1　DSM 的标签同步解码

在基于音素的 CTC 模型中，从式(5)可以推导出式(7). 而根据 CTC 中输出标签之间的条件独立性假设，$P(\boldsymbol{l}|\boldsymbol{x})$ 可以如下获得：

$$P(\boldsymbol{l}|\boldsymbol{x})=\prod_{l\in\boldsymbol{l}}P(l|\boldsymbol{x})\tag{11}$$

因此在标签级别上，维特比搜索如下所示：

$$\boldsymbol{w}^{*}=\arg\max_{\boldsymbol{w}}\left\{P(\boldsymbol{w})\max_{\boldsymbol{l}_{\boldsymbol{w}}}\frac{\prod_{l\in\boldsymbol{l}_{\boldsymbol{w}}}P(l|\boldsymbol{x})}{P(\boldsymbol{l}_{\boldsymbol{w}})}\right\}\tag{12}$$

在 $P(l|\boldsymbol{x})$ 的计算中，本文提出在帧级神经网络的输出上进行一步后处理. 其中公共 blank 帧的集合定义如下：

$$U=\{u:y_{\text{blank}}^{u}>\mathcal{T}\}\tag{13}$$

其中 y_{blank}^{u} 是神经网络在第 u 帧输出 blank 单元的概率. 在 CTC 模型中的 softmax 层，如果 blank 单元的声学得分足够大且接近常数 1，则可以认为所有竞争路径共享相同跨度的 blank 帧. 因此，忽略这些帧的分数并不会影响解码中的声学得分排序.

$$P(l|\boldsymbol{x})=\sum_{\pi\in\mathcal{B}(\boldsymbol{l})}\prod_{\pi}P(\pi|\boldsymbol{x})\simeq\sum_{\pi\in\mathcal{B}(\boldsymbol{l})}\prod_{\pi\in U}y_{b_l}^{u}\prod_{\pi\notin U}y_{p_l}^{u}\tag{14}$$

由于 $\prod_{\pi\in U}y_{b_l}^{u}\simeq1$，式(14)可被推导为式(15)：

$$P(l|\boldsymbol{x})\simeq\sum_{\pi\in\mathcal{B}(\boldsymbol{l})}\prod_{\pi\notin U}y_{p_l}^{u}\tag{15}$$

3.2　GSM 的标签同步解码

在 GSM 中，相邻 HMM 之间的输出标签也是条件独立的：

$$\mathrm{P}(\boldsymbol{x}|\boldsymbol{l})=\prod_{l}P(\boldsymbol{x}|l)\tag{16}$$

类似地，在标签级别上进行的维特比搜索如下所示.

$$\boldsymbol{w}^{*}=\arg\max_{\boldsymbol{w}}\left\{P(\boldsymbol{w})\max_{\boldsymbol{l}_{\boldsymbol{w}}}\prod_{l\in\boldsymbol{l}_{\boldsymbol{w}}}P(\boldsymbol{x}|l)\right\}\tag{17}$$

在标签中，$P(\boldsymbol{x}|l)$的计算如下所示：

$$P(\boldsymbol{x}|l)=\sum_{\pi:\pi\in L',\mathcal{A}(\pi_{1;T})=l}\prod_{t=1}^{T}P(\boldsymbol{x}|\pi_t)P(\pi_t|\pi_{t-1})$$
$$=\sum_{\pi:\pi\in L',\mathcal{A}(\pi_{1;T})=l}\prod_{t=1}^{T}P(\pi_t|\boldsymbol{x})P(\boldsymbol{x})\frac{P(\pi_t|\pi_{t-1})}{P(\pi_t)}$$
$$\simeq\sum_{\pi:\pi\in L',\mathcal{A}(\pi_{1;T})=l}\prod_{t=1}^{T}P(\pi_t|\boldsymbol{x})\frac{P(\pi_t|\pi_{t-1})}{P(\pi_t)} \quad (18)$$

最近，研究人员们提出了一些新的 HMM 拓扑结构[18-19]，它们具有与式(4)中 CTC 的$\mathcal{B}$函数类似的一对多映射. 以文献[18]为例，每个 CD 音素由两个状态建模，如图 1(c)所示，且转移概率设置为常数值 0.5，因此在式(18)中可被省略. 具体来说，其中一个状态模拟 blank 建模，如图 1(b)中的$\langle b\rangle$，另外一个状态则模拟输出标签单元，如图中的 l. 不同之处在于文献[18]中的每个 CD 音素都保留了自己的 blank 版本. 因此 HMM 中的状态由标签输出状态或者与 CTC 类似的 blank 状态组成. 虽然在我们的实验中，这些模型的输出分布比 CTC 中的更平滑，但 DSM 中提出的式(13)和(14)可以被扩展到 GSM.

这里，本文提出对神经网络的输出 $P(\pi_t|\boldsymbol{x})$进行后处理，其中 π_t是帧 t 的推理模型单元. 由于这些模型中的模拟 blank 状态，式(17)中的维特比束搜索不必包括候选标签输出序列的所有帧. 因此，给出某一帧的模型推理分布时，是否从维特比搜索中排除某帧的判决如下：

$$U=\left\{u:\sum_{l\in L}(y^u_{b_l}-y^u_{p_l})>\mathcal{T}\right\} \quad (19)$$

其中 $y^u_{p_l}$是帧 u 处标签输出状态 l 的神经网络输出，$y^u_{b_l}$是对应的 blank 状态的输出. 在第 u 帧是否有标签输出，是由所有 blank 状态与标签输出状态的概率差异的总和决定的. $\mathcal{T}$是在开发集中得到的阈值. 因此，$P(\boldsymbol{x}|l)$的计算可以根据 $\pi\in U$ 与否分为如下两部分：

$$P(\boldsymbol{x}|l)\simeq$$
$$\sum_{\pi:\pi\in L',\mathcal{A}(\pi_{1;T})=l}\left\{\prod_{\pi\notin U}\frac{y^u_{b_l}P(b_l|\boldsymbol{x})}{P(b_l)}\prod_{\pi\in U}\frac{y^u_{p_l}P(p_l|\boldsymbol{x})}{P(p_l)}\right\} \quad (20)$$

公式中第一部分是标签输出状态. 这种情况下，每个标签输出均在 WFST 中进行维特比搜索. 另外一部分为 blank，假设没有标签输出. 但不同于 CTC，不同标签输出保留自己的 blank 状态版本. 即使是 blank 帧，也可能包含不同的输出标签信息. 因此，$\prod_{\pi\notin U}\frac{y^u_{b_l}P(b_l|\boldsymbol{x})}{P(b_l)}$的分数不能被丢弃. 本文 4.2 节将提出一种高效的算法对这一项进行计算.

这里所提出的后处理可以被视为输出标签概率 $P(\pi|\boldsymbol{x})$的近似，从而使得维特比束搜索得以在标签级别上进行.

3.3 FSD 和 LSD 的对比

本文提出将特征层面的搜索过程改变为标签层面，即搜索空间是由不同历史的标签组成的，使得解码速率等于标注速率，从而小于特征速率. 具体来说，本文所提出的 LSD 的解码复杂度如下：

$$\mathbb{C}\propto(T-|U|)\cdot|L'|\cdot|W| \quad (21)$$

其中空白帧的数量 $|U|$ 总是接近于 T. 对比式(10)和式(21)，可以发现 FSD 得到了很大的加速. FSD 和 LSD 的主要区别总结如下：

(1) 不同的信息率. 在 FSD 中，声学信息和语言信息均在每帧进行处理，使得二者的处理速率和声学特征的帧率相同. 而在 LSD 中，声学信息是以声学特征的帧率进行处理的，而语言信息则按声学模型推理的标注速率进行处理. 声学信息和语言信息处理的不同速率去除了大量的搜索冗余.

(2) 可调整的搜索间隔. 在 FSD 框架下，WFST 网络是以等间隔方式遍历的(虽然带有跳帧的深度神经网络在解码[20]时是以更长的间隔遍历语言搜索空间，但其间隔仍然是相等的). 而在 LSD 中，搜索间隔可通过灵活的自我调整(在不造成性能下降的前提下)来去除 blank 帧带来的语言搜索空间搜索冗余，大大提升了解码效率.

4 标签同步解码算法及其应用

4.1 模 型

本文将图 1(d)～(e)所示的几种改进的 HMM 拓扑结构应用在了 GSM 中. 具体来说，在图 1(c)中，每个 CD 音素都有独立的 blank 状态，称为 CD 音素 blank(CD phone blank). 为减少模型单元的数量并进一步加快算法速度，将中心音素相同的 blank 状态绑定在一起，称为音素级 blank(phone blank)；最后如果绑定所有的 blank 状态则称作全局 blank(global blank). 此外，鉴于标签延迟带来的性能改进[11]，图 1(d)中提出 HMM-PB 模型的延迟标签变种，即 HMM-BP. 也就是说，模型在确定性标签输出之前输出混淆输出 blank. 另外，作为对 CTC 的完整模拟，图 1(e)中提出了 HMM-BPB，允许在标签输出前后都存在 blank. 我们的初步实验结果表明，这两种类型的 blank 展现出了不同的功能. 因此没有将它们绑定在一起. 而输出标签单元后面的

所有 blank 则都被绑定在了一起，以减少所需的模型单元数量.

4.2 算　法

DSM 的标签同步解码算法如算法 1 所示. S 和 E 是预编译的 WFST 网络的起始和结束节点. Q 指有效令牌，$\hat{B}$ 指解码路径，T 是总帧数. $NNPropagate(t)$ 是每帧的声学模型推理过程. $isBlankFrame(F)$ 用于检测每帧是否为 blank. $ViterbiBeamSearch(F,Q)$ 是 FSD 中的标准维特比搜索算法，但在 LSD 中仅在标签级别执行. $finaTransition(E,S,Q)$ 用于搜索 WFST 的终止节点[21].

算法 1.　DSM 的标签同步维特比束搜索算法.

输入：起始节点，结束节点，令牌队列，时间帧

输出：识别结果

```
PROCEDURE LSD for DSM(S,E,Q,T)
Q←S / * 起始节点初始化 * /
FOR each t∈[1,T] DO / * 逐帧神经网络前向传播 * /
    F←NNPropagate(t)
    IF !isBlankFrame(F) THEN / * 逐音素解码 * /
        Q←ViterbiBeamSearch(F,Q)
B̂ ←finalTransition(E,S,Q) / * 到达结束节点 * /
backtrace(B̂)
```

用于 GSM 的标签同步解码算法如算法 2 所示. 与算法 1 相比，在每个 blank 帧中，输出序列可以包含不同的 blank 单元. 因此对相邻的 blank 帧计算 $\prod_{\pi\notin U}\frac{y_{b_l}^{u}P(b_l|x)}{P(b_l|x)}$. 在非 blank 帧中，首先将各个 blank 单元各自累积得到的概率得分分别添加到当前帧的所有候选序列分数中，之后再进行维特比搜索算法.

算法 2.　GSM 的标签同步维特比束搜索算法.

输入：起始节点，结束节点，令牌队列，时间帧

输出：识别结果

```
PROCEDURE LSD for GSM(S,E,Q,T)
Q←S / * 起始节点初始化 * /
FOR each t∈[1,T] DO / * 逐帧神经网络前向传播 * /
    F←NNPropagate(t)
    IF !isBlankFrame(F) THEN / * 逐音素解码 * /
        F←addAccumulatedBlankScore(V,F)
        reset(V)
        Q←ViterbiBeamSearch(F,Q)
    ELSE / * 积累 blank 得分 * /
        V←accumulateBlankScore(V,F)
B̂ ←finalTransition(E,S,Q) / * 到达结束节点 * /
backtrace(B̂)
```

4.3 剪　枝

在维特比搜索中，本文除了使用传统的束剪枝算法[7]和直方图剪枝算法[22]（自适应束剪枝[23]）之外，提出了另外两种剪枝方法. 在 LSD 中，blank 帧占总帧数的百分比与加速比成正比，而 blank 帧可通过式(13)和(19)进行判定. 作为束剪枝算法的变体，这里提出了基于 blank 帧阈值 $\mathcal{T}$ 的剪枝算法，称为 blank 剪枝. 当阈值 $\mathcal{T}$ 固定时，推理分布的尖峰属性决定了加速比，而尖峰属性显示了神经网络输出分布的置信度. 在神经网络的模型训练阶段，本文又提出了基于假设剪枝的熵剪枝算法. 在文献[24]中，作者通过惩罚确定的输出分布来防止过拟合和提高神经网络的泛化能力. 受这项工作的启发，我们在 LSD 框架中对输出分布的熵进行了控制，作为候选序列的剪枝方法. 具体来说，在模型训练中将输出分布的熵添加到负对数似然 $\mathcal{L}(\theta)$ 中，公式如下：

$$\mathcal{L}(\theta)=-(p_\theta(\pi|x)-\beta H(p_\theta(\pi|x))) \tag{22}$$

其中 $H(\cdot)$ 是输出分布 $(p_\theta(\pi|x)$ 的熵，β 是正比例因子. 与文献[24]不同的是，基于熵剪枝算法的训练目的是最小化模型的原有训练准则以及输出分布的熵. 而通常情况下，基于熵剪枝算法是基于一个已经训练好的模型对参数进行微调. 在使用新的准则训练之后，LSD 框架可在少量性能损失的情况下得以加速. 在接下来的实验部分，本文将详细比较这四种剪枝方法.

5 实验及分析

本文实验使用 300 小时的英语 Switchboard 数据集作为训练数据[25]，使用 NIST 2000 CTS 作为测试集，对 NIST 2000 CTS 测试集所包含的 switchboard（称为 swb）和 call-home（称为 callhm）两个子集分别进行了评估. 在所有实验中使用的是经过工程优化的标准 WFST 解码器；实验过程中没有生成词图，也没有使用语言模型重打分[26]技术. 解码过程中使用在 Switchboard 和 Fisher 转录文本上训练的插值的 4 阶语言模型；在 DSM 算法验证中，默认使用了经过剪枝的 3 阶语言模型；在 GSM 算法验证中，默认使用 4 阶语言模型，使得结果与文献[18]具有可比性. 解码使用的机器配置为 Intel(R) Xeon(R) CPU E5-2690 v2@3.00 GHz.

本文 5.1 节的 DSM 实验中，使用具有 1.2M 参数的小型 CTC 模型，使得其适用于嵌入式设备，与文献[27]可比；使用 40 维的对数滤波器组特征，特

征提取窗宽为 25 ms，帧移为 10 ms；使用 46 个单音素作为声学建模单元；声学模型使用 3 层带有投影层的长短时记忆网络，每层包括 400 个节点并通过投影层压缩为 128 个节点[9]；使用 EESEN[28] 作为训练工具，训练过程与文献[29]相似.

本文 5.2 节的 GSM 实验是在一系列由 KALDI 流程[30]训练的基于 HMM 的大型模型上进行的，这些模型均适用于服务器应用. 声学模型建模单元是上下文相关音素. 为了提升解码性能[18-19]，相对于输入层特征 10 ms 每帧的帧率，将输出层的输出帧率下降到 30 ms 每帧. 声学模型分别使用每层 625 个节点的 7 层时延神经网络(TDNN)；以及 3 层带有投影层的双向长短时记忆网络(BLSTM)，其中每层的前向后向层均具有 1024 个节点，并通过投影层压缩为 256 个节点.

本文实验过程中，使用词错误率(Word Error Rate，*WER*)来评估不同解码框架下的模型性能，使用搜索过程中的实时率(Real Time Factor (*RTF*) of the Search process，*SRTF*)和每帧中的活动令牌的平均数量(Active Tokens，$\sharp AT$)来评价搜索速度. 在降低帧率的声学模型中，$\sharp AT$ 使用降帧率之前的帧数进行计算；*SRTF* 指解码时间与音频时间的百分比. 值得注意的是，这里的解码时间不包括神经网络传播的时间[31-32]①. 本文所提出的框架主要加速搜索过程而非神经网络传播，因此不针对不同声学模型的计算速度进行比较，同时这里采用 *SRTF* 而不是 *RTF* 来评价搜索速度. 由于在维特比搜索的搜索迭代过程-即令牌传递算法[33]中，搜索迭代速度与有效令牌的数量相关，因此搜索速度评价指标 *AT* 与 *SRTF* 总是正相关. 为了更加清晰地对比结果，我们还提供了上述指标的相对变化率(Δ)作为参考.

5.1 DSM 实验

(1) 加速. 表 1 给出了 CTC 模型下，LSD 系统相对 FSD 系统的加速对比. 基于 FSD 的 CTC 模型是基线系统. 在我们之前的音素同步解码工作[15]中，曾对比了 CTC 模型的性能和基于 HMM 的系统的性能[34-35]②.

表 1 判别式序列模型 DSM 中 LSD 与 FSD 的对比

测试子集	解码性能		搜索加速			
	FSD↦LSD		FSD↦LSD		FSD↦LSD	
	WER	Δ/%	*SRTF*	Δ/%	$\sharp AT$	Δ/%
swb	18.7	+0.5	0.075	−71	2221	−77
callhm	33.3	+0.0	0.073	−70	2211	−77

swb 测试子集中，在词错误率相对损失不到 0.5%的情况下，与 FSD 框架相比，LSD 框架实现了相对 70%以上的 *SRTF* 下降(也就是 3.4 倍的解码加速). 解码加速主要是因为解码过程中减少了搜索迭代次数，解码过程中的活动令牌的数量也体现了这一结果. 同时在 callhm 子集的实验中也能观察到一致的加速效果.

(2) 速度鲁棒性. 上面的实验解码过程中都使用一个中等大小的语言模型(3 阶，3.1 M 语言模型)，为了测试 LSD 框架相对于 FSD 加速效果的鲁棒性(也即对复杂的语言搜索空间下的可扩展性)，如图 2 所示，这里将解码语言模型从 2 阶变大到 4 阶[36]③，*N*-gram 个数从 0.2M 变大到 4.7M，使用每帧中的平均活动令牌数($\sharp AT$)来评价解码速度. 从图 2 中可以看出，随着语言模型的增大，LSD 的 $\sharp AT$ 值几乎没有变化，而与此同时，FSD 的 $\sharp AT$ 值则明显加速增长. 此外，FSD 的 $\sharp AT$ 值总是远远超过 LSD 的 $\sharp AT$ 值. 也就是说，LSD 实现的加速对 LM 搜索空间的增加是鲁棒的，GSM 的实验也得出了类似的结论，因此 LSD 适合应用于更复杂的 LM.

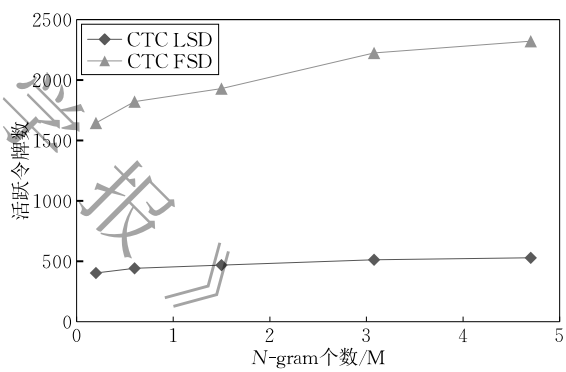

图 2 LSD 和 FSD 框架中平均活跃令牌数随 LM 变大的变化趋势(为清晰起见，这里仅绘制 swb 子集，calhm 子集具有类似变化趋势)

(3) 结合跳帧方法. 该部分对比了跳帧方法下的 LSD 与 FSD 框架，实现结果表明可以将跳帧方法与 FSD 框架进行结合使用. 值得注意的是，在后面的实验中，LSD 也可以应用于跳帧或降低帧率的 GSM 声学模型中.

① 神经网络传播时间总是与商业应用中搜索过程的时间相当，因为前者可以使用 GPU 并行独立进行[31-32].

② 我们试图让 CTC 在数百小时的数据上通过交叉熵训练系统的尝试最终没有成功. 结论与文献[18]类似，但我们也注意到了近来一些来自于更好神经网络结构的进展[34]. 此外，在更多数据下，CTC 模型会表现得更好[14,27]. 我们此前在大型数据集中的工作也有类似的发现[35].

③ 这里选择的是 *N*-gram LM，但结果可以很容易地扩展到其它 LM，如 RNN[36].

实现方法类似于文献[37]，这里使用 LSTM-CTC 的 2 倍跳帧(FS)，并且在神经网络后验概率输出层上没有根据原始特征帧率补全后验概率，因此 FS 也可以加速解码过程①；在没有性能损失的情况下，应用于 CTC 模型的 FS 可以获得近 2 倍的解码性能加速②. 这与文献[37]中的观察结果一致，并且在 DNN-HMM[20] 和 LSTM-HMM[37] 也有类似的结果. LSD 可以进一步与 FS 组合，并且获得更高的效率，即在搜索过程中进一步减少 57%(累计为 78%)的时间(表 2).

表 2　LSD 与跳帧方法的对比

测试子集	解码性能		搜索加速		
	FSD↦FS+LSD		FSD↦FS↦FS+LSD		
	WER	Δ/%	*SRTF*	Δ_{FS}	$\Delta_{+LSD}(\Sigma)$
swb	18.7	−1.6	0.075	−48	−57 (−78)
callhm	33.3	−0.6	0.073	−47	−57 (−77)

(4) 候选序列剪枝. 图 3 比较了本文提出的 LSD 框架与传统的剪枝技术，即束剪枝(表示为 beam)和直方图剪枝(表示为 histogram). 在 LSD 中，通过调整式(13)和(19)中定义的 $\mathcal{T}$ 来调整加速比和性能，这也可以被视为另一种候选序列剪枝方法(表示为 blank). 第 4.3 节中提出的基于熵剪枝算法表示为 entropy.

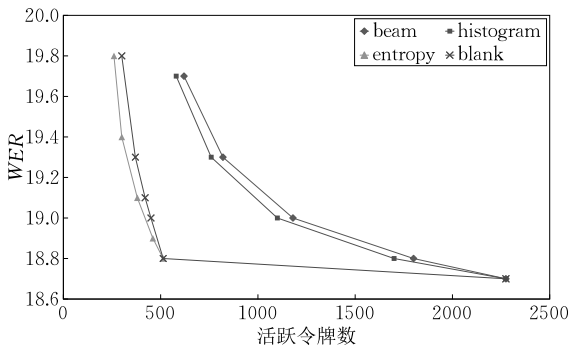

图 3　对于 swb 子集，CTC 中，使用不同剪枝技术时 *WER* 随平均活跃令牌数的变化趋势. callhm 子集结果类似③

从图 3 中可以看出，基于 LSD 的方法，entropy 和 blank 与基于 FSD 的方法 beam 和 histogram 相比保持显著优势. 其原因在于，基于 FSD 的候选序列剪枝对所有候选序列进行统一处理，而相反，LSD 框架可以看作是将候选序列划分为特征级别和标签级别. 因此，如式(13)及式(12)所示，LSD 框架中，可以在特征层和标签层进行剪枝. 也即，基于 LSD 的候选序列剪枝方法受益于将搜索过程从特征层转移到标签层.

另外结果显示，为了加速解码，两种框架中的方法都会降低解码性能，而且基于 LSD 的方法解码性能下降更严重. 如前所述，基于 FSD 和 LSD 的方法之间的关键区别在于后者的阈值 $\mathcal{T}$ 仅与特征层上的候选序列剪枝有关. 特征层候选序列剪枝的加速比几乎是固定的，在不损害性能的情况下，70%～80%的候选序列可以被剪枝掉. 图 3 中 blank 曲线具有明显的拐点($\sharp AT=513$, $WER=18.8$)，也源于同样的原因. 在图的最右侧，未进行特征层剪枝时，blank 曲线最终达到了与基于 FSD 方法的曲线所达到的相同位置. 由于上面讨论的明显拐点及固定加速比，可以很容易得到等式(13)中的阈值 $\mathcal{T}$. 此外，特征层的 entropy 和 blank 剪枝可以进一步与标签层的 beam 和 histogram 方法相结合，以得到最佳的解码加速效果. 为了使对比更加清晰，图中没有给出融合系统的曲线.

最后，entropy 的效率略高于 blank(相对约 10%). 我们认为原因在于神经网络中剪枝能更好地利用神经网络输出分布中的信息，并且产生更好的精度和效率. 而 blank 剪枝则仅利用了输出分布中的最佳分数而未使用整个分布的信息.

5.2　GSM 实验

(1) 在各种模型和准则中的应用. LSD 应用于生成式序列模型(GSM)时，本文使用了多种不同的神经网络模型结构和模型训练准则进行对比；默认使用上下文相关音素作为模型建模单元. 表 3 给出了在 NIST 2000 CTS 测试集合上的结果. 总体而言，LSD 框架也可以取得比较显著的解码加速，但是与表 1 中的结果相比，解码加速性能变差. 这是因为 FSD 基线的帧率已经降低到原来的 1/3[19](类似于第 5.1 节(3)中的结果，帧率改变技术可以与提出

表 3　生成式序列模型中 LSD 与 FSD 的对比

模型	准则	解码性能		搜索加速			
		FSD↦LSD		FSD↦LSD		FSD↦LSD	
		WER	Δ/%	*SRTF*	Δ/%	$\sharp AT$	Δ/%
TDNN	CE	17.8	+1.0	0.16	−38	3705	−41
	LF-MMI	15.6	+1.0	0.13	−43	3386	−45
	+sMBR	15.4	+1.0	0.12	−41	3295	−43
	LF-bMMI	15.0	+1.0	0.11	−42	3198	−44
	LF-sMBR	15.3	+1.0	0.12	−41	3288	−44
BLSTM	LF-MMI	15.2	+1.0	0.12	−44	3290	−47
	LF-bMMI	14.3	+1.0	0.11	−43	3205	−45

① 该过程主要是处理 HMM 状态级循环和转换. 初步实验表明，去除它并不会影响性能.

② *WER* 的轻微改善与文献[19]中观察到的结果相似.

③ 横轴之所以描绘平均活动令牌数而不是实时率；首先平均活动令牌数与搜索实时率正相关；另外实时率或者搜索实时率是不稳定的，与机器设备 CPU 性能，或机器状态等外部因素相关. 所以这里使用平均活动令牌数.

的 LSD 框架结合). 而且与表 2 相比,加速比也略小,原因是这些模型的推理分布概率不像 CTC 那样尖锐. 如何在 GSM 中获取更尖锐的推理分布概率将在该部分的(2)和(3)中进行讨论.

具体地,如表 3 所示,第 1 行列出了文献[19]中提出的低帧率模型(LFR)的结果;第 2 行是使用 LF-MMI 准则[18]训练出来的结果,显示出比 LFR 更快的搜索速度;此外,还可以看到,从 FSD 到 LSD,基于 LF-MMI 准则训练的模型可以取得更快的加速. 与文献[38]中观察到的类似,这都源于序列区分性训练准则得到的模型相对于交叉熵准则训练的模型有更尖锐的输出概率分布. 第 3 行表示为+sMBR,是在 LF-MMI 模型的基础上,使用基于 LM 的 sMBR 准则微调模型参数得到的结果;第四、五行列出了基于增强的 MMI[39]及 sMBR 准则变体的无需词图的区分性训练准则取得的结果,分别表示为 LF-bMMI 和 LF-sMBR. 可以观察到,本文提出的 LSD 框架在以上模型和准则中可以取得一致的解码加速. 另外我们还使用了 BLSTM 模型,也都取得了相似结果.

(2)候选序列剪枝. 如图 4 所示,与 5.1 节中(4)类似,我们在生成式序列模型下进行了一系列候选序列剪枝相关的实验,其变化趋势类似于 DSM 中的结果,读者可以参考那里的讨论. 与图 3 中一个区别在于,beam,histogram,blank,entropy 在图 4 中的最左边位于相同点,这表明在降低帧率的情况下,特征层候选序列剪枝的最大比例较小. 然而,在解码性能的 *WER* 达到最佳时,仍然有接近两倍的解码加速.

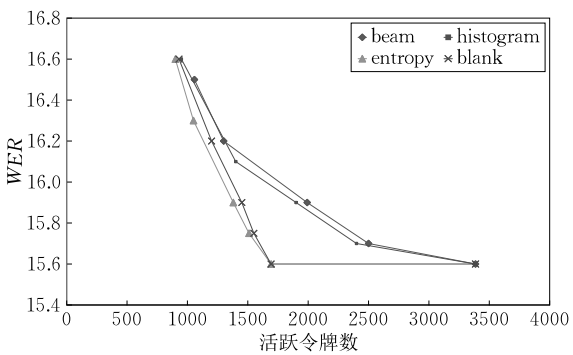

图 4 LF-MMI 中,使用不同剪枝技术时 *WER* 随平均活跃令牌数的变化趋势

(3)进一步设计. 本节将对比 4.1 节中讨论的各种转移模型,以及由此获得的效率的进一步提高的情况. 该部分所有实验均在 LF-MMI 准则上进行,但实验结论可以扩展到其他神经网络和训练准则上.

表 4 生成式序列模型中的 blank 粒度

系统	blank	解码性能		搜索加速			
		FSD↦LSD		FSD↦LSD		FSD↦LSD	
		WER	Δ/%	*SRTF*	Δ/%	#*AT*	Δ/%
TDNN LF-MMI	CD phone	15.6	+1.0	0.13	−43	3386	−45
	phone	15.7	+0.9	0.09	−47	2785	−50
	global	16.8	+0.8	0.09	−49	2512	−54

表 4 中列出了不同 blank 粒度的对比结果,即第 4.1 节中定义的上下文相关音素 blank(CD phone blank)、音素 blank(phone blank)和全局 blank(global blank). 与 CD phone blank 基线相比,phone blank 在取得近似的解码性能的同时,实现了显著的搜索过程加速;这里搜索加速主要源于较少的模型建模单元,即模型状态数从 6K 减少到 3K. 此外,从表中可以看出,global blank 会带来明显的性能下降;global blank 需要足够的数据来覆盖不同相邻音素之间的上下文环境(我们认为这也是 CTC 准则在这个语料库中表现更差的原因之一);CD phone blank 可以缓解 blank 训练数据不足的问题,但会导致搜索速度变慢;因此,绑定中心音素相同的 CD phone blank 在加速搜索过程的同时,也可以更好地建模 blank 模型;因此,phone blank 是解码性能和搜索速度之间的最佳平衡. 此外,从表 4 中可以看出,在 LSD 框架下,较少的模型单元可以持续带来明显的搜索过程时间缩短:43%→47%→49%. 最后,可以得知 phone blank 是基于 GSM 的 LSD 框架的最佳选择.

表 5 生成式序列模型中的 blank 拓扑结构

测试集	拓扑	解码性能		搜索加速			
		FSD↦LSD		FSD↦LSD		FSD↦LSD	
		WER	Δ/%	*SRTF*	Δ/%	#*AT*	Δ/%
TDNN LF-MMI	PB	15.6	+1.0	0.13	−43	3386	−45
	BP	15.6	+1.0	0.13	−46	3392	−49
	BPB	15.6	+1.0	0.13	−47	3388	−51

表 5 对比了第 4.1 节中提出的不同 HMM 拓扑结构. 在 FSD 框架下,所有拓扑结构都有相似的解码结果和相同的搜索速度. 对比表 5 前两行可以看出,与基线 PB 拓扑结构相比,在 LSD 框架下,BP 可以获得更大的搜索加速. 我们认为这个更优的搜索加速源于标签延迟现象,类似于文献[11]中观察到的现象,这使得模型能更可靠地推断标签输出状态并减少混淆. 因此,这能带来更尖锐的输出分布. 从表中还可以看出,BPB 拓扑结构可以进一步改善搜索速度;一些解码路径的例子也表明这种拓扑结构可以使每个上下文相关的隐马尔可夫模型输

出更多的 blank 状态. 最后，与表 2 中 CTC 的结果相比，GSM 中 LSD 框架能减少 49%的搜索时间.

6 结　论

序列标注任务的最大特点在于各个数据帧之间的序列相关性. 序列模型如 HMM 和 CTC 等可以用来建模相邻帧之间的标签跳转关系. 而近来神经网络的最新进展在上下文和历史序列模型中实现了更强的建模效果. 因此，可以将序列分解成更大的模型粒度并对它们进行直接推理，例如子词或词标签等. 然而，当前主流的推理方法仍然是在帧层面上进行的维特比搜索算法，该方法较大的计算复杂度阻碍了语音识别等序列标注任务的广泛应用. 本文中，作者提出将搜索过程从帧层面改变到标签层面，以加速解码. 在标签推理阶段，本文提出了有效的后处理方式以消除声学模型的 blank 输出并获得输出标签的近似概率；且提出的 LSD 框架可以统一应用于基于 HMM 和 CTC 的声学模型. 在 switchboard 数据集上的实验显示，在不降低解码性能的情况下，LSD 在上述所有模型上都有 2～4 倍的搜索加速. 此外，本文还同时研究了搜索空间，候选序列剪枝，转移模型，降帧率等对加速比的影响，并在所有情况下取得一致性加速.

该研究未来的工作包括：(1) 本文提出的熵剪枝，blank 剪枝与传统的束剪枝之间的理论比较；(2) 将 LSD 框架扩展应用到更多模型上，如 neural segmental model[40]，RNN transducer[41] 和 Gram-CTC[34]；(3) 将框架应用于其他的序列标注任务中.

致　谢　感谢苏州思必驰信息科技有限公司语音技术组提供的基础设施支持和宝贵的技术讨论. 实验的主要计算在上海交通大学高性能计算中心的 π 超级计算机上完成！

参　考　文　献

[1] Graves A. Supervised Sequence Labelling with Recurrent Neural Networks: Supervised Sequence Labelling. Berlin, Germany: Springer, 2012

[2] Woodland P C, Odell J J, Valtchev V, Young S J. Large vocabulary continuous speech recognition using HTK// Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Adelaide, Australia, 1994: 125-128

[3] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 2012, 29(6): 82-97

[4] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks//Proceedings of the Neural Information Processing Systems (NIPS). Montreal, Canada, 2014: 3104-3112

[5] Chan W. End-to-End Speech Recognition Models [Ph. D. dissertation]. Carnegie Mellon University, Pittsburgh, USA, 2016

[6] Chen Z, Liu Q, Li H, Yu K. On modular training of neural acoustics-to-word model for LVCSR//Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Calgary, Canada, 2018: 4754-4758

[7] Forney G. D. The Viterbi algorithm. Proceedings of the IEEE, 1973, 61(3): 268-278

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

[9] Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling//Proceedings of the Annual Conference of the International Speech Communication Association (InterSpeech). Singapore, 2014: 338-342

[10] Qian Y, Bi M, Tan T, Yu K. Very deep convolutional neural networks for noise robust speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(12): 2263-2276

[11] Amodei D, et al. Deep speech 2: End-to-end speech recognition in English and Mandarin//Proceedings of the 33rd International Conference on Machine Learning. New York, USA, 2016: 173-182

[12] Soltau H, Liao H, Sak H. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition //Proceedings of the Annual Conference of the International Speech Communication Association (InterSpeech). Stockholm, Sweden, 2017: 3707-3711

[13] Collobert R, Puhrsch C, Synnaeve G. Wav2Letter: An end-to-end ConvNet-based speech recognition system//Proceedings of the 5th International Conference on Learning Representation (ICLR). Toulon, France, 2017: arXiv eprint. arXiv: 1609.03193

[14] Sak H, Senior A, Rao K, Beaufays F. Fast and accurate recurrent neural network acoustic models for speech recognition //Proceedings of the Annual Conference of the International Speech Communication Association (InterSpeech). Dresden, Germany, 2015: 1468-1472

[15] Chen Z, Zhuang Y, Qian Y, Yu K. Phone synchronous speech recognition with CTC lattices. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(1): 86-97

[16] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780

[17] Graves A, Fernandez S, Gomez F, Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks//Proceedings of the 23rd International Conference on Machine Learning. Pittsburgh, USA, 2006: 369-376

[18] Povey D, Peddinti V, Galvez D, et al. Purely sequence-trained neural networks for ASR based on lattice-free MMI //Proceedings of the Annual Conference of the International Speech Communication Association(InterSpeech). San Francisco, USA, 2016: 2751-2755

[19] Pundak G, Sainath T N. Lower frame rate neural network acoustic models//Proceedings of the Annual Conference of the International Speech Communication Association (InterSpeech). San Francisco, USA, 2016: 22-26

[20] Vanhoucke V, Devin M, Heigold G. Multiframe deep neural networks for acoustic modeling//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Vancouver, Canada, 2013: 7582-7585

[21] Hori T, Hori C, Minami Y, Nakamura A. Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(4): 1352-1365

[22] Steinbiss V, Tran B-H, Ney H. Improvements in beam search//Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP). Yokohama, Japan, 1994: 2143-2146

[23] Van Hamme H, Van Aelten F. An adaptive-beam pruning technique for continuous speech recognition//Proceedings of the 4th International Conference on in Spoken Language Processing(ICSLP). Philadelphia, USA, 1996: 2083-2086

[24] Pereyra G, Tucker G, Chorowski J, et al. Regularizing neural networks by penalizing confident output distributions// Proceedings of the 5th International Conference on Learning Representation(ICLR). Toulon, France, 2017: arXiv preprint. arXiv:1701.06548

[25] Godfrey J J, Holliman E C, McDaniel J. Switchboard: Telephone speech corpus for research and development// Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). San Francisco, USA, 1992: 517-520

[26] Povey D, Hannemann M, Boulianne G, et al. Generating exact lattices in the WFST framework//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). Kyoto, Japan, 2012: 4213-4216

[27] McGraw I, Prabhavalkar R, Alvarez R, et al. Personalized speech recognition on mobile devices//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). Shanghai, China, 2016: 5955-5959

[28] Miao Y, Gowayyed M, Na X. An empirical exploration of CTC acoustic models//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Shanghai, China, 2016: 2623-2627

[29] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding//Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding(ASRU). Scottsdale, USA, 2015: 167-174

[30] Povey D, Ghoshal A, Boulianne G, et al. The KALDI speech recognition toolkit//Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). Hawaii, USA, 2011

[31] You K, Chong J, Yi Y, et al. Parallel scalability in speech recognition. IEEE Signal Processing Magazine, 2009, 26(6): 124-135

[32] Hauswald J, Laurenzano M A, Zhang Y, et al. Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers. ACM SIGPLAN Notices, 2015, 50(4): 223-238

[33] Hori T, Nakamura A. Speech recognition algorithms using weighted finite-state transducers. Synthesis Lectures on Speech and Audio Processing, 2013, 9(1): 1-162

[34] Liu H, Zhu Z, Li X, Satheesh S. Gram-CTC: Automatic unit selection and target decomposition for sequence labelling// Proceedings of the 34th International Conference on Machine Learning(ICML). Sydney, Australia, 2017: 2188-2197

[35] Chen Z, Deng W, Xu T, Yu K. Phone synchronous decoding with CTC lattice//Proceedings of the Annual Conference of the International Speech Communication Association (InterSpeech). San Francisco, USA, 2016: 1923-1927

[36] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model//Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP). Prague, Czech Republic, 2011: 5528-5531

[37] Miao Y, Li J, Wang Y, et al. Simplifying long short-term memory acoustic models for fast training and decoding// Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). Shanghai, China, 2016: 2284-2288

[38] Paulik M. Improvements to the pruning behavior of DNN acoustic models//Proceedings of the 16th Annual Conference of the International Speech Communication Association (ISCA). Dresden, Germany, 2015: 1463-1467

[39] Povey D, Kanevsky D, Kingsbury B, et al. Boosted MMI for model and feature-space discriminative training//Proceedings

of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Las Vegas, USA, 2008: 4057-4060

[40] Tang H, Lu L, Kong L, et al. End-to-end neural segmental models for speech recognition. IEEE Journal of Selected Topics in Signal Processing, 2017, 11(8): 1254-1264

[41] Graves A. Sequence transduction with recurrent neural networks//Proceedings of the 29th International Conference on Machine Learning (ICML). Edinburgh, UK, 2012: eprint. arXiv: 1211.3711

CHEN Zhe-Huai, Ph. D. candidate. His research interests include speech recognition, speech synthesis, and deep learning.

ZHENG Wen-Lu, Ph. D., research assistant. Her research interest is speech recognition.

YOU Yong-Bin, Master, research assistant. His research interest is speech recognition.

QIAN Yan-Min, Ph. D., associate professor. His research interests include speech recognition, understanding and machine learning.

YU Kai, Ph. D., professor. His research interests include cognitive spoken dialogue system, speech synthesis, recognition, understanding and machine learning.

Background

This work has been supported by the National Key Research and Development Program of China (Grant No. 2017YFB1302400), the China NSFC Project (No. U1736202), and the Jiangsu NSFC Project (No. BE2016078).

In this work, recent advances in deep learning based acoustic modeling and stronger sequential models are utilized to do the end-to-end modeling on larger model granularities. Moreover, this paper proposes to change the search process from the frame level to the label level. For the label inference, an efficient post-processing is proposed to obtain the approximated probability of each output label. The proposed framework can be uniformly applied in both HMM and CTC based acoustic models.

The current dominant inference methods in speech recognition is Viterbi beam search conducted on frame level, which results in a large computational complexity and hinders wide applications. Our group has been working on reducing the computational complexity by end-to-end modeling based new decoding frameworks. New algorithms have been applied into practical works and many high-quality research papers were published in key transactions and journals such as *IEEE/ACM Transactions on Audio, Speech, and Language Processing* and *Speech Communication*.