# 说话人识别

2019 年 1 月 29 日

## 1 领域自适应与新词处理

在第七章节中我们谈到说话人的领域自适应（Domain Adaptation），对于语音识别领域，当现有模型与使用场景不一致的时候，我们也会考虑对其做领域自适应。在第二章节我们提到现有的语音识别系统主要由声学模型、语言模型和解码器构成，这里我们分别对声学模型和语言模型怎么做自适应做一下简单介绍。

## 2 声学模型

无论传统的GMM-HMM网络还是现在火热的DNN，声学模型都是一个语音识别系统的重中之重，声学模型为一个分类器，计算从语音信号中提取到的特征向量与对应因素之间的匹配度。如何提高两者之间的相似度，使我们要着力的地方。

### 2.1 区分性训练

区分性训练通过定义某一目标函数，通常称准则，来近似一个与分类代价相关的度量，例如可以定义一个与分类错误相关的量并最小化它，或是定义一个与识别正确率相关的量，并最大化它。通过区分性训练，我们可以从一定程度上弱化模型假设错误所带来的影响。同时，由于区分性训练致力于优化与识别效果好坏相关的度量，因此也就为提高识别器性能提供了更直接的途径。区分性训练更重视模型之间的分类面，以更好的根据设定的目标函数对训练数据进行分类。目前常用的区分性训练准则主要包括：最大互信息量准则（Maximum Mutual Information，MMI），最小因素错误准则（Minimum Phone Error，MPE），最小状态化错误（Stat Level Minimum

Bayes Risk，sMBR）[1]。我们使用区分训练来对特定领域做自适应，通常会把新数据集的特征作为区分性训练的输入，用之前训练的模型对新数据做Alignment和Lattice，然后将生成的对齐文件、解码网络以及新数据的特征文件归档为新的数据格式（egs），然后做训练，如下图所示（为了清晰，删除了一部分参数设置）。

```
if [ $stage -le 1 ]; then
  # hardcode no-GPU for alignment, although you could use GPU [you wouldn't
  steps/nnet3/align.sh  --cmd "$decode_cmd" --use-gpu false \
    --online-ivector-dir $online_ivector_dir \
     --nj $nj $train_data_dir data/lang $srcdir ${srcdir}_ali ;
if [ -z "$lats_dir" ]; then
   steps/nnet3/make_denlats.sh --cmd "$decode_cmd" --determinize true \
      --online-ivector-dir $online_ivector_dir \
      --nj $nj --sub-split $subsplit --num-threads "$num_threads_denlats" --config conf/decode_dnn.config \
      $train_data_dir data/lang $srcdir ${lats_dir} ;
fi
if [ -z "$degs_dir" ]; then
   steps/nnet3/get_egs_discriminative.sh \
      --cmd "$decode_cmd --max-jobs-run $max_jobs --mem 20G" --stage $get_egs_stage --cmvn-opts "$cmvn_opts" \
      --online-ivector-dir $online_ivector_dir \
      --left-context $left_context --right-context $right_context \
      $frame_subsampling_opt \
      --frames-per-eg $frames_per_eg --frames-overlap-per-eg $frames_overlap_per_eg \
      $train_data_dir data/lang ${srcdir}_ali $lats_dir $srcdir/final.mdl $degs_dir ;
  fi
if [ $stage -le 4 ]; then
  steps/nnet3/train_discriminative.sh --cmd "$decode_cmd" \
    --stage $train_stage \
    --effective-lrate $effective_learning_rate --max-param-change $max_param_change \
    --criterion $criterion --drop-frames true \
    --num-epochs $num_epochs --one-silence-class $one_silence_class --minibatch-size $minibatch_size \
    --num-jobs-nnet $num_jobs_nnet --num-threads $num_threads \
    --regularization-opts "$regularization_opts" \
    ${degs_dir} $dir
fi
```

Figure 1: Kaldi中的wsj recipe 提供的区分性训练的脚本

我们在使用区分性训练做自适应过程中发现一些现象，一个是模型的推广问题，区分性训练对新数据学习的过于精细，在未知测试集上难以到达到与所在训练集上一样的提升效果，有时甚至会变的更差，还有一个问题就是模型收敛过快，往往在前几个epoch的时候就已经收敛了，继续训练，模型效果有时并没有提升反而是下降，这就产生过拟合的问题，所以在训练的时候，我们可以调小学习率、及时测试和查看loss，发现模型表现效果变差或loss不再下降，就可以及时停止。

## 2.2 模型拼接

在文章 [2]中介绍了两种模型拼接的方法。如下图的左一所示，model1模型的前几层或者全部层作为model2的前几层，然后训练后面几层，kaldi中的nnet3-init命令支持模型拼接。下图左二所示，将model1作为model2的初始化模型，在此基础上继续训练。这样做的目的使得model2在开始训练的时候，能站在一个比较好的训练起始点，最终能达到一个比较好的收敛效果。

总在我们实际的工程应用中，对一些特殊领域数据做增强，我们通常采用借鉴model1的前几层，作为小模型的初始化模型，通常会将借用的前几层模型参数固定，方法比如将学习率设置为0或者FixAffineComponent（nnet3-copy中

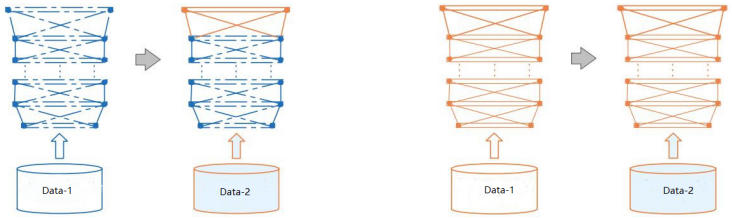


Figure 2: 模型拼接

的支持这个功能）等，或者用大模型作为小模型的初始化模型，参数不固定，大模型作为小模型的初始化模型（0.mdl），这种是我们目前做自适应常用的方法。

# 3 新词添加与语言模型自适应

语音识别是计算机将人发出的语音转换为文字的系统，现有的成熟技术还不支持无约束语音解码，需要施以一定的语言结构或语法限制解码时的搜索路径，以此来提高识别率，所以语言模型成为了解码网络HCLG中的重要组成部分，是语音识别系统中把识别结果从字或词变通顺的语句的重要模块，一个与识别领域相关的语言模型可以对语音识别的效果有质的提升。

语言模型的领域泛化可以在语料更新和新词处理等方面着力。如果有特定场景的文本预料，可以先基于此场景文本预料做语言模型，然后与原始的语言模型做差值融合，通过调整二者权重，达到对特定场景的自适应。如果没有语料，则需要先收集特定场景词表，再基于此词表收集文本语料构建相应的语言模型，也可定向的提高某些词在HCLG中的概率等，具体介绍如下所示：

## 3.1 热词表语言模型的构建

热词表语言模型的构建有两种方式，一种是基于语料的统计语言模型；另一种是基于语法结构语言模型[3]

(1)统计语言模型：根据用户热词表，通过网络搜索相关热词语料，然后基于热词表和原有词表相关词训练新的统计语言模型，再将其与原始语言模型做融合，得到新的语言模型。另一种情况下，若很难获取到热词表语料，而特定场景又有相对固定的语法结构，则可通过穷举句式的方式制作语料文本。再基于这些语料制作统计语言模型。

(2)语法结构语言模型：在某些垂直领域应用场景，由于句式或命令词有

限，则直接用基于语法结构语言模型即可，不同热词根据场景中使用频率可设定不同的权重。上述热词表语言模型构建完成以后，与原始语言模型进行融合，arpa格式直接用ngram进行加权平均；G.fst或HCLG.fst格式用fstunion命令并联两个fst，并修改两个fst进入边权重，实现通用场景与特定场景平衡。

## 3.2 基于相似词（similar pair）的热词表语言模型构建[4] [5]

在原有此表中，针对每个热词选择一个相似词，再将每个热词参照相似词插入到原有的语言模型中，操作可以在lm.arpa或G.fst或者HCLG.fst上进行，热词插入的权重参考原有相似词的权重（w）,另外修改其权重为aw或a+w，通过调节的值来改变新词的插入权重，若存在某些热词很难在原有此表中找到相似词，则需对其用原有词表进行分词，构建热词子词（sub-word）模型，权重操作如上述段落所述，相似词如苹果和香蕉、中关村西路和学府路等，具有同一或相似属性的词便可称之为相似词。若原始词表中有苹果一词，而香蕉为新添加热词，则可用原有语言模型中苹果的统计概率来指代香蕉一词。对于相似词对的选取，可以通过人工先验指定的方式，也可通过计算td-idf/word-vec向量计算其余弦距离得到。

## 3.3 基于类别（classes）信息的热词表语言模型构建

此种方法要求首先构建基于类别的统计语言模（BigClass.arpa/BigClass.G.fst），需要对原词表分类，如大类按名次、代词、形容词等，或只对部分常用类别，如人名、地名、水果类、器具类等分类，用统一的“name”、“location”、“fruit”等表示，每种类别可构建统计语言模型或语法结构语言模型（SmallClass.arpa、SmallClass.G.fst），最后将其嵌入到大的语言模型（BigClass.arpa、BigClass.G.fst）中，形成最终的语言模型（lm.arpa、G.fst）。在对热词进行处理时，首先确定其类别，根据BigClass.arpa或BigClass.fst中的权重动态调其权重。

### 3.3.1 基于文本后处理的热词语音识别

此种方法是基于因素串匹配的热词识别方法，首先基于原词表与原语言模型对语音进行识别，待所有语音识别成文本后，将文本转成想用的因素串，同样的，对要添加热词也转成相应的因素串，遍历热词因素串，在识别文本因素串序列中搜寻最优匹配串，因素串匹配度可采用匹配个数比例、余弦距离（将因素串表示成向量格式）等，选取匹配度最高的热词因素串并转化成相应热词，替换掉原来识别语句中字词，来达到对热词的快速识别

### 3.3.2 基于fst的可定义热词识别

热词识别的难点在与语言模型训练时，没有热词相关的概率信息，在解码时只能通过单字路径拼凑出热词，但单字存在路径稀疏及同音异形字，从而使得解码引擎识别时给出似是而非的结果，另外没在噪音背景较强时，声学模型区分性急剧降低，而语言模型部分有没有太多正确的信息约束，引入了更多的竞争性路径，造成了大量的错误识别。以修改解码图fst的方式实现热词识别，首先定义热词词表映射字典，热词用词表内词进行描述，其他所有词表依旧映射到自身，然后将热词映射字典编译成C.fst，再与原有非热此词表加码图HCLG.fst融合成新的HCLGC.fst，新生成的解码图便具有了对热词的识别能力，且热词的权重可以调节。具体过程如代码所示：

```
rawfstdir=old_graph
mdldir=model
dir=./
mapfile=$dir/new_words
# Make C.fst
./utils/make_lexicon_fst.pl --pron-probs $mapfile 0.5 SIL | \\
    fstcompile --isymbols=$rawfstdir/words.txt --osymbols=$dir/words.txt \\
    --keep_isymbols=false --keep_osymbols=false | \\
     fstarcsort --sort_type=ilabel > $dir/C.fst

# Sort raw HCLGa.fst with sort_type=olabel and compose it with C.fst
fstarcsort --sort_type=olabel $rawfstdir/HCLGa.fst | \\
    fsttablecompose - C.fst > $dir/HCLGCa.fst
# Add self-loop
add-self-loops --self-loop-scale=0.1 --reorder=true \\
 $mdldir/final.mdl < $dir/HCLGCa.fst > $dir/HCLGC.fst
ln -s $dir/HCLGC.fst $dir/HCLG.fst
```

Figure 3: fst自定义热词

# 4 小结

本文介绍了在缺少相应语料的情况下，使用领域泛化的方法，分为声学模型和语言模型方面，声学模型方面主要是增强模型对特定场景的拟合能力，语言模型方法主要是增加特定场景的语料权重或者提高热词在搜索网络中的权重来提高识别效果，其实目前数据为王的情况下，最好的方法就是增加相应的数据，在平时的学习和工作中，还是要做注意各个方面的语料的收集和整理工作，做到未雨绸缪。

# References


[1] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2005.

[2] X. Zhuang, A. Ghoshal, A.-V. Rosti, M. Paulik, and D. Liu, "Improving dnn bluetooth narrowband acoustic models by cross-bandwidth and cross-lingual initialization," *Proc. Interspeech 2017*, pp. 2148–2152, 2017.

[3] "https://www.w3.org/tr/2000/note-jsgf-20000605."

[4] X. Ma, X. Wang, and D. Wang, "Low-frequency word enhancement with similar pairs in speech recognition," in *Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on.* IEEE, 2015, pp. 343–347.

[5] X. Ma, D. Wang, and J. Tejedor, "Similar word model for unfrequent word enhancement in speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 10, pp. 1819–1830, 2016.