

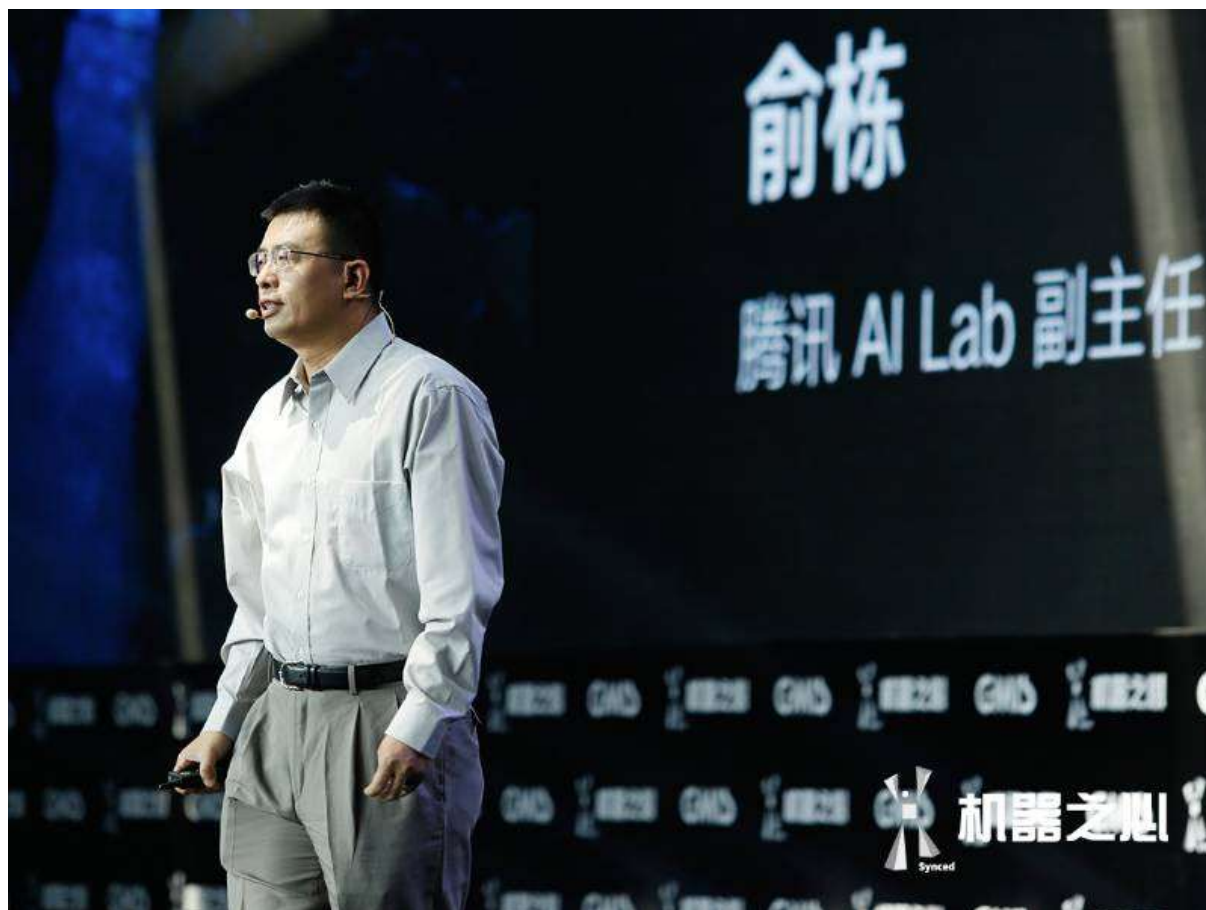
# GMIS 2017 | 腾讯AI Lab副主任俞栋：语音识别研究的四大前沿方向

原创 2017-06-02 机器之心 机器之心

机器之心整理

演讲者：俞栋

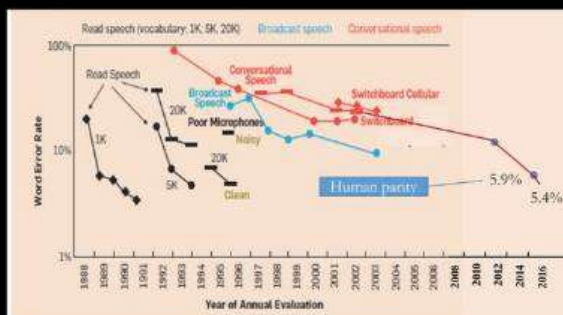
5月27-28日，机器之心在北京顺利主办了第一届全球机器智能峰会（GMIS 2017），来自美国、加拿大、欧洲、香港及国内的众多顶级专家分享了精彩的主题演讲。在这篇文章中，机器之心整理了腾讯AI Lab副主任、西雅图人工智能研究室负责人俞栋在大会第一天发表了主题为《语音识别领域的前沿研究》的演讲，探讨分享了语音识别领域的4个前沿问题。



俞栋是语音识别和深度学习领域的著名专家。他于1998年加入微软公司，此前任微软研究院首席研究员，兼任浙江大学兼职教授和中科大客座教授。迄今为止，他已经出版了两本专著，发表了160多篇论文，是60余项专利的发明人及深度学习开源软件CNTK的发起人和主要作者之一。俞栋曾获2013年IEEE信号处理协会最佳论文奖。现担任IEEE语音语言处理专业委员会委员，之前他也曾担任IEEE/ACM音频、语音及语言处理汇刊、IEEE信号处理杂志等期刊的编委。

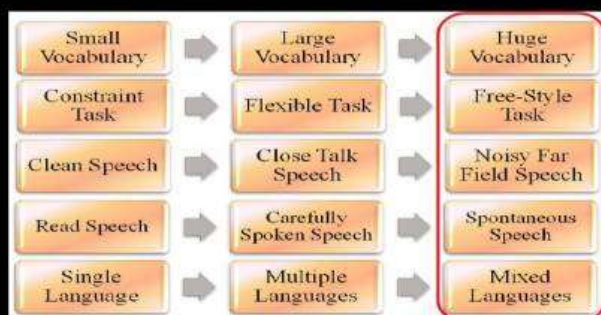
以下是俞栋演讲的主要内容：

## Progresses in Speech Recognition



- Speech recognition is a field with decades of research
- Tackled the easier/more constrained tasks one by one along the way
- Super-human performance was claimed even on Switchboard

## Research Frontier is Once Again Shifted



- Less constrained (vocabulary, speaking style, accent, environment)
- More complicated
- Many problems hidden by close-talk microphones now surface in far-field scenarios

大家好，我是俞栋，现在腾讯 AI Lab，是西雅图研究室的负责人，我的主要的研究方向是语音识别，所以今天我在这里也给大家介绍一下最近的一些语音识别方向的研究前沿。

大家都知道语音识别领域有着悠久的历史，在过去的几十年里面，研究人员从最简单的非常小词汇量的阅读式的语音识别问题开始，逐渐转向越来越复杂的问题。现在即便是在以前认为非常难的自由对话形式的语音识别问题，机器也已经能够达到甚至超过人的识别水准。不过我们要看到，虽然我们取得了这些进展，但是离真正非常自由的人机交流还有一定的距离。这也是为什么我们现在语音识别的研究前沿又往前推进了一步，现在我们研究的问题越来越多地是不对环境、说话的风格、口语做任何限定（不像以前有非常多的限制）。而这些非限定的环境，就使得语音识别难度有了大幅度的增加。尤其在最近的几年里面我们发现在真实的应用场景里，很少有人会愿意戴着麦克风，所以现在研究的前沿就从近场麦克风向远场麦克风改变。

从近场到远场麦克风的改变有一个很重要的区别，即远场的情况下，当人的声音传达到麦克风的时候，声音的能量衰减得很厉害。所以近场麦克风很难见到的一些困难，在远场麦克风里面就变得非常重要。最著名的就是鸡尾酒会问题，本文稍后会对其做一个详细的介绍。如果这些远场问题不解决的话，在很多的场合，用户仍然会觉得语音识别并不是很方便。

## Recent Research Focuses

- Effective sequence-to-sequence direct mapping
- Cocktail party problem
- Models that continuously predict and adapt
- Frontend-backend joint optimization

所以今天在这样的背景下，我介绍一下最近在语音识别当中的一些前沿的研究方向，主要有四个：

研究方向一：更有效的序列到序列直接转换模型

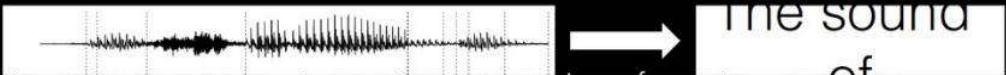
研究方向二：鸡尾酒会问题

研究方向三：持续预测与适应的模型

研究方向四：前端与后端联合优化

**研究方向一：更有效的序列到序列直接转换模型**

### Effective Sequence-to-Sequence Direct Mapping



- Acoustic sequence to word sequence transformation problem
- Direct mapping bypasses manually designed components to achieve better performance if model is right, training is effective, and data are sufficient
- Direct mapping significantly reduces training pipeline if trained from scratch

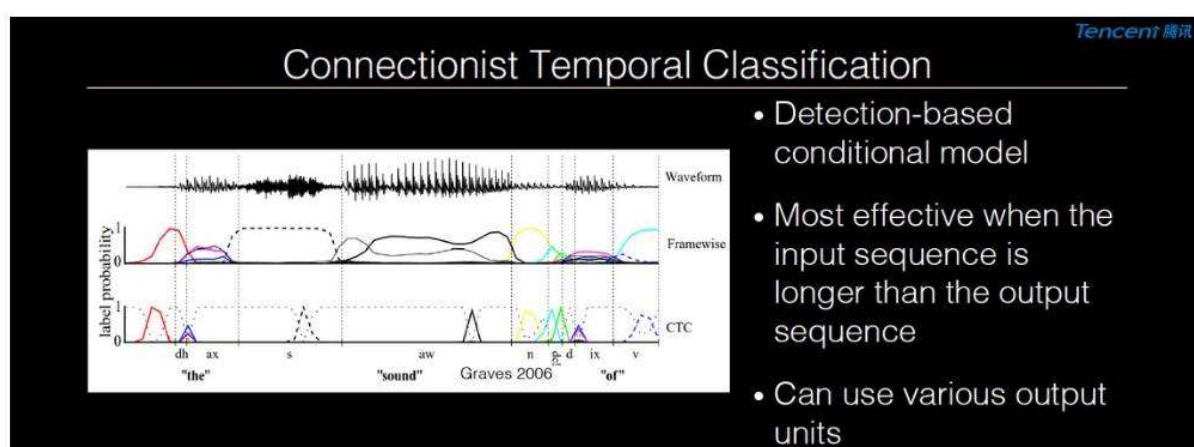
如果我们仔细想想语音识别这个问题的话，大家都会看到，语音识别其实就是一个从语音信号序列转化为文字或者词序列的问题。这也就是为什么很多研究人员都一直认为要解决这个问题其实只要找到一个非常有效的，从一个序列到另外一个序列转换的模型就可以了。

在以前的所有的研究里面，绝大部分的工作都是研究人员通过对问题做一些假设，然后根据这个

假设从语音信号序列到词信号之间，生成若干个组件，然后通过逐步地转换，最后转换成词的序列。有许多假设在某些特定场合中是合理的，但是在很多真实的场景下还是有问题的。那么直接转换这样序列模型的想法就是，如果我们能够把这些可能有问题的假设去掉，然后通过数据驱动让模型自己学习，就有可能找到一个更好的方法，使得这个序列的转换更准确。这样做还有另外一个好处，因为所有的这些人工的 component 都可以去掉了，所以整个的训练流程也就可以缩短。

序列到序列直接转换、直接映射这样的研究目前来讲主要有两个方向：

### 方向一：CTC (Connectionist Temporal Classification) 模型



如上图所示，方向一是 CTC (Connectionist Temporal Classification) 模型，从上图中最下面一行可以看到，在 CTC 模型里面，系统会一直保留一个内部状态，当这个内部的状态提供足够的信息可以做某一个决定的时候，它就会生成一个尖峰 (spike)。其表明到某个位置的时候可以非常确定地推断到底听到了哪个字或者哪个词。而在没有听到足够的信息的时候，只会产生空信号以表明还不能有足够的信息来判断是不是听到了某一个字或者词。这样的模型在语音识别问题上是非常合适的模型，因为它要求输出序列的长度比输入序列的长度要短很多。

CTC 模型还有一个优势，即传统的深度神经网络与混合模型一般来说建模单元非常小，但是在 CTC 模型可以相对自由地选择建模单元，而且在某些场景下建模单元越长、越大，识别效果就越好。

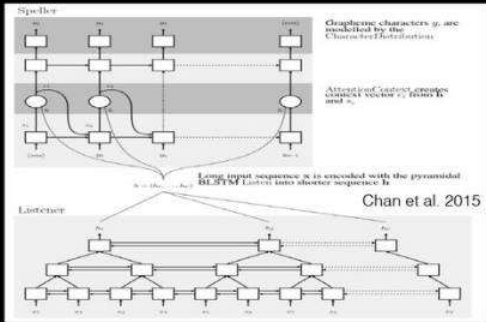
最近 Google 有一项研究，他们在 YouTube 上采用几十万小时甚至上百万小时的数据量训练 CTC 的模型，发现可以不用再依赖额外的语言模型就能够做到超过传统模型的识别率。CTC 模型相对来说比传统的模型仍会更难训练，因为其训练稳定性还不是很好。



## 方向二：带有注意力机制的序列到序列转换模型

Tencent 腾讯

### Sequence-to-Sequence Transformation with Attention



Grapheme characters  $g_i$  are modeled by the Character Embeddings.

Attention of context creates context vector  $c_i$  from  $h$  and  $x_i$ .

Long input sequence  $x$  is encoded with the pyramidal RNN into shorter sequence  $h$ .

Chan et al. 2015

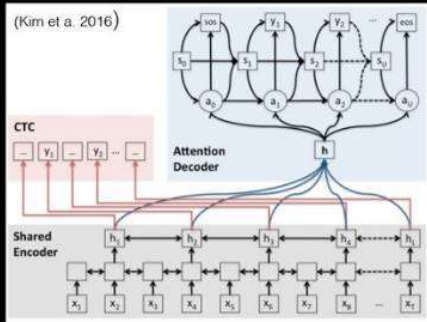
- Encoder-decoder framework
- Soft-alignment for better generation through attention mechanism
- Training is less stable compared to CTC
- Need to handle the training and decoding mismatch problem

第二个比较有潜力的方向是带有注意力机制的序列到序列转换模型（Sequence-to-Sequence Transformation with Attention）。这个模型基本的想法是首先把输入的序列、语音信号序列，转换成一个中间层的序列表达，然后基于中间层的序列表达提供足够的信息给一个专门的、基于递归神经网络的生成模型，并每次生成一个字、一个词或者一个音符。现在这个方法在机器翻译里面成为了主流方案，但是在语音识别里面它还是一个非常不成熟的技术。它有如下几个问题：

- 问题 1：训练和识别过程有很大的不匹配性，在训练过程中所依赖的信息是准确的、已知的，但是在识别过程中，信息却是估算出来的，是不准确的。所以一旦错误产生以后，这个错误就可能会累加到下一个字或词的生成，所以该方法比较适合只是一个短语的识别，对长的句子效果比较差。
- 问题 2：该模型和语音识别本身也有不匹配性，这个不匹配是其在进行注意力机制时产生的，因为注意力可以在不同的位置上移动，但是对于语音识别，下一个词的 attention 肯定是在前一个词的 attention 的后面，其有一个持续固定的约束，这个约束在目前带注意力机制的序列到序列模型里是不存在的，所以这个模型目前在做语音识别的时候效果非常不稳定。

Tencent 腾讯

### Better When Combined



(Kim et al. 2016)

CTC

Shared Encoder

Attention Decoder

- Encoder shared by CTC and attention models
- CTC's left to right constraint helps to learn good representation
- Significantly speed up learning alignments between output units and acoustic frames with multi-task learning

如何解决这样的问题而得到更好的结果呢？目前最佳的解决方案就是把 CTC 模型跟 Attention 模型联合在一起，最基本的想法是因为 CTC 有持续信息，其词的生成是根据后面好几帧的语音信号信息而得出，因此它会帮助 Attention 模型生成更好的 embedding space 表达。结合这两个方法所最终产生的结果既比 CTC 模型训练的好，也比 Attention 模型训练的好，所以这就变成了一个 1+1 大于 2 的结果。

Tencent 腾讯

## Effective Sequence-to-Sequence Direct Mapping

- Stability: Can we improve upon both CTC and attention models with a better formulation (model and/or training criterion)?
- Language model: Can we find a better integrated model so that decoding can be simple and the LM can be optimized jointly with AM when audio signals are available and separately when only text data is available?
- Data requirement: Can the components be transferred at different scales to work with different data sizes?

我们稍后会看到，即便把两种成本函数和模型结构联合在一起，它的效果与传统的混合模型相比并没有太大的长进。所以我们仍然需要解决一些问题。

- 问题一：在这样的架构下面，有没有更好的模型结构或训练准则，能够比现有的 CTC 或者 Attention 模型更好。
- 问题二：我们看到 YouTube 用 CTC 模型训练的时候，它的效果比用语言模型的传统方法更好，很大的原因就在于它的训练集有很多的训练语料，因此我们可以在里面训练非常好的语言模型，所以语言模型和声学模型是紧密结合在一起的。那么当我们没有这么多的数据时，有没有办法也建造一个结构，使得这个语言模型和声学模型紧密结合在一起。但是当训练数据不够多的时候，如果有足够的文本数据，我们也可以用它来加强语言模型的训练，使两个部分能够相辅相成。
- 问题三：到底有没有办法结合各种语料的数据，因为一种语料可能数据量不够多，所以到底有没有办法在模型的各个层次上都做迁移学习，这样的话我们就有办法可以利用各种语料的数据，整合起来训练一个更好的序列到序列的转换模型。

## 研究方向二：鸡尾酒会问题

## Cocktail Party Problem

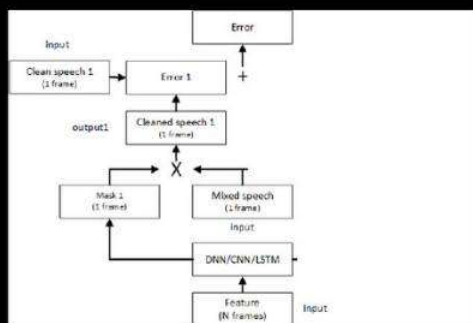


- Trace and recognize at least one stream of speech in the mixture of multi-talker speech and noise
- A key and difficult problem in far-field speech recognition
- Requires most research work

众所周知，在非常嘈杂或者多人同时说话的环境中，人有一个非常好的特点，即能够把注意力集中在某一个人的声音上，屏蔽掉周围的说话声或者噪音，非常好地听懂所需关注之人的说话声音。现在，绝大多数语音识别系统无法做到这一点。如果不做特殊处理，你会发现只要旁边有人说话，语音识别系统的性能就急剧下降。

由于人的信噪比非常大，这个问题在近场麦克风时并不明显；但在远场情况下，信噪比下降很厉害，问题也就变得很突出，进而成为了一个难以解决的关键问题。

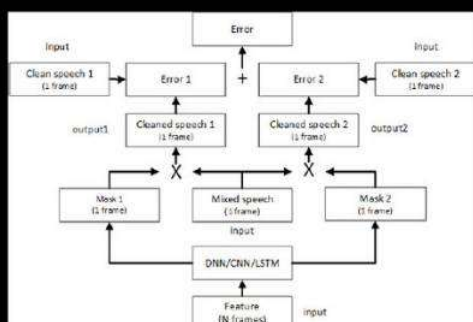
## Speech + Others



- Most widely studied
- Speech + noise (music, etc)
- Specific speaker + other speakers
- Key: convert the problem from an unsupervised learning problem to a supervised one

鸡尾酒会中一个相对简单的问题是语音加上噪声（或者语音加上音乐、语音加上其他的東西）。因为你已经知道要关注的语音部分，可以忽略掉其他，所以这个问题就可以从之前的非监督学习盲分类问题，转换到人为定制的 supervision 信息的有监督学习问题。

## Multi-Talker Mixed Speech : Label Permutation

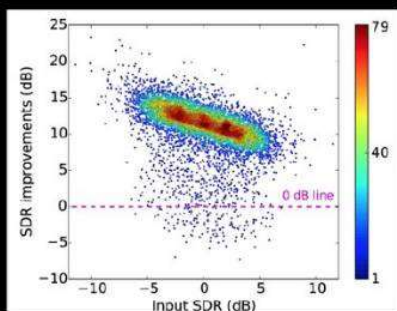


- Task: speaker-independent multi-talker mixed speech separation
- Simple supervised training does not work due to label permutation problem
- Speaker 1 should be assigned to output 1 or output 2 ?

但是有监督学习会在多人说话时碰到困难，这个困难就在于这时你无法轻易地提供 supervision 信息，因为当麦克风收到信息时，它收到了两个或者多个麦克风的混合语音，但并不能知道这个混合语音是 A+B 还是 B+A（因为两者结果是一样的）。所以在训练过程当中，你无法预先知道是把说话人 A 的声音作为输出 1 的 supervision 还是输出 2 的 supervision。这个问题有一个专门的术语叫做标签排列问题（Label Permutation Problem），目前它有两个比较好的解决方案：

### 方案一：Deep Clustering

## Multi-Talker Mixed Speech : Deep Clustering

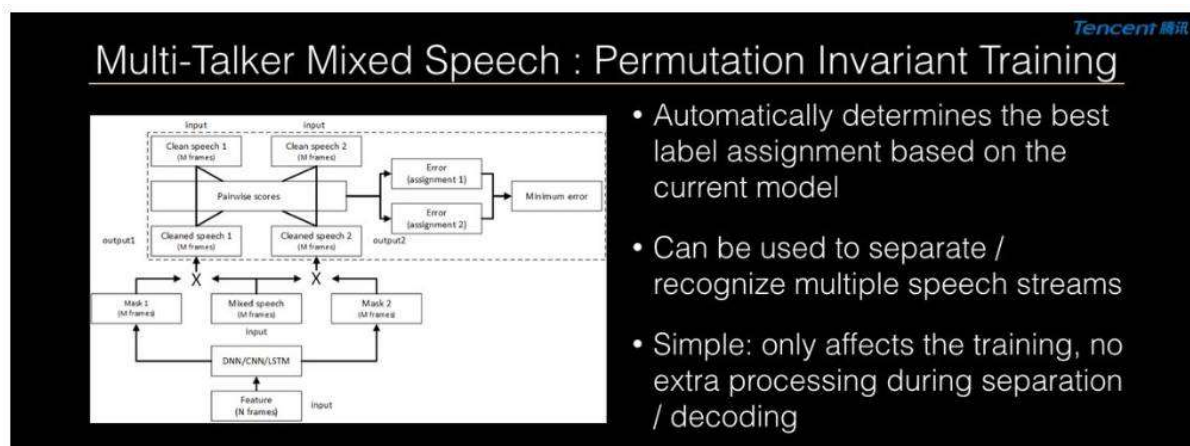


- Learn a unit-size embedding for each time-frequency bin
- Bins belong to the same speaker are close in the embedding space, and farther away otherwise
- Separation is done by clustering embedding space representations (i.e., segment the bins)

假设当两个人说话时，每一个时频点都会被一个说话人掌控；在这个情况下，它可以把整个语谱图分割成两个集群，一个属于说话人 A，一个属于说话人 B，进而训练一个嵌入空间表达。如果两个时频点同属一个说话人，它们在嵌入空间里的距离则比较近；如果属于不同的说话人，距离则比较远。所以训练准则是基于集群的距离来定义的，在识别的时候，它首先将语音信号映射到嵌入空间，然后上面训练一个相对简单的集群，比如用 k-means 这样的方法。这个想法非常有意思，但是同时聚类算法的引入也带来了一些问题，使得训练和识别变得相对复杂，也不易于与其他方法融合。

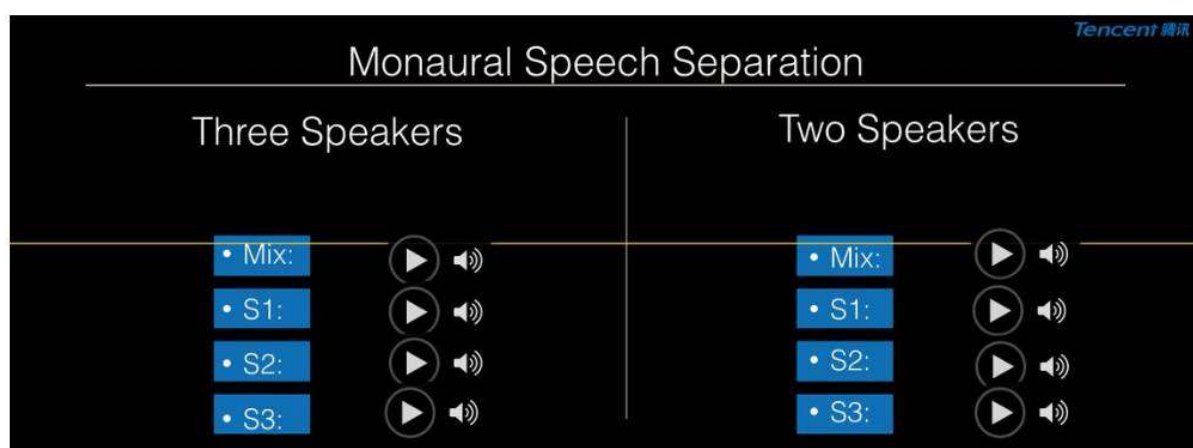


## 方案二：Permutation Invariant Training



这个想法是因为我们真正做分离的时候，其实并不在乎它是 A+B 还是 B+A，而只关注两个信号分离的水准是不是好。在真正做判定的时候，我们其实会专门比较音频信号，并选择成本最小的作为最后判别的分类。这也提醒我们在训练时也可以这样做。所以训练时怎么做呢？

每次我拿到新的混合语音时，并不预先设定它的 supervision 标签什么样，而是针对当前模型动态地决定当下我的 supervision 应该是什么样。由于取 supervision 的最小错误率，所以我又在其上进一步优化，它的错误率也进一步减小，这是其基本想法。它唯一需要改变的就是训练的标注分配，其他部分则不用变。所以识别相对简单，也很容易与其他方法做融合。



那么我放几个声音大家听一下：

- 三个说话人：三个人混合的声音是比较难分离。这个方法的另外一个好处是不需要预先知道有几个人说话，所以当有两个人说话的时候，它也能做得很好。
- 两个说话人：当有两个说话者时，第三个数据就没有输出，只有保留沉默，所以它有一个非

常好的特性：不需要你做特殊处理，输出结果即分离结果。

## Cocktail Party Problem

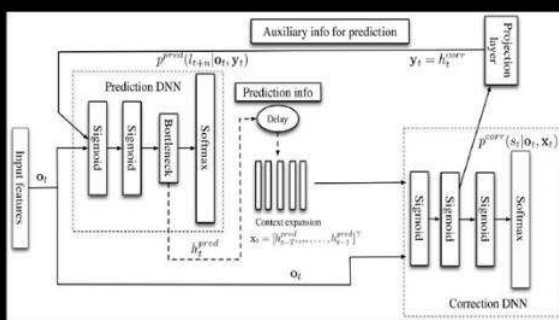
- Raw signal: How to best exploit multi-channel information in PIT?
- More powerful model: Are there models that are better than conventional LSTMs for speech separation and tracing?
- Other constraints: How to exploit additional information, such as language model and decoding information, to improve speech enhancement and separation?

但是目前为止，我们所使用的一些信息只来自单麦克风。众所周知，麦克风阵列可以提供很多信息，所以：

- 第一个很重要的问题是如何有效地利用多麦克风信息来继续加强它的能力；
- 第二个问题是说我们有没有办法找到一个更好的分离模型，因为现在大家使用的依然是 LSTM，但是其不见得是最佳模型。
- 第三个问题是说我们有没有办法利用其他的信息作为约束进一步提升它的性能。

### 研究方向三：持续预测与适应的模型

## Models that Continuously Predict and Adapt




- Prediction-adaptation-correction RNN
- Continuously adapt the AM for better next-moment performance using predicted information
- Fast implicit adaptation with augmented information

第三个大家关注的研究热点是能否建造一个持续地做预测（prediction）和适应（adaptation）的系统。我们之前做了一个模型，如上图所示；它的优势是能够非常快地做适应，持续地做预测，然后改进下一帧的识别结果。但是由于目前这个模型回路比较大，所以性能上还是很难训练，这和 CTC 模型情况相似。所以我们现在的问题是如何建造一个更好的模型能够持续地做预

测。这种模型需要有哪些特性呢？

- 一是模型能够非常快地做适应；
- 二是可以发现一些一致的规律性，并将其变为长远记忆里面的信息，使得下一次再做识别时会变成稳定的状态，其他状态则变成需要适应的状态；
- 三是我们有没有办法把类似说话者的信息，用更好的方式压缩在其模型之中，所以当见到一个新说话者时，可以很快地做适应。

#### 研究方向四：前端与后端联合优化

Frontend-Backend Joint Optimization

- How to combine signal processing (usually simplified with assumptions) and machine learning (usually requires large data) techniques?
- How to automatically balance between frontend processing and backend processing thorough joint optimization?
- How to design the backend so that it is robust to frontend differences

第四个研究前沿是出于远场识别的需要，即如何更好地做前端和后端的联合优化。这其中包含几个问题，因为传统来讲，处理前端信号使用的是信号处理技术，其一般只用到当前状态下的语音信号信息，比如训练集信息；而机器学习方法，则用到很多训练器里的信息，并很少用到当前帧的信息，也不会对它进行数据建模，所以我们能否把这两种方法更好地融合在一起，是目前很多研究组织正在继续努力的一个方向。

另外，我们是否有办法更好地联合优化前端的信号处理与后端的语音识别引擎。因为前端信号处理有可能丢失信息，且丢失的信息很可能无法在后端恢复，所以我们能否做一个自动系统以分配这些信息的信号处理，使得前端更少地丢失信息，后端则这些信息更好地利用起来。

今天的演讲就到这，谢谢大家。

