Report

Overlapped Speech Recognition

**Zhaohui Yin**

**Tue, Nov 10, 2020**

* 重叠语音识别研究计划（11月10日至12月10日）

1）AED-based 单人语音识别的理论基础及模型理解（End-to-End）；（已完成）

✅Kanda (2020)：初步了解现在重叠语音识别的研究现状：单人说话系统的错误率为5.5%，两人说话系统的错误率却为84.7%。

✅Chorowski (2014): 深入了解AED-based单人语音识别的模型构建和理论基础。

✅Gelabert (2018): 了解语音识别模型的发展进程（从HMM-based模型的End-to-end语音识别模型，从单人到多人），深入了解其中的技术细节，为后面复现做准备。

2）end-to-end 单人语音识别（seq2seq）的模型复现；

3）通过文献把握现在的研究进程，明确重叠语音识别的研究问题；

1. 重叠音识别研究

1.1 文献阅读

Kanda (2020)：Serialized Output Training for End-to-End Overlapped Speech Recognition. INTERSPEECH 2020.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 痛点 | 1. 多人演讲（multi-speaker speech）的自动化语音识别（ASR）目前是一个十分困难的问题，尤其是单声道录音（monaural recording）中的多种发音明显重叠时。 | | | |
| 2. 目前研究进度：单一说话者ASR系统的错误率（Word Error Rate WER）为5.5%，两个说话者ASR系统的WER为84.7%。  [11] N. Kanda, S. Horiguchi, R. Takashima, Y. Fujita, K. Nagamatsu, and S. Watanabe, “Auxiliary interference speaker loss for target- speaker speech recognition,” in Proc. Interspeech, 2019, pp. 236– 240. | | | |
| 研究进展 | DNN-based ASR\* | 描述 | 同时训练两个ASR模型，较高的瞬时能量和较低的瞬时能量。 | |
| 局限性 | 该模型仅能应用于两个说话者的会话场景 | |
| \* [12] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, “Deep neural networks for single-channel multi-talker speech recognition,” IEEE/ACM Trans. on TASLP, vol. 23, no. 10, pp. 1670–1679, 2015. | | |
| PIT：Permutation Invariant training  (置换不变训练) | 描述 | 对不同的说话者分别设置输出层，训练过程中考虑所有可能的说话者排列组合。 | |
| 局限性 | i) 会存在最大说话者数目的限制；----> 模型需要先验知识  ii) 没有考虑不同说话者语音之间的依赖性； ----> 输出层之间是相互独立的 ---> 不同输出层可能会重复生成  iii）计算复杂度高，先前研究主要是基于两个说话人场景\*。 | |
| 语音分割：[13] D.Yu,M.Kolbæk,Z.-H.Tan,andJ.Jensen,“Permutationinvari- ant training of deep models for speaker-independent multi-talker speech separation,” in Proc. ICASSP. IEEE, 2017, pp. 241–245.  多人识别：[14] D. Yu, X. Chang, and Y. Qian, “Recognizing multi-talker speech with permutation invariant training,” Proc. Interspeech 2017, pp. 2456–2460, 2017. | | |
| 相似研究\*  【弄清楚】 | 1）The joint speech recognition and diarization with a recurrent neural network transduer (RNN-T)\*  （递归神经网络变换器的联合语音识别和差分，利用交错的标签来对不同说话者进行转录）    [22] L. El Shafey, H. Soltau, and I. Shafran, “Joint speech recogni- tion and speaker diarization via sequence transduction,” in Proc. Interspeech, 2019, pp. 396–400. | | | |
| 2）AED-based multi-lingual ASR for mixed-language speech\*（基于AED\* 的多语言ASR，使用语言标签来作为额外的输出标志，从而实现利用简单的模型进行混合语言会议的翻译）  [24]H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, “An end-to-end language-tracking speech recognizer for mixed- language speech,” in Proc. ICASSP, 2018, pp. 4919–4923. | | | |
| 不理解的地方 | \* monotonic alignment constraint  （单调对齐约束？）  ✅ 通俗理解（by龙哥）：将每个字所对应的帧给对应起来，实际操作中会先平均分配帧数进行第一轮训练，训练过程中需要进行消音处理（关键词：SIL）[还需跟进查询]（已跟进解决） | | |  |
| \* End-to-End（端到端？）  ✅ 传统的语音识别是需要使用字典的，字典中会对标记的每个字进行音素的拆分，然后利用GMM-HMM模型进行识别。目前新兴的端到端，则是省去这些过程，直接利用神经网络来完成输入端到输出端，关键词：CTC [WED NOV 11]  ✅ 深入理解：  语音识别的技术原理是什么？ - Dr.Wu的回答 - 知乎  https://www.zhihu.com/question/20398418/answer/1304118597 | | | |
| \* CE loss（如何计算？）（已理解） ✅ 就是和神经网络中的交叉墒类似。 | | | |
| 创新点 | 1. 没有说话者最大数目的限制； | | | |
| 2. 可以对不同说话者的输出之间的依赖性进行建模，有效避免了重复生成； | | | |
| 3. 降低了计算复杂度，从 S)，其中*S*是指说话人数目； | | | |
| 技术手段 | 技术手段需要详细的个人理解，见Section 1.2 | | | |
| 评估结果 | WER | | | |

1.2 个人总结

作为重叠音识别的基础，进行单人场景语音识别的工作首当其冲。目前端到端语音识别的研究更是前沿热点。利用传统的HMM-based模型（Fig1）进行语音识别时，声学模型，语言模型都需要单独的组成和培训。其中声学模型采用声学特征预测音素的集合，发音模型利用发音词典将音素拼接成词语，语言模型为单词序列分配概率。与联合训练所有模块相比，单独训练各个模块部署起来更为复杂且次优，这也是端到端语音识别如此热门的原因。端到端语音识别目前有两种主要方法，其一是Conectionist Temporal Classification（CTC），其二是Sequence to Senquence（seq2seq）

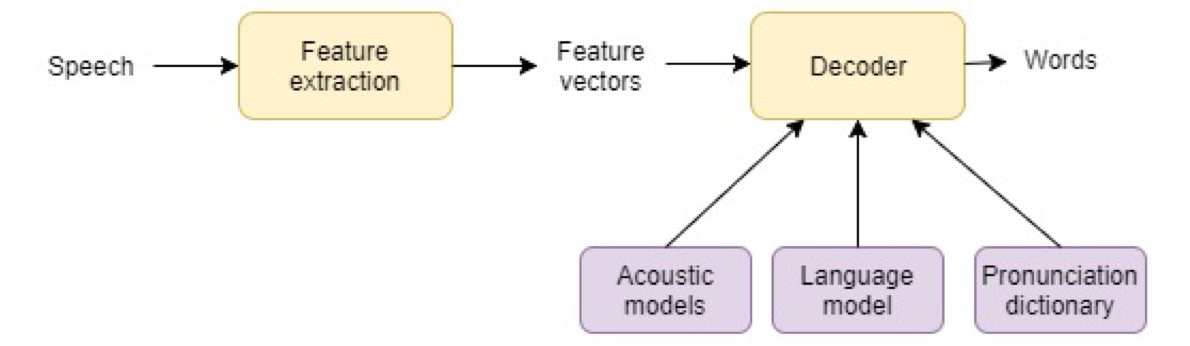


Fig1 HMM进行语音识别的流程图

|  |  |  |
| --- | --- | --- |
| 基于注意力模型的单人说话自动语音识别算法 | 存在两个问题：  1）公式2中中间语义和注意力权重的计算需要用到么？  分析：注意力权重是根据计算而来，其中对齐模型最常用的是dot product乘积矩阵，仅跟target端的隐层节点状态和source端的隐层节点状态相关。未查询到此处需要。十分不解  2）公式3中的是否表达有误，我认为此处应该是利用对target端的隐层节点状态进行更新。[涉及到对模型的理解] (已解决)  ✅Chorowski (2014) : End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. (module)  ✅Gelabert (2018): Exploring Automatic Speech Recognition with TensorFlow. (module) |  |

模型的技术性细节仅看模型图是无法知道的，因此需要复现。[这个不是重点，重点是开始着手看重叠音相关研究的文章]

✅ 目的：复现，熟悉技术上的细节，对结果有一个更为清晰的认知。

✅ 参考文献：

[1] Vincent Renkens. Nabu. https://github.com/vrenkens/nabu, 2017.