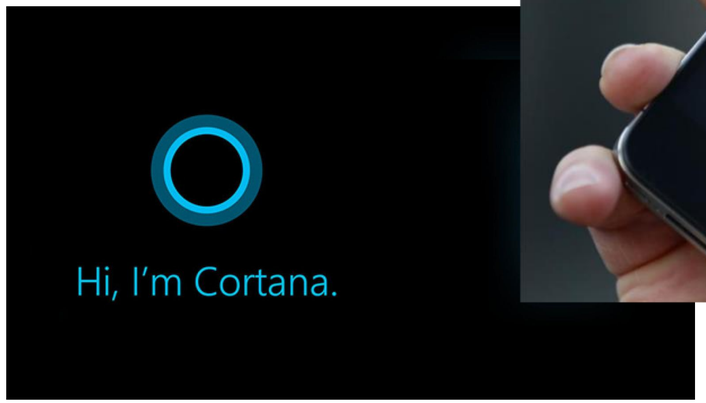
ASR

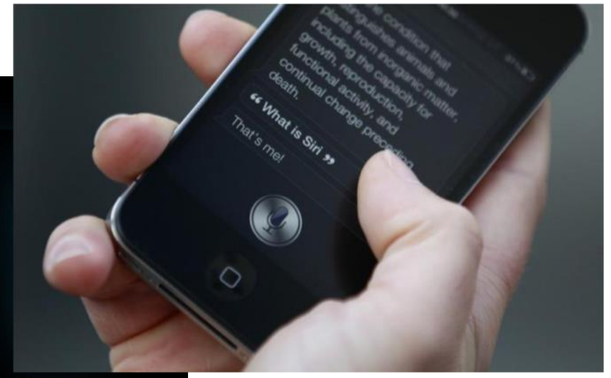
Automatic Speech Recognition

ASR概念

自动语音识别(Automatic Speech Recognition, 简称ASR)技术是使人与人、人与机器更顺畅交流的关键技术

- 语音识别是智能助手的第一步
- 苹果Siri，微软Cortana，谷歌Home，亚马逊 Alexa
- 语音识别ASR与问答系统QA



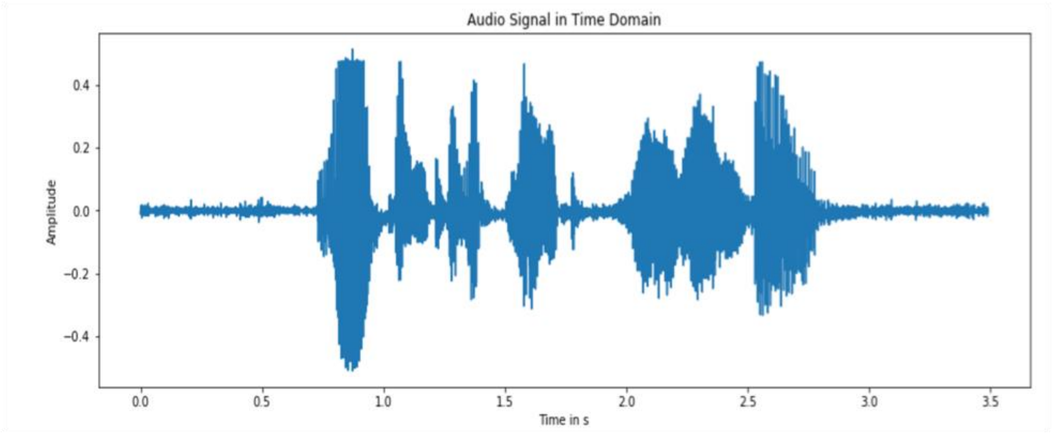


为什么ASR这么难

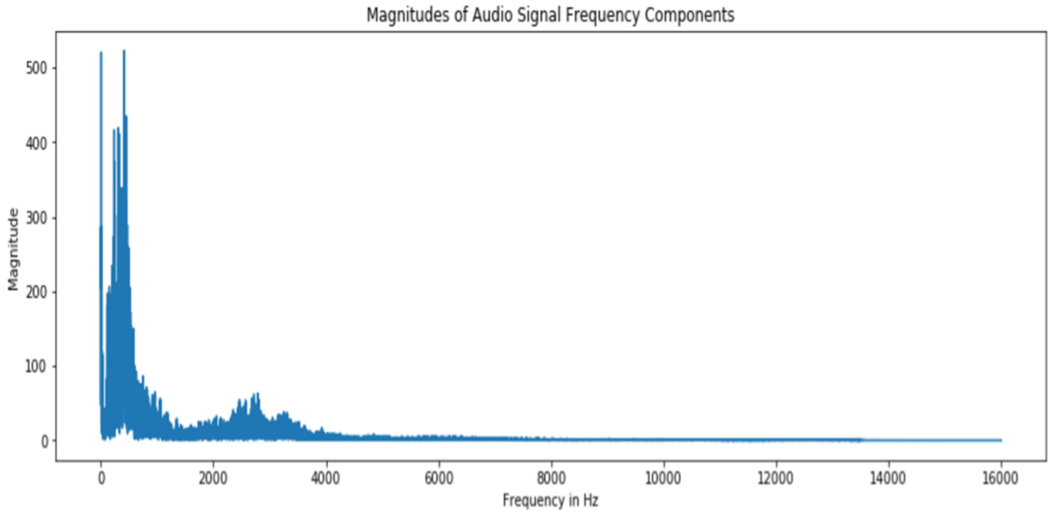
- 词汇表的大小, 同音词
- 口音, 语气, 腔调
- 噪声, 鸡尾酒问题, 录音质量
- 说话的方式, 感情

语音信号

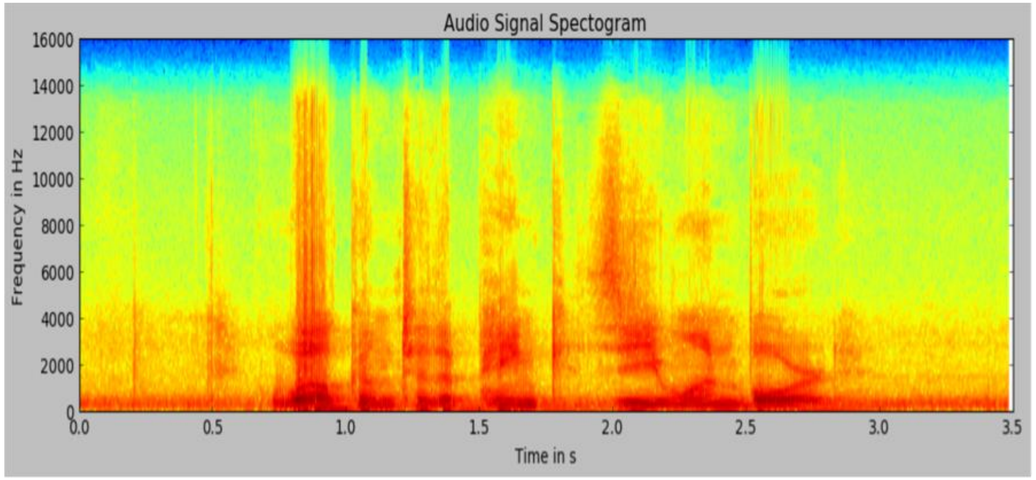
- 语音信号是声波, 也是时变信号.
- 有波形图, 频域图, 时频谱多种表示法



波形图



FFT 频域图



STFFT 时频谱

ASR基本方程

对于给定的语音信号X和单词W, 最优的识别结果W*可表述为:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

应用贝叶斯公式:

$$P(\mathbf{W} \mid \mathbf{X}) = \frac{p(\mathbf{X} \mid \mathbf{W}) P(\mathbf{W})}{p(\mathbf{X})}$$

$$\propto p(\mathbf{X} \mid \mathbf{W}) P(\mathbf{W})$$

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \underbrace{p(\mathbf{X} \mid \mathbf{W})}_{\text{Acoustic Model 声学模型}} \quad \underbrace{P(\mathbf{W})}_{\text{Language Model 语言模型}}$$

经典架构

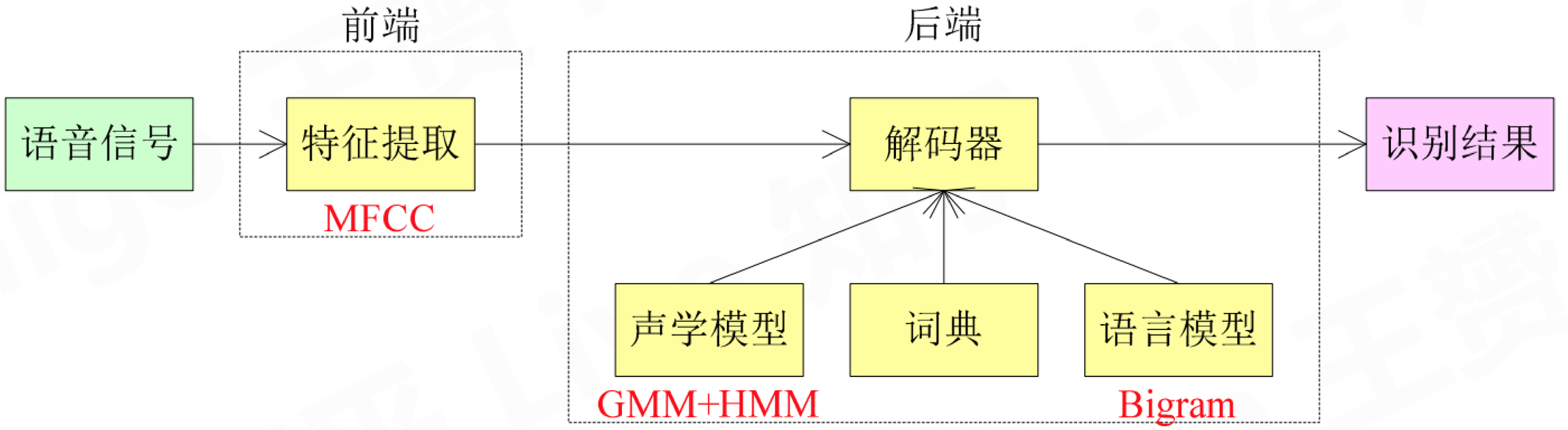
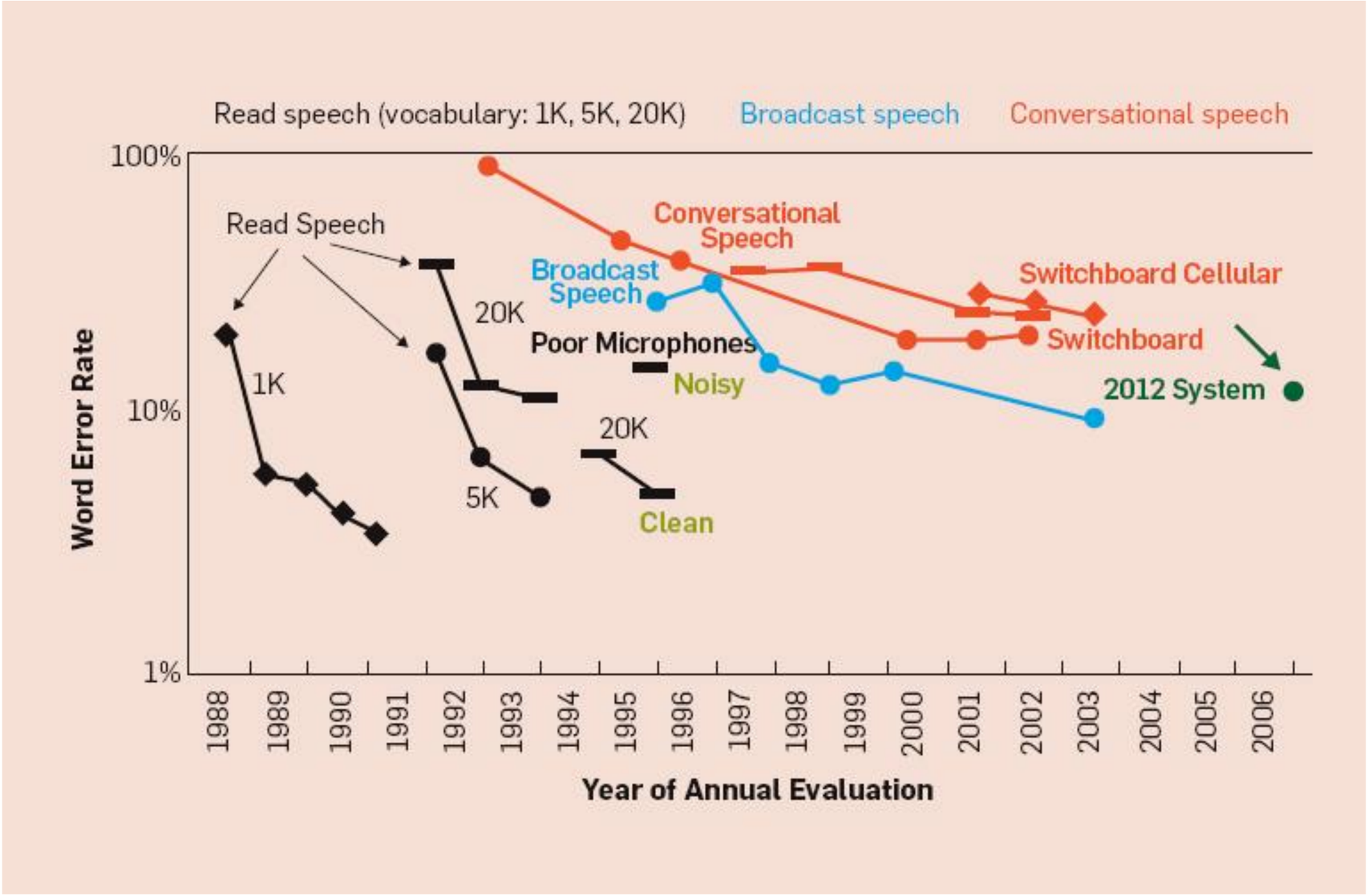


image by maigo



2006年以前的ASR性能

评价指标WER

- 计算方法
  - 将标准答案和识别结果对齐
  - 错误率等于替换,插入,删除的总数除以标准答案长度
  - 对齐应使错误率最小.

- 标准答案：too young too simple sometimes naive
- 识别结果：too young simple some times knife
- 错误： 删除 替换 插入 替换
- 词错误率 (word error rate)： 4 / 6 = 66.7%

image by maigo

最优对齐不一定唯一

- 标准答案：too young too simple sometimes naive
- 识别结果：too young simple some times knife
- 错误： 删除 替换 替换 插入
- 词错误率：4 / 6 = 66.7%

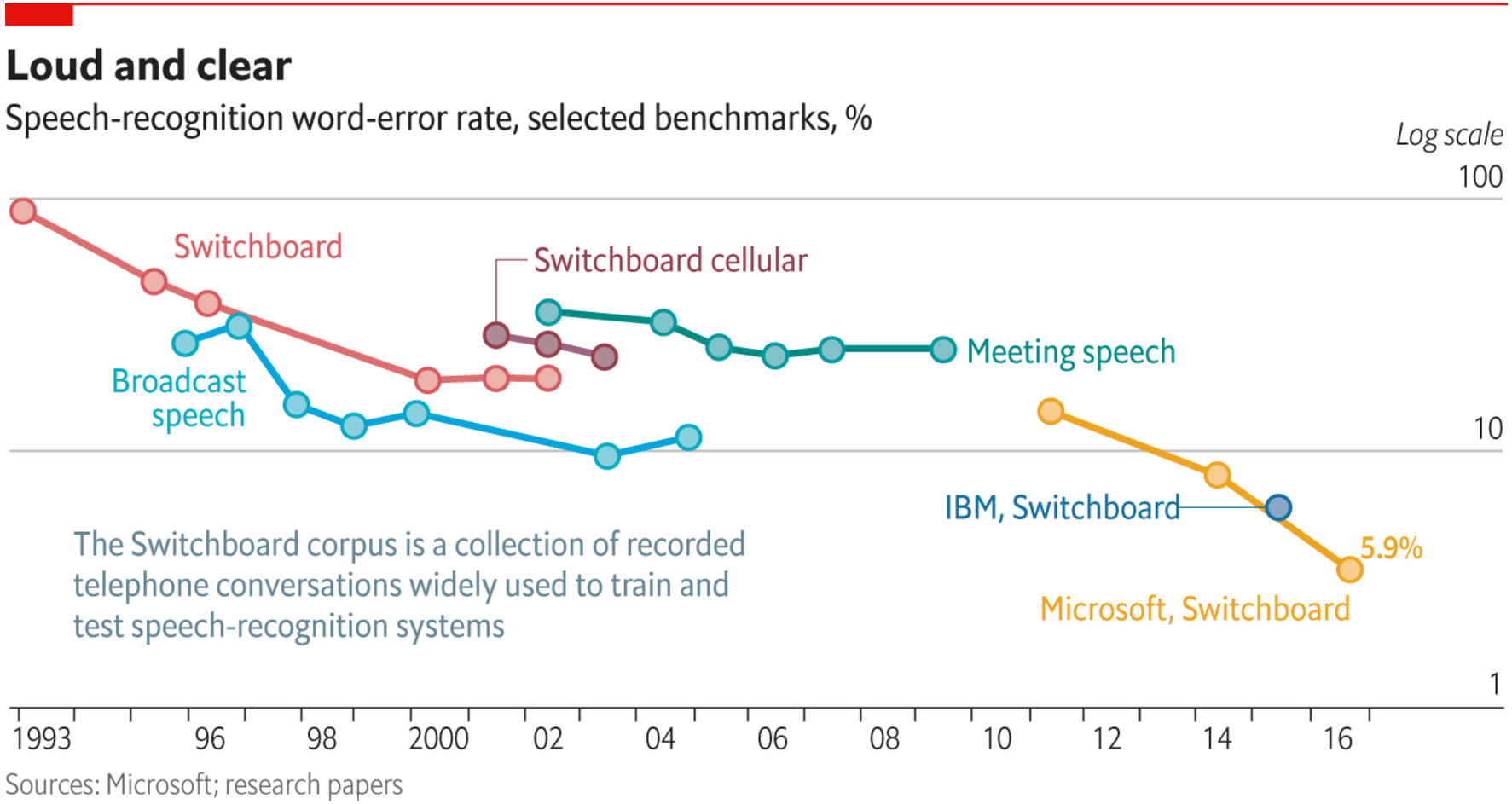
WER可能高于100%

- 标准答案：recognize speech
- 识别结果：wreck a nice beach
- 错误： 替换 插入 插入 替换
- 词错误率：4 / 2 = 200%

image by maigo

Alex Graves，Google DeepMind研究员
语音识别多项技术开创者
开启了神经网络处理语音问题的时代

1. **Connectionist temporal classification labelling unsegmented sequence data with recurrent neural networks, ICML 2006.**
2. Speech recognition with deep recurrent neural networks, ICASSP 2013.
3. Hybrid speech recognition with deep bidirectional LSTM, ASRU 2013.
4. **Towards End-To-End Speech Recognition with Recurrent Neural Networks**, ICML 2014.



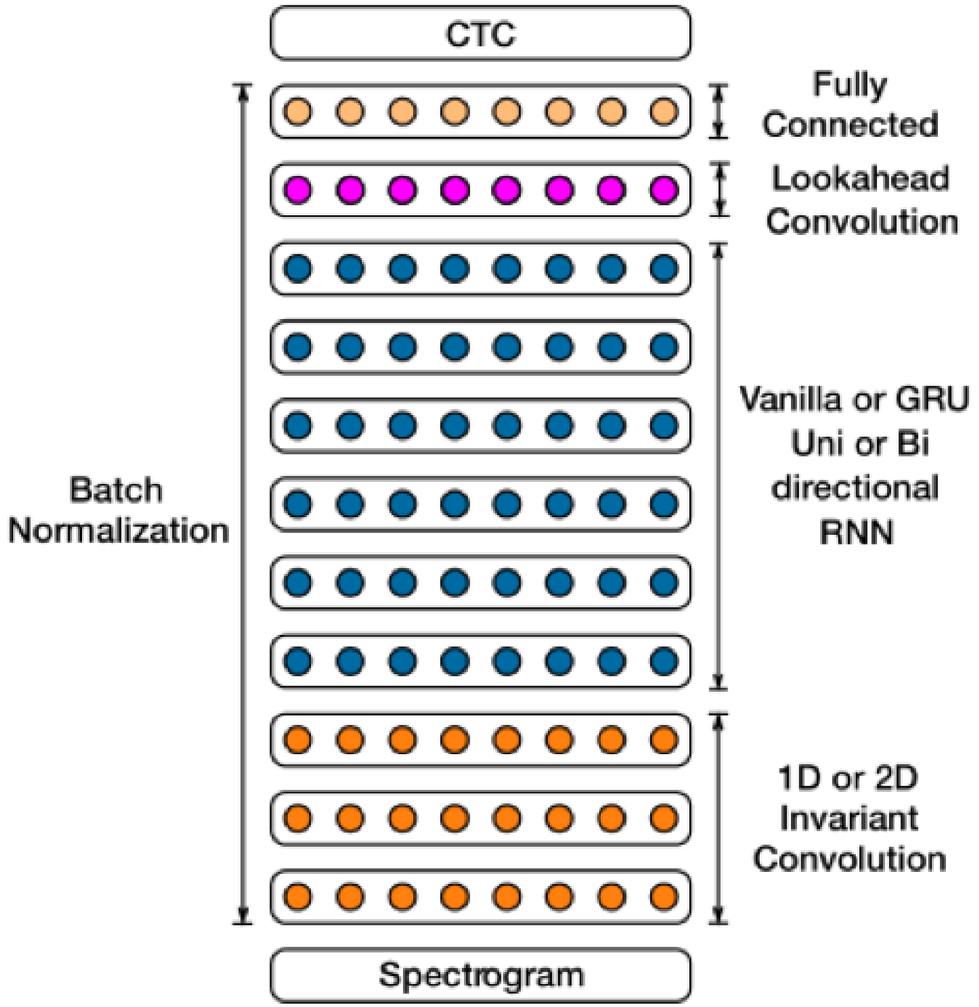
2006年以后的ASR性能

用RNN处理一下怎么样?

输入: $X = [x_1, x_2, \ldots, x_T]$

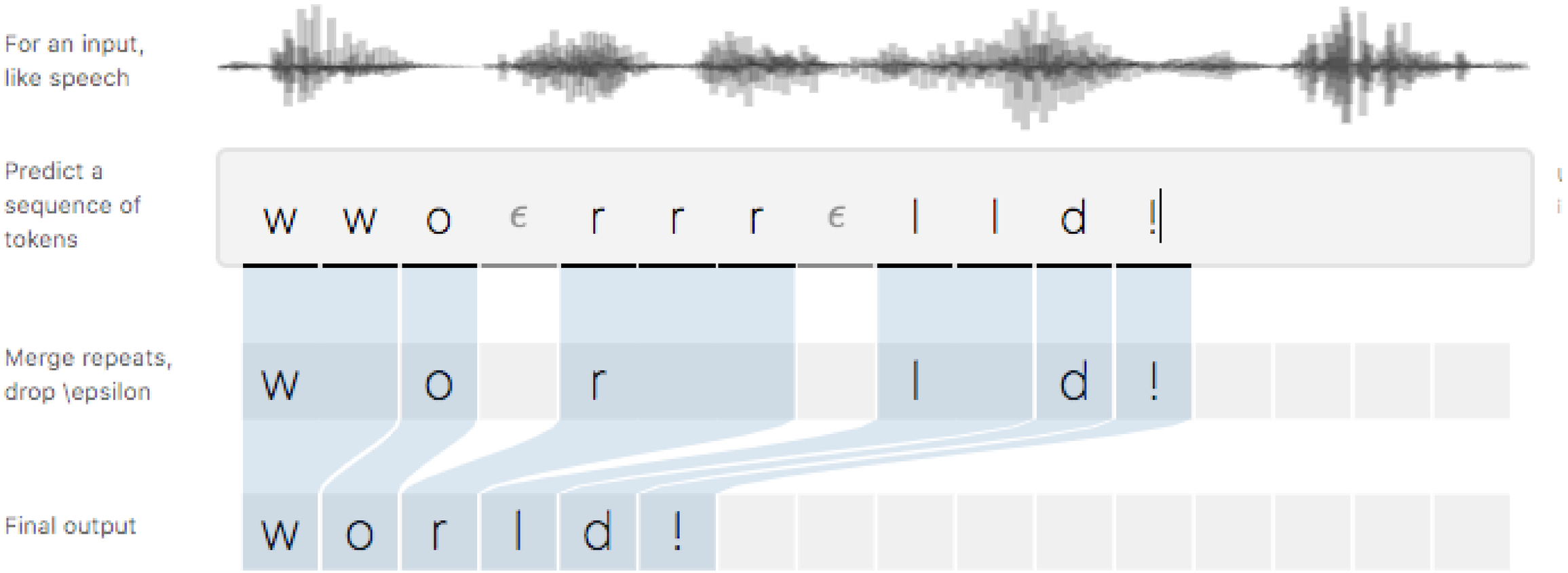
输出: $Y = [y_1, y_2, \ldots, y_U]$

我们希望能找到一个输入到输出的映射函数,
并且这个函数是可导的



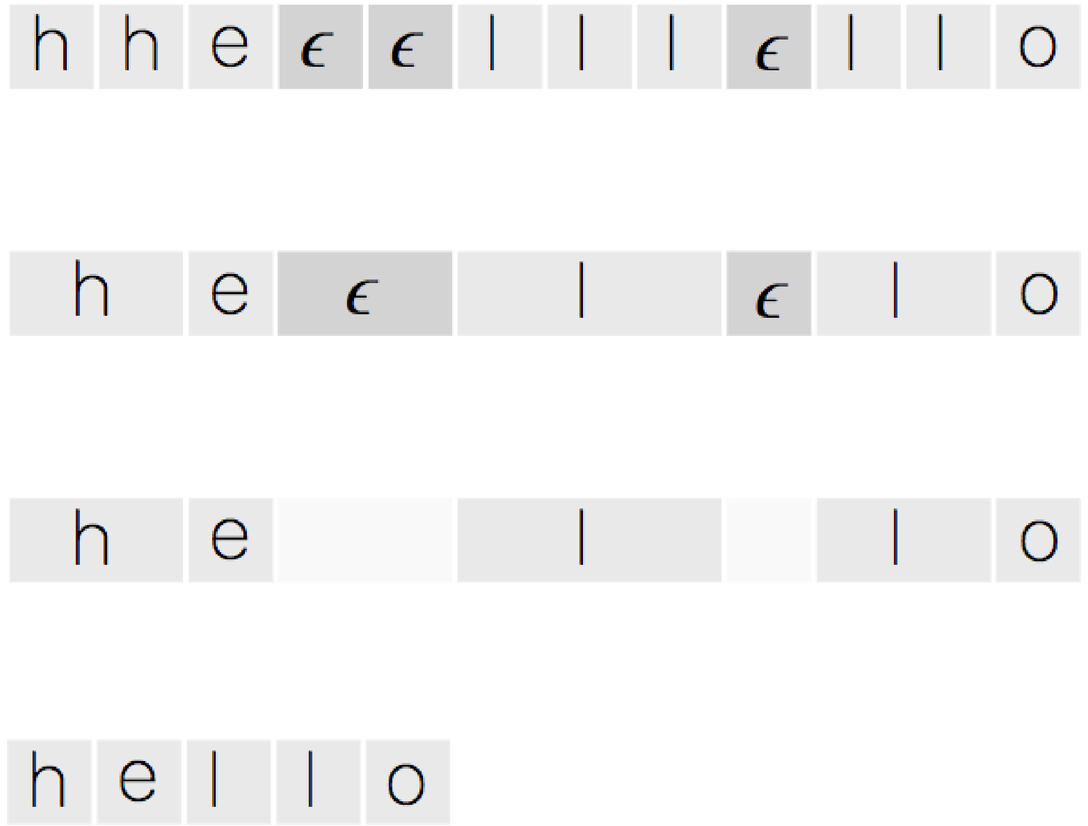
- 挑战在于:
  - X和Y都是变长的
  - 我们并没有一种确定的X和Y的对齐方式

How CTC collapsing works



https://distill.pub/2017/ctc/

CTC折叠

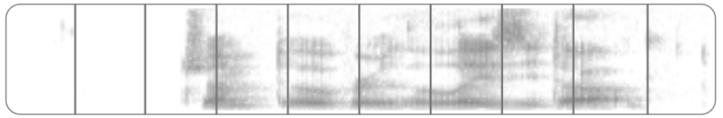


First, merge repeat characters.

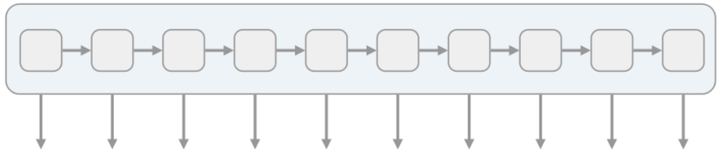
Then, remove any ϵ tokens.

The remaining characters are the output.

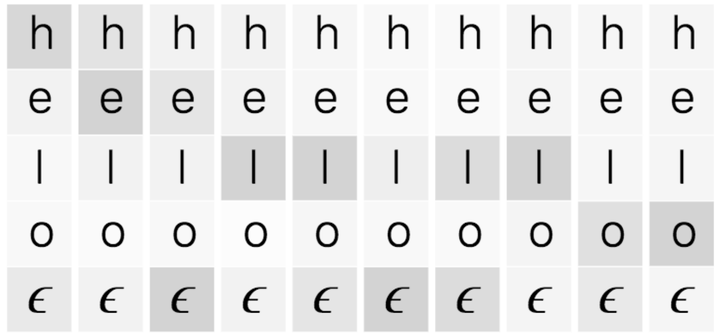
- 通过RNN网络后形成一个概率分布矩阵
- 可以生成无数的识别Sequence
- Sequence折叠后有不同的输出
- 我们只有一个标签
- 如何计算Loss?



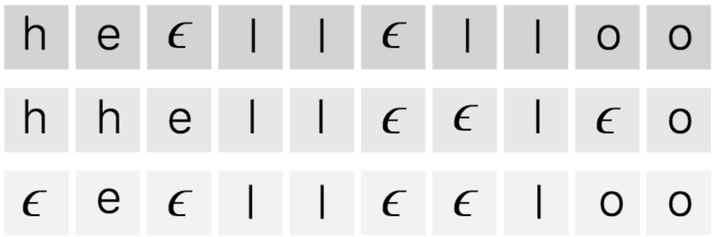
We start with an input sequence, like a spectrogram of audio.



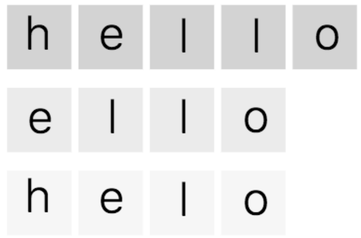
The input is fed into an RNN, for example.



The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.



With the per time-step output distribution, we compute the probability of different sequences



By marginalizing over alignments, we get a distribution over outputs

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

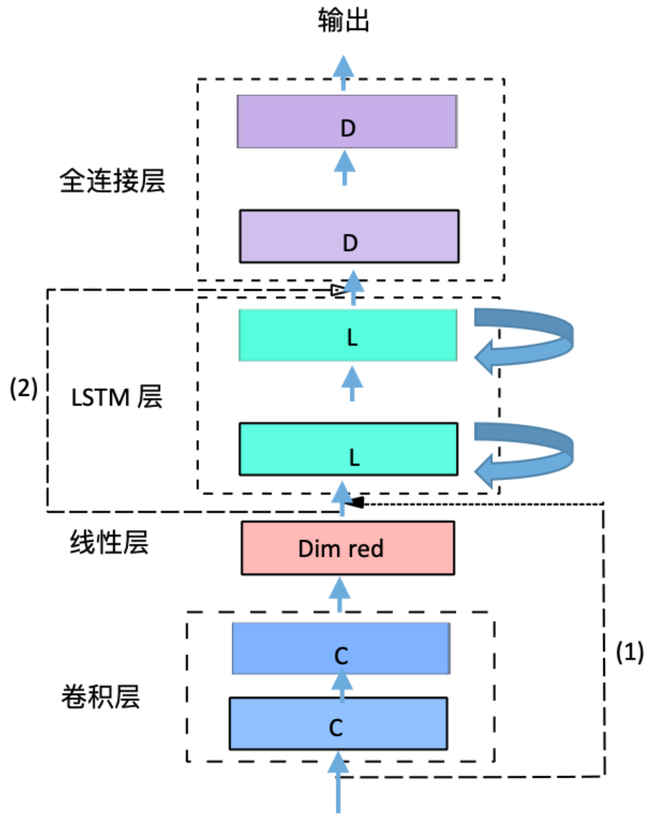
The CTC conditional **probability**

marginalizes over the set of valid alignments

computing the **probability** for a single alignment step-by-step.

CLDNN

- CLDNN结合了卷积网络，LSTM和DNN
  - 当输入信号进行时间域的卷积操作之后，输出数据再进行一次频率域的卷积操作以减少频谱的变化之后再通过三层LSTM，最后再通过一层DNN。
- 训练过程中，时间卷积层和其他层会一起进行训练。
- 输入数据为以时间为下标的连续向量



[X] Sainath, Tara N., Oriol Vinyals, Andrew Senior, and Haşim Sak. "Convolutional, long short-term memory, fully connected deep neural networks." ICASSP 2015.

audioPlot

- git clone http://gitlab.icenter.tsinghua.edu.cn/saturnlab/audioPlot.git
- 采集录音，语音预处理，统一格式

指令识别audioNet

- git clone http://gitlab.icenter.tsinghua.edu.cn/saturnlab/audioNet.git
- cd audioNet && ./install.sh
- python -m augmentation & 启动数据增强服务
- python train.py 开始训练

androidAudioRecg

- http://gitlab.icenter.tsinghua.edu.cn/saturnlab/androidAudioRecg
- An Android app for speech recognition based on TensorFlow library.
- Install Android Studio

谢谢指正!