语音识别个人工作介绍

2020-7-20

语音识别系统结构

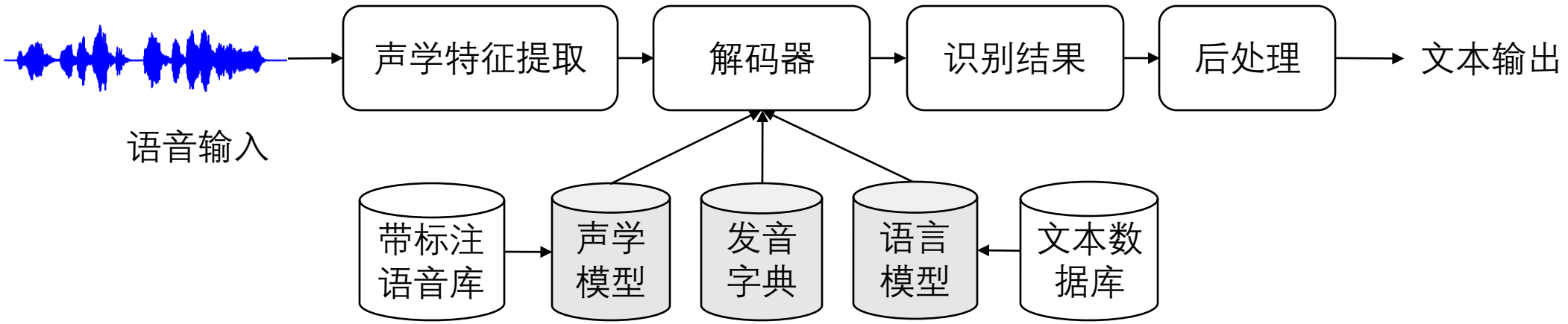

图1 传统语音识别系统结构图

目录

声学模型

- 基于X-vector的说话人适应性训练
- X-vector：说话人特征向量
  - 目标：将一段不定长语音映射成一个特征向量
  - 沿时间方向将各帧声学特征的统计量进行拼接
    - 统计量：均值、标准差

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

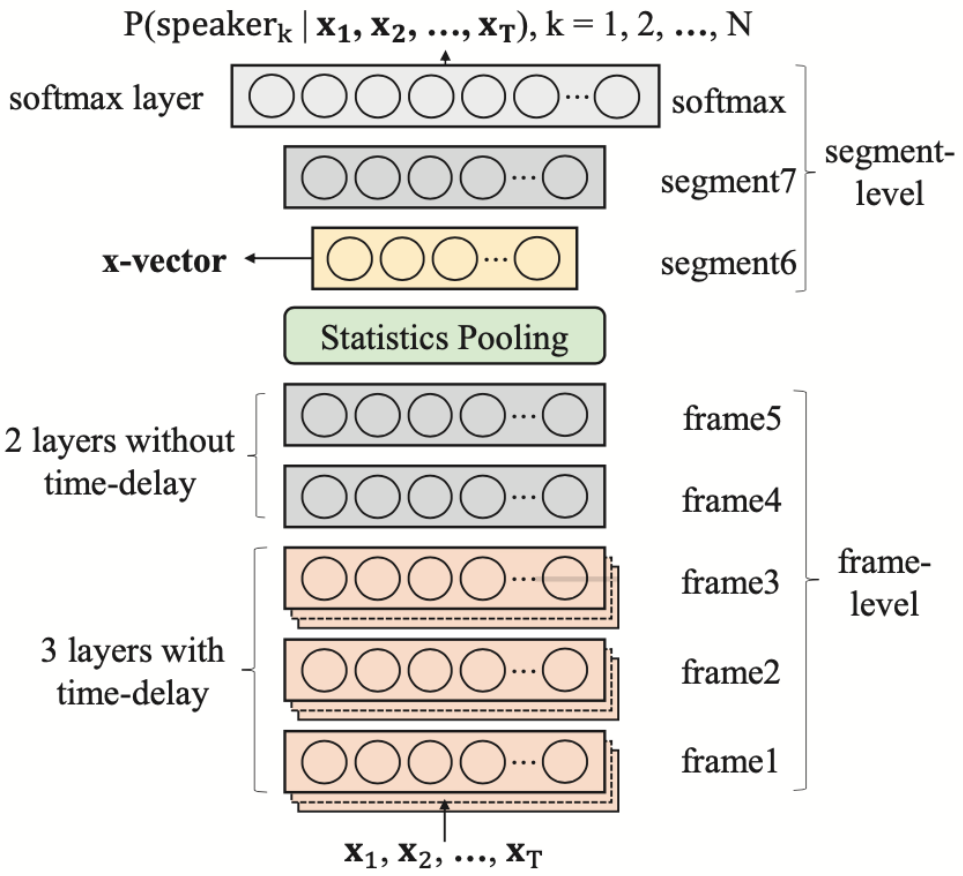

图2 x-vector网络结构

声学模型

- X-vector：说话人特征向量
  - 模型的训练目标：说话人多分类
  - 说话人识别的后端：LDA降维 + PLDA分类器
  - 提高鲁棒性：数据增强
    - MUSAN噪声库：babble music noise
    - RIRS_NOISES混响库：reverb
  - 相比于i-vector的优势
    - 在说话人识别任务中性能更优
    - 数据增强后提升效果相比于i-vector更明显
    - 适合大数据量级的说话人识别模型训练

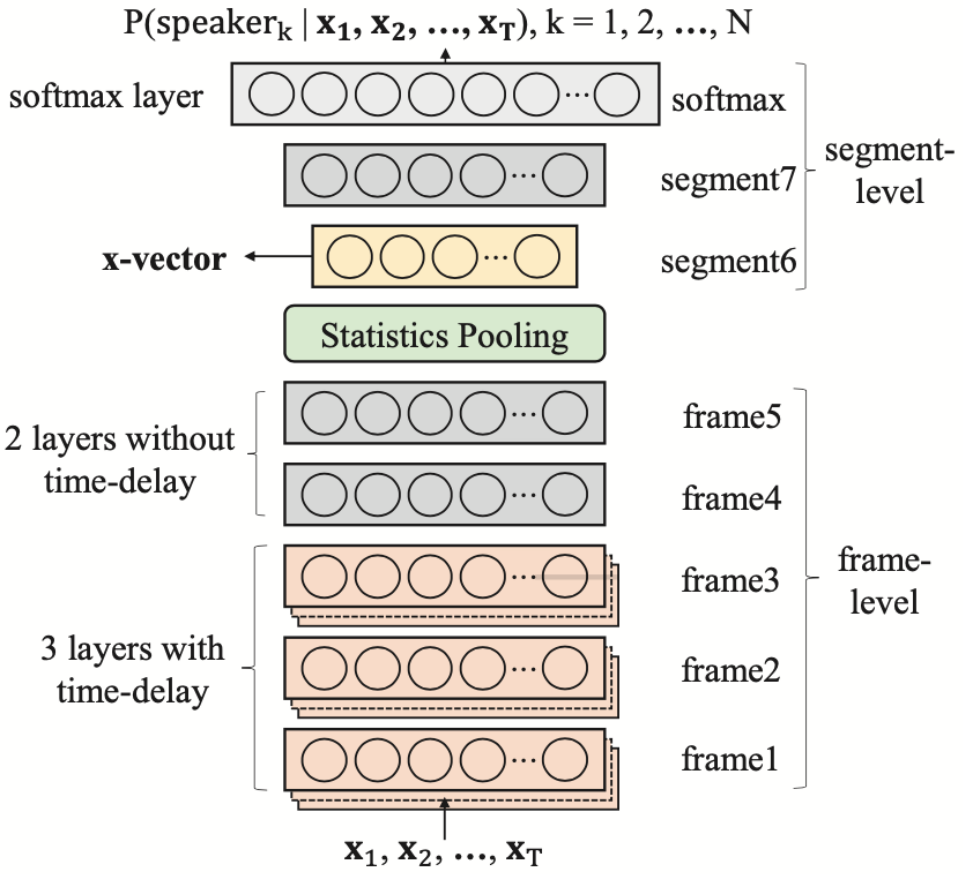

图2 x-vector网络结构

声学模型

- 基于X-vector的说话人适应性训练
  - fMLLR：说话人适应 → 说话人识别
  - i-vector：说话人识别 → 说话人适应
  - x-vector：说话人识别 → 说话人适应?
- X-vector的应用方法
  - 借鉴i-vector在说话人适应中的方法
    - 训练数据筛选：去除时长短和语音片段数少的说话人
    - 细分说话人标签，增加说话人丰富性
      - 将语音片段数多的说话人拆分成多个说话人
    - 与mfcc特征拼接作为声学模型的输入
  - 去除说话人识别中的LDA降维模块
    - 类似于Bottleneck，直接将神经网络中提取所需维度的特征

声学模型

- 基于X-vector的说话人适应性训练
  - 最终实验方案（TED_LIUM r3数据）
    - 说话人拆分
    - x-vector选用200维
    - 200维X-vector直接从DNN中提取，不用LDA
  - i-vector与x-vector在结果层面的融合
    - ROVER融合
  - 其他优化思路（未实验验证）
    - 数据增强：加噪、加混响
    - Statistics Pooling增加均值、方差之外的统计量

d	Speaker Modification	WER		WER 4-gram		WER RNNLM	
		Dev	Test	Dev	Test	Dev	Test
100	No	8.37	8.53	7.82	8.13	6.69	7.07
	Yes	8.44	8.28	7.83	7.94	6.76	6.97
200	No	8.18	8.40	7.73	7.95	6.49	6.94
	Yes	8.29	8.36	7.74	7.89	6.48	**6.78**

表1 说话人切分后的实验结果对比

d	Extraction Strategy	WER		WER 4-gram		WER RNNLM	
		Dev	Test	Dev	Test	Dev	Test
100	LDA	8.52	8.48	7.87	8.05	6.69	7.15
	no LDA	8.44	8.28	7.83	7.94	6.76	6.97
200	LDA	8.31	8.57	7.78	8.01	6.65	6.91
	no LDA	8.29	8.36	7.74	7.89	6.48	6.78

表2 是否有LDA对实验结果的影响

system	WER		WER 4-gram		WER RNNLM	
	Dev	Test	Dev	Test	Dev	Test
i-vector	7.85	8.39	7.22	7.76	6.20	6.95
x-vector	8.29	8.36	7.74	7.89	6.48	**6.78**
feature fusion	8.10	8.36	7.46	7.80	6.40	6.90
system fusion	7.70	8.28	7.19	7.71	**6.06**	**6.71**

表3 i-vector与x-vector的融合

声学模型

- 增量训练策略
  - 使用新加入的语音数据更新模型
    - 对齐模型采用已训练的chain声学模型（TDNN, TDNN-F等）
  - 在已有模型的基础上继续迭代训练
    - 使用已训练模型初始化网络参数
    - 方案一：降低所有层的学习率和迭代轮数
    - 方案二：其它网络层参数固定，更新最后的输出层
    - 方案三：靠近输出层增加随机初始化的网络层
      - 新加层和输出层较大学习率，其他层较小学习率
  - 方案一适用于模型的进一步训练，方案二、三适用于将模型迁移到新数据集上
  - 方案二、三中，方案三的效果更优
    - 参考样例：https://github.com/kaldi-asr/kaldi/tree/master/egs/rm/s5

语言模型

- 文本清洗流程
  - 以网络爬取的文本数据为例，清洗的流程为：
    - 去除特殊符号(网页标记符号、其他语种文本)，根据需要保留中英文
    - 正则化、分词等后续工作（并行处理）
    - 集外训练数据 → 数据筛选

语言模型

- 文本筛选方法
  - 标注文本的数据不足（低资源场景），用集外文本提高语言模型效果
  - 筛选任务的目标：从集外文本中筛选出与训练标注文本相近的文本数据
  - 筛选数据的用途：训练数据扩充、发音词典扩充（结合g2p）
  - 文本相似度 / 相关性分析
    - 困惑度排序
    - 相对交叉熵排序
    - TF-IDF向量相似度
    - Doc2vec向量相似度

语言模型

- 文本筛选方法：困惑度排序
  - 使用集内数据训练的语言模型，对每个集外文本句子求出困惑度（Perplexity，PPL）

$$\mathrm{PPL(W)} = \mathrm{P}(w_1 w_2 w_3 \cdots w_N)^{-\frac{1}{N}}$$

$$\mathrm{PPL(W)} = \left[\prod_{i=1}^{N} P(w_i | w_1 w_2 \cdots w_{i-1})\right]^{-\frac{1}{N}}$$

2gram：$\mathrm{PPL(W)} = \left[\prod_{i=1}^{N} P(w_i | w_{i-1})\right]^{-\frac{1}{N}}$

  - 按照困惑度对集外文本排序，选出困惑度低于某一阈值的句子

语言模型

- 文本筛选方法：相对交叉熵
  - 交叉熵：$$H(P_{LM}) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{LM}(w_i|w_1,\cdots,w_{i-1})$$
  - 集内文本训练语言模型A，集外文本训练语言模型B
  - 对于集外每一句文本
    - 分别使用语言模型A、B计算交叉熵，作差得到相对交叉熵
  - 按照相对交叉熵排序

语言模型

- 文本筛选方法：TF-IDF

$$\mathrm{TF} = \frac{C(W|D)}{C(D)}$$

$$\mathrm{IDF} = \log \frac{C(M)}{C(M|W)+1}$$

  - 某词的TF-IDF值越大，该词对文章重要性越高，对于文档越关键(关键词)
  - 每篇文章各取出若干个关键词，合并成一个集合
  - 对于每个文档，计算关键词集合中每个词的TF-IDF值，合为一个向量
    - 计算文档之间向量的相似度，值越大表示越相似。

语言模型

- 文本筛选方法：doc2vec
  - Distributed Memory Model
  - 将文档投影为向量，参与到训练中
    - 文档向量在每个CBOW预测中都起到了作用
  - 文本筛选流程
    - 集内文本对应一个文档向量
    - 将集外文本的文档向量与集内文档向量求相似度
    - 根据相似度进行排序

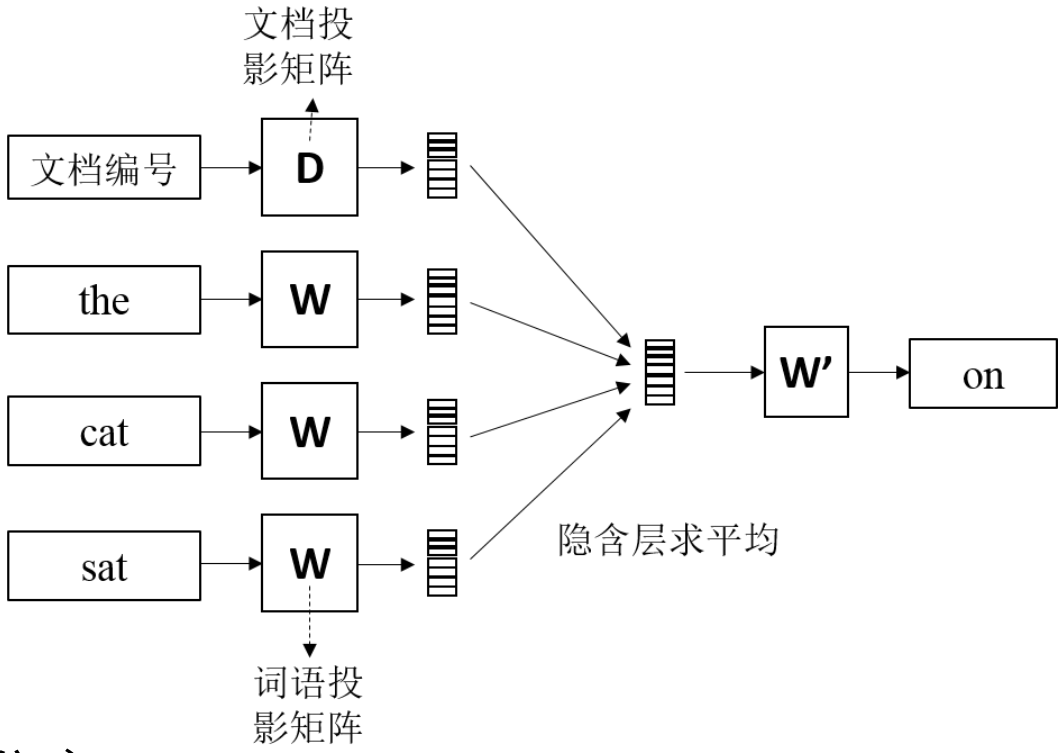

图3 doc2vec结构示意图

语言模型

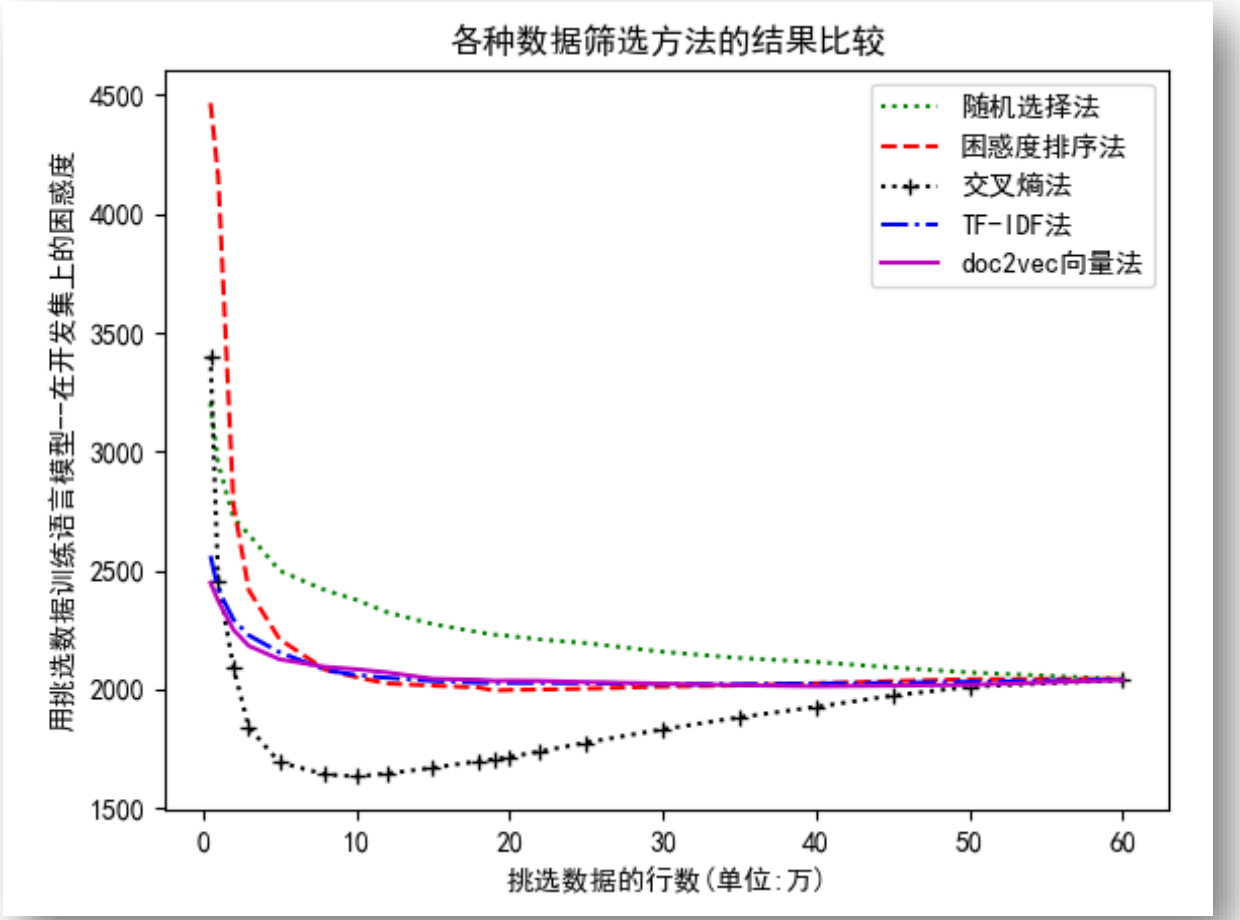

图4 不同数据筛选方法对比

语言模型

- 如何使用筛选后的数据?

方案一：与集内文本数据融合，训练语言模型
方案二：分别训练语言模型，采用线性插值方法，插值系数使得在开发集上PPL最低

建模方法	最优语言模型	语言模型困惑度
原始训练数据	ME3	429.30
相对交叉熵法筛选的数据	KN4	1632.75
两部分数据合并	KN4	598.42
两部分数据分别训练语言模型后插值	ME3	409.11

表4 筛选数据的不同使用策略对比

数据来源	原始训练数据	困惑度法	交叉熵法	TF-IDF法	Doc2vec法
数据行数	3.77万	20万	10万	30万	40万
语言模型	ME3	KN4	KN4	KN4	KN4
困惑度	429.30(WER = 47.7%)	1994.43	1632.75	2021.46	2012.21
插值系数	0.825	0.054	0.017	0.042	0.062
模型融合结果	PPL = 387.32, WER = 46.8 %				

表5 数据筛选方法实验对比

语言模型

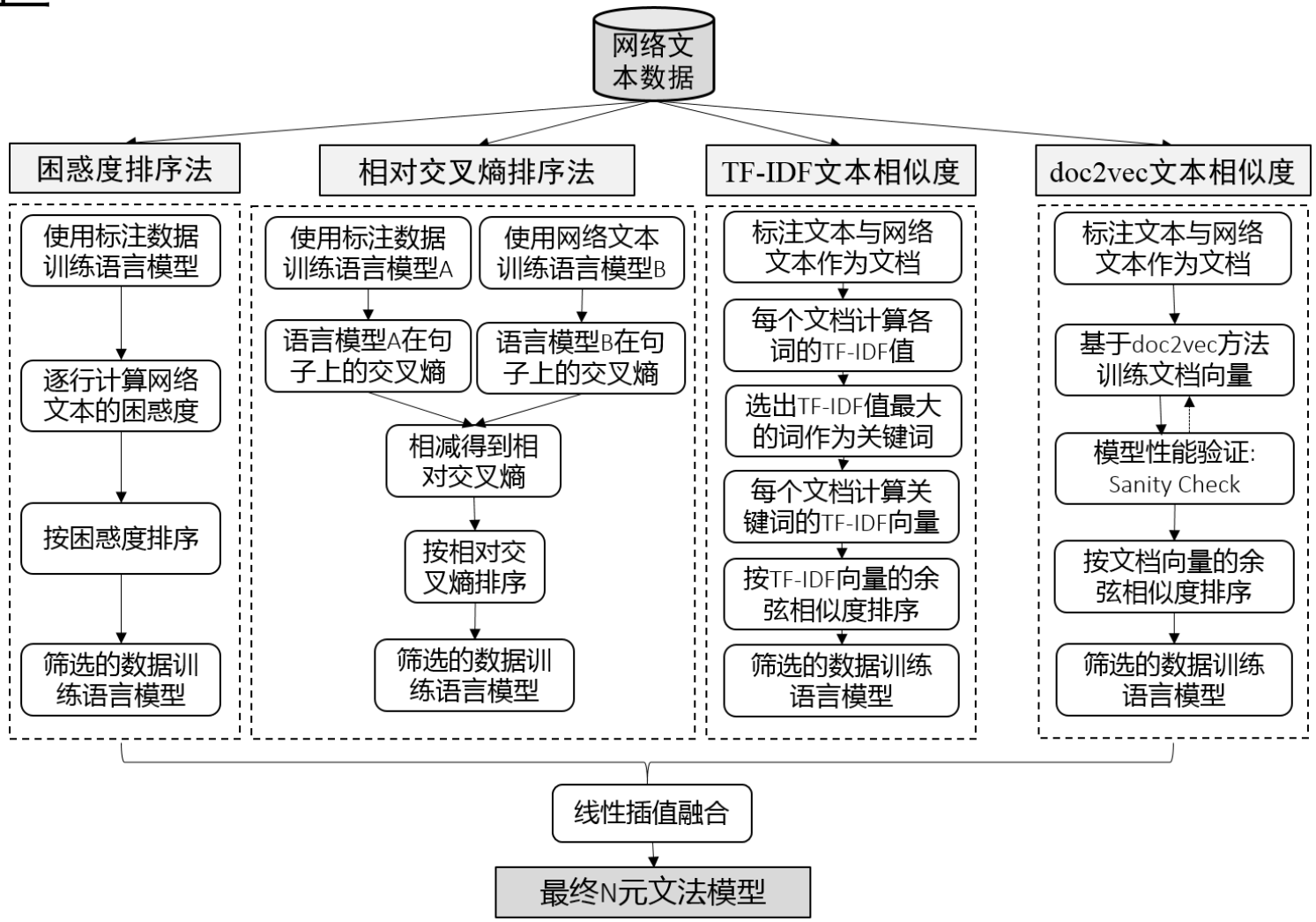

图5 完整的语言模型训练数据筛选流程

大规模语音识别系统训练

- 声学模型
  - 三万小时语音标注数据（不同批次数据）
  - 使用全部数据重新训练
    - 合并不同来源数据的对齐结果：steps/combine_lat_dirs.sh
    - 数据增强无需重新对齐：steps/copy_lat_dirs.sh

- 语言模型
  - 将场景语言模型与通用语言模型插值
  - 通用语言模型训练（400GB文本）
    - 并行化进行正则化和分词
    - SRILM工具
      - 并行词频统计： make-batch-counts
      - 词频融合生成语言模型：make-big-lm
  - 根据需要对语言模型进行剪枝

语音识别后处理

- 标点符号恢复
  - 双向LSTM结构

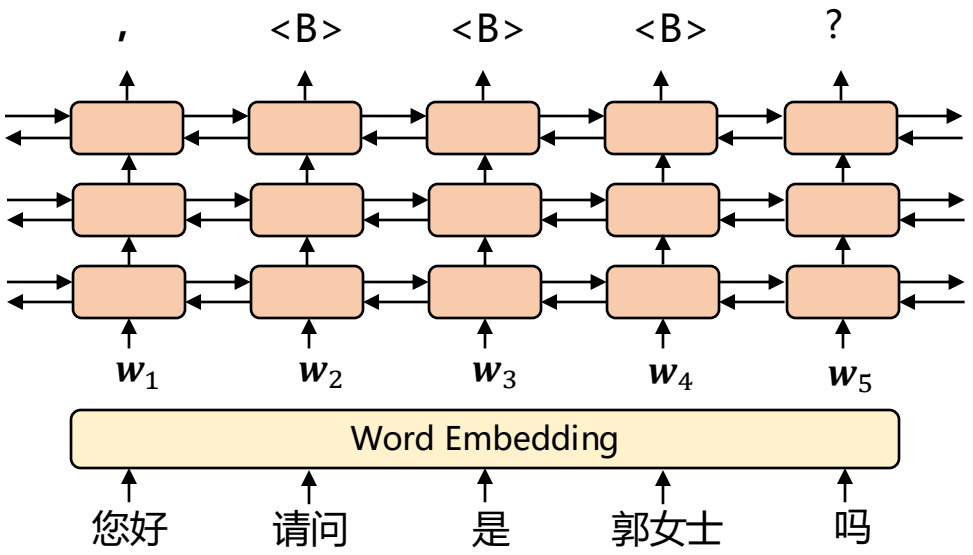

图6 LSTM标点符号预测模型

语音识别后处理

- 标点符号恢复
  - 使用多层一维CNN替代LSTM结构

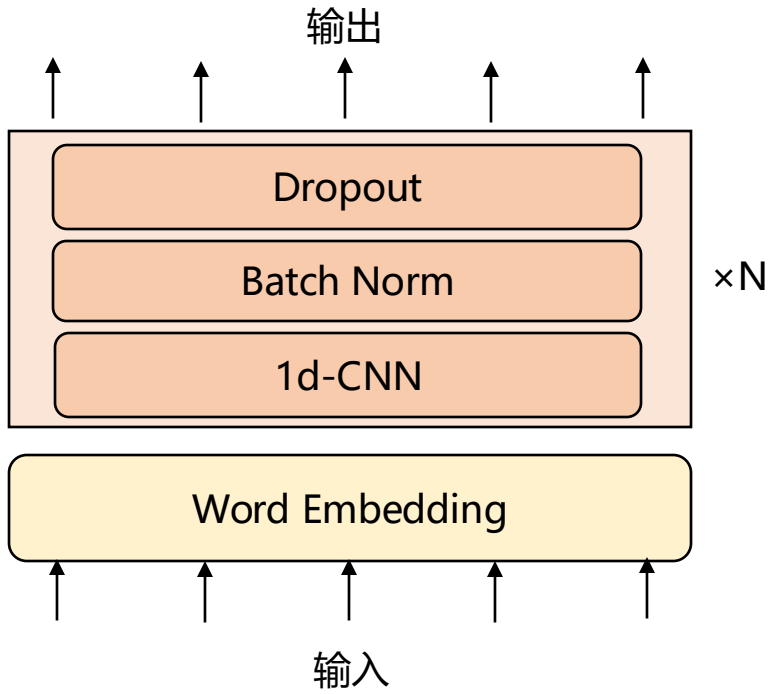

图7 CNN标点符号预测模型

语音识别后处理

- 标点符号恢复
  - 输入特征加入词之间的停顿时间信息

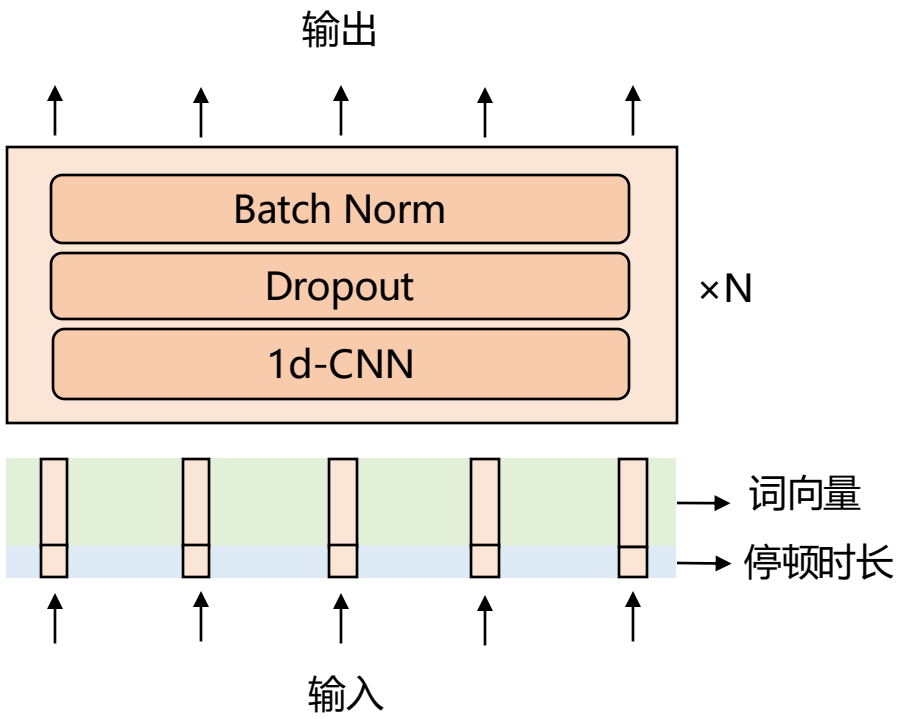

图8 输入增加停顿时长信息

语音识别后处理

- 实验结果
  - 网络搜集的中文NLP数据集

模型	层数	逗号，F值	句号。F值	问号?F值	Macro F1值
LSTM	2	0.64	0.75	0.54	0.64
CNN	1	0.57	0.66	0.46	0.56
	2	0.62	0.70	0.52	0.61
	4	0.64	0.73	0.53	0.63

表6 LSTM和CNN实验结果对比

方法	是否加入停顿信息	逗号，F值	句号。F值	问号?F值	Macro-F
LSTM	否	0.64	0.75	0.54	0.643
	是	0.75	0.88	0.63	**0.753**

表7 增加停顿时长前后实验结果对比

语音识别后处理

- 文本检错纠错
  - 第一步：基于语言模型分数的错误检测
    - 对语言模型分数序列，用绝对中位差进行异常检测
    - 绝对中位差：分数序列与中位数作差，取差的中位数
    - 使用不同尺度的窗(2, 3, 4)扫描，将错误区间进行合并

$$P(|X-\mu| \le MAD) = P\left(\left|\frac{X-\mu}{\sigma}\right| \le \frac{MAD}{\sigma}\right) = P\left(|Z| \le \frac{MAD}{\sigma}\right) = \frac{1}{2}$$

$$\frac{MAD}{\sigma} = \Phi^{-1}\left(\frac{3}{4}\right) \approx 0.6745$$

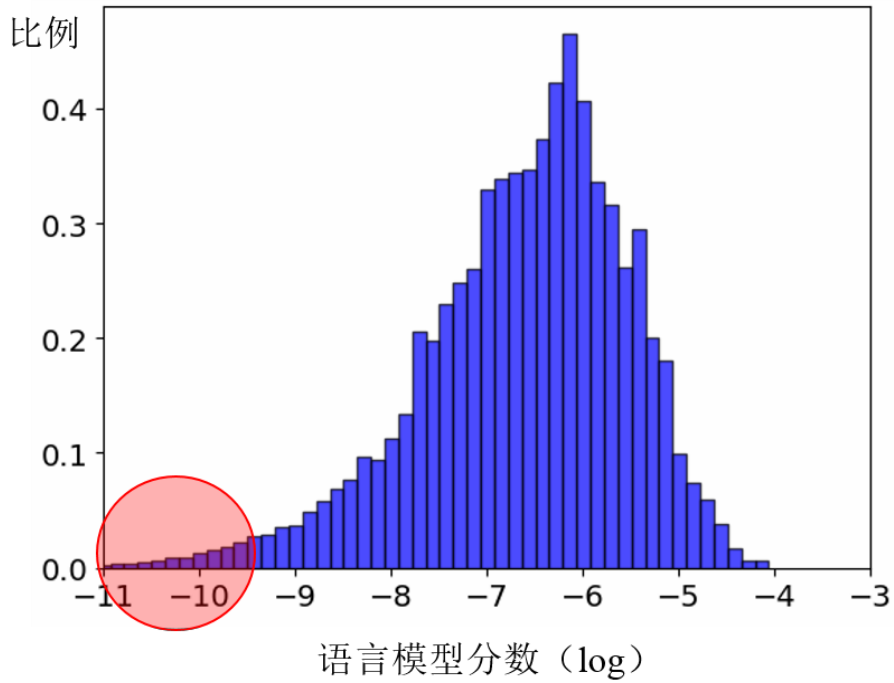

图9 语言模型分数分布

语音识别后处理

• 文本检错纠错

- 候选词生成
  - 根据发音音素序列，筛选出音近字词集合
  - 易错词统计，整理成易错词对集合

- 文本纠错
  - 若包含易错词，替换后打分判断是否替代
  - 选择分数最高的候选词作为结果
  - 通过设定阈值调整纠正力度

- N-gram：多组语言模型提高稳健性
  - 字级别 / 词级别
  - 2-gram 3-gram 4-gram

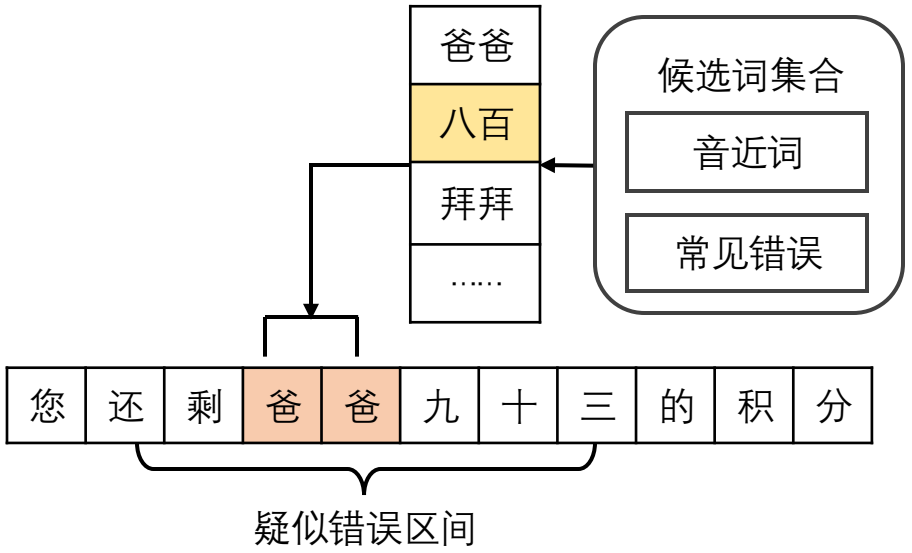

图10 识别结果纠错示意图

纠错前：	我是保险公司的**带你**小钱友印象吗
纠错后：	我是保险公司的**代理**小钱有印象吗
纠错前：	目前给您订的机票是**男孩**公司明天的航班
纠错后：	目前给您订的机票是**南航**公司明天的航班
纠错前：	**把**电话给您是因为您可以参加积分兑一**反**一的活动
纠错后：	**打**电话给您是因为您可以参加积分兑一**返**一的活动

语音识别后处理

- 文本检错纠错

	A组 (0–20%) (错误率较低)		B组 (20–50%) (错误率中等)		C组(50–100%) (错误率较高)	
	纠错前	纠错后	纠错前	纠错后	纠错前	纠错后
替换	6.1%	5.8%	18.1%	19.3%	43.2%	45.8%
插入	3.0%	3.0%	7.1%	7.2%	4.2%	4.4%
删除	2.2%	2.2%	5.4%	5.3%	12.2%	12.2%
合计	11.3%	11.0%	30.6%	31.8%	59.6%	62.4%
相对变化	↓ 2.7%		↑ 3.9%		↑ 4.7%	

表8 识别结果检错纠错在不同错误率下的对比

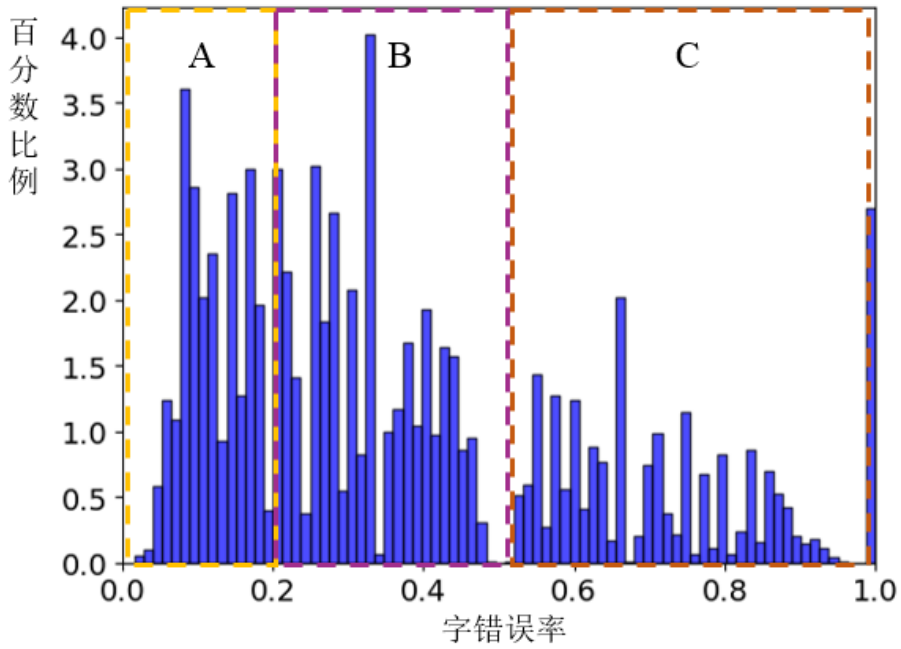

图11 测试集字错误率分布情况

- 结论一：适用于纠正替换类型的错误
- 结论二：适用于错误率较低的识别结果

语音识别后处理

- 文本逆正则化
  - 关键功能：根据数字读法转换成阿拉伯数字
  - 基于特殊规则：
    - 电话号码：连续的单个数字
    - 百分数/单位：符号对应
    - 优先级：单位符号 > 百分数 > 普通数字

转换前：请问一下您是**幺五三六四零幺三零七五**的机主吗 转换后：请问一下您是**15364013075**的机主吗
转换前：截止到**十二**月**六**日您名下还有**六千二百三十三**的积分 转换后：截止到**12**月**6**日您名下还有**6233**的积分
转换前：速度提升了**百分之二十**达到**三百六十一点六千米每小时** 转换后：速度提升了**20%**达到**361.6km/h**

参考文献

- Snyder, David, et al. "X-vectors: Robust dnn embeddings for speaker recognition." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.
- Ferras, Marc, et al. "Constrained MLLR for speaker recognition." 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. Vol. 4. IEEE, 2007
- Najim Dehak, Patrick J Kenny, Re´da Dehak, Pierre Dumouchel, and Pierre Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2010.
- Vijayaditya Peddinti, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Reverberation robust acoustic modeling using i-vectors with time delay neural networks,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.
- George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, 2013, pp. 55–59.
- Miao, Yajie, Hao Zhang, and Florian Metze. "Speaker adaptive training of deep neural network acoustic models using i-vectors." IEEE/ACM Transactions on Audio, Speech, and Language Processing 23.11 (2015): 1938-1949.

参考文献

- Snyder, David, Guoguo Chen, and Daniel Povey. "Musan: A music, speech, and noise corpus." arXiv preprint arXiv:1510.08484 (2015)
- Rousseau, Anthony. "Xenc: An open-source tool for data selection in natural language processing." The Prague Bulletin of Mathematical Linguistics 100.1 (2013): 73-82.
- Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. 2014.
- Xu, Kaituo, Lei Xie, and Kaisheng Yao. "Investigating LSTM for punctuation prediction." 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP). IEEE, 2016.
- Żelasko, Piotr, et al. "Punctuation prediction model for conversational speech." arXiv preprint arXiv:1807.00543 (2018).
- Tilk, Ottokar, and Tanel Alumäe. "LSTM for punctuation restoration in speech transcripts." Sixteenth annual conference of the international speech communication association. 2015.
- Errattahi, Rahhal, Asmaa El Hannani, and Hassan Ouahmane. "Automatic speech recognition errors detection and correction: A review." Procedia Computer Science 128 (2018): 32-37.