序列判别式训练 (Sequence Discriminative Training)

大纲

- 知识点1：语音模型训练的最大似然损失
- 知识点2：语音模型训练的判别式损失函数(MMI,MPE,SMBR)
- 知识点3：Lattice free MMI

Fundamental Equation of Statistical Speech Recognition

If $\mathbf{X}$ is the sequence of acoustic feature vectors (observations) and $\mathbf{W}$ denotes a word sequence, the most likely word sequence $\mathbf{W}^*$ is given by

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

Applying Bayes' Theorem:

$$\begin{aligned} P(\mathbf{W} \mid \mathbf{X}) &= \frac{p(\mathbf{X} \mid \mathbf{W})\, P(\mathbf{W})}{p(\mathbf{X})} \\ &\propto p(\mathbf{X} \mid \mathbf{W})\, P(\mathbf{W}) \\ \mathbf{W}^* &= \arg\max_{\mathbf{W}} \underbrace{p(\mathbf{X} \mid \mathbf{W})}_{\text{Acoustic model}} \; \underbrace{P(\mathbf{W})}_{\text{Language model}} \end{aligned}$$

NB: $\mathbf{X}$ is used hereafter to denote the output feature vectors from the signal analysis module rather than DFT spectrum.

P(X|W)

- HMM建模Acoustic Model ,记当M
- $P(X|W)=P(X|M(W))$
- 训练数据（Wi, Xi）i=1,2…n

最大似然估计MLE

INTRODUCTION

- **Conditional probability**
  - The notion of *conditional probability* allows us to incorporate other potentially important variables
  - Mathematically, we write

$$P(X \mid Y)$$

  - meaning the probability of X *conditional on Y* or *given Y*

Introduction

- **Conditional probability**
  - The probability of an outcome will be *conditional* upon the parameter values of this model. In the case of the coin toss,

$$P\ (H \mid p=0.5)$$

  - where H is the event of obtaining a head and p is the model parameter, set at 0.5

- **Conditional probability**
  - Say we toss a coin a number of times and record the number of times it lands on heads
  - The probability distribution that describes just this kind of scenario is called the *binomial* probability distribution. It is written as follows :

$$\frac{n!}{h!(n-h)!}p^{h}(1-p)^{n-h}$$

    - n = total number of coin tosses
    - h = number of heads obtained
    - p = probability of obtaining a head on any one toss

- **The concept of likelihood**
  - If the probability of an event X dependent on model parameters *p* is written

    $$P (X \mid p)$$

    then we would talk about the likelihood

    $$L (p \mid X)$$

  - That is, the likelihood of *the parameters given the data*

  - **The aim of maximum likelihood estimation is to find the parameter value(s) that makes the observed data most likely**

- **The concept of likelihood**
  - In the case of *data analysis*, we have already observed all the data: once they have been observed they are fixed, there is no 'probabilistic' part to them anymore (the word data comes from the Latin word meaning 'given')
  - We are much more interested in the likelihood of the model parameters that underlay the fixed data.

    - **Probability**

      Knowing parameters -> Prediction of outcome

    - **Likelihood**

      Observation of data -> Estimation of parameters

- **A simple example of MLE**
  - How would we go about this in a simple coin toss experiment?
  - That is, rather than assume that p is a certain value (0.5) we might wish to find the *maximum likelihood estimate* (MLE) of p, given a specific dataset.
  - Beyond parameter estimation, the likelihood framework allows us to make *tests* of parameter values. For example, we might want to ask whether or not the estimated p differs *significantly* from 0.5 or not. This test is essentially asking: is there evidence that the coin is biased?

- **A simple example of MLE**
  - Say we toss a coin 100 times and observe 56 heads and 44 tails. Instead of *assuming* that *p* is 0.5, we want to find the MLE for *p*. Then we want to ask whether or not this value differs significantly from 0.50
  - How do we do this? We find the value for *p* that makes the observed data most likely
  - As mentioned, the observed data are now fixed. They will be constants that are plugged into our binomial probability model :-

    n = 100 (total number of tosses)

    h = 56 (total number of heads)

Model-fitting

- **A simple example of MLE**
  - Imagine that *p* was 0.5. Plugging this value into our probability model as follows :-

$$L(p=0.5 \mid data) = \frac{100!}{56!44!} 0.5^{56} 0.5^{44} = 0.0389$$

  - But what if *p* was 0.52 instead?

$$L(p=0.52 \mid data) = \frac{100!}{56!44!} 0.52^{56} 0.48^{44} = 0.0581$$

  - So from this we can conclude that *p* is more likely to be 0.52 than 0.5

- **A simple example of MLE**
  - We can tabulate the likelihood for different parameter values to find the maximum likelihood estimate of *p*:

p	L
0.48	0.0222
0.50	0.0389
0.52	0.0581
0.54	0.0739
0.56	0.0801
0.58	0.0738
0.60	0.0576
0.62	0.0378

HMM参数估计最大似然

Maximum likelihood objective:

$$F_{ML}(\lambda) = \sum_{u} \log p_{\lambda}(X_u | W_u)$$

大纲

- 知识点1：语音模型训练的最大似然损失
- 知识点2：语音模型训练的判别式损失函数(MMI,MPE,SMBR)
- 知识点3：Lattice free MMI

最大似然的问题

Switch training criterion

- Maximum likelihood is theoretically optimal (even for classification), but only when the model is correct
- If not, an explicitly discriminative training criterion might be better
- Minimum Classification Error (MCE) is a natural choice for classification, but not for sequences
- Try to maximise mutual information instead

Objective functions: MMI & ML

ML objective function is product of data likelihoods given speech file $\mathcal{O}_r$.

$$\mathcal{F}_{\mathrm{ML}}(\lambda) = \sum_{r=1}^{R} \log p_\lambda\left(\mathcal{O}_r | s_r\right), \tag{1}$$

MMI objective function is posterior of correct sentence:

$$\begin{aligned} \mathcal{F}_{\mathrm{MMIE}}(\lambda) &= \sum_{r=1}^{R} \log \frac{p_\lambda\left(\mathcal{O}_r | s_r\right)^\kappa P(s_r)^\kappa}{\sum_s p_\lambda\left(\mathcal{O}_r | s\right)^\kappa P(s)^\kappa} \\ &= \sum_{r=1}^{R} \log P^\kappa\left(s_r | \mathcal{O}_r, \lambda\right) \end{aligned} \tag{2}$$

K:概率scale参数，可以调节，以便在测试集获得更好的性能
接下来要讲一下MMI的推导，在论文中简单过一遍

熵

2.1 Entropy and Mutual Information

An intuitively plausible measure of the uncertainty in a random event X is the average number of bits necessary to specify the outcome of X under an optimal encoding scheme. By the fundamental theorem of information theory, also known as the noiseless coding theorem, this measure is the *entropy* of X [Shannon 48]:

$$\mathrm{H}(X) \stackrel{\Delta}{=} -\sum_{x} \Pr(X = x) \log \Pr(X = x). \tag{2.1}$$

Similarly, a formal measure of the uncertainty in a random event X given the outcome of a random event Y is the *conditional entropy* of X given Y:

$$\mathrm{H}(X \mid Y) \stackrel{\Delta}{=} -\sum_{x,y} \Pr(X = x, Y = y) \log \Pr(X = x \mid Y = y). \tag{2.2}$$

互信息(Mutual information

An intuitively plausible measure of the average amount of information provided by the random event Y about the random event X is the average difference between the number of bits it takes to specify the outcome of X when the outcome of Y is not known and when the outcome of Y is known. This is just the difference in the entropy of X and the conditional entropy of X given Y:

$$\begin{aligned} \mathrm{I}(X;Y) &\triangleq \mathrm{H}(X) - \mathrm{H}(X \mid Y) \\ &= \sum_{x,y} \Pr(X = x, Y = y) \log \frac{\Pr(X = x, Y = y)}{\Pr(X = x)\Pr(Y = y)}. \end{aligned} \tag{2.3}$$

Since $\mathrm{I}(X;Y) = \mathrm{I}(Y;X)$, $\mathrm{I}(X;Y)$ is referred to as the *average mutual information* between X and Y.

语音Y与文本W

Let $\boldsymbol{W}$ be a random variable over sequences of words. Let $\boldsymbol{Y}$ be a random variable over sequences of acoustic information. On average, the uncertainty of a word sequence given a sequence of acoustic information is the conditional entropy of $\boldsymbol{W}$ given $\boldsymbol{Y}$:

$$\mathrm{H}(\boldsymbol{W} \mid \boldsymbol{Y}) = \mathrm{H}(\boldsymbol{W}) - \mathrm{I}(\boldsymbol{W}; \boldsymbol{Y}). \tag{2.4}$$

语音识别求解问题

We would like to choose the model, m, to minimise $\mathrm{H}_m(\boldsymbol{W} \mid \boldsymbol{Y})$. Analogous to equation (2.4) we have

$$\mathrm{H}_m(\boldsymbol{W} \mid \boldsymbol{Y}) = \mathrm{H}_m(\boldsymbol{W}) - \mathrm{I}_m(\boldsymbol{W}; \boldsymbol{Y}), \tag{2.11}$$

in which

$$\mathrm{H}_m(\boldsymbol{W}) = -\sum_{\boldsymbol{w}} \Pr(\boldsymbol{W} = \boldsymbol{w}) \log \Pr_m(\boldsymbol{W} = \boldsymbol{w}), \tag{2.12}$$

and

$$\mathrm{I}_m(\boldsymbol{W}; \boldsymbol{Y}) = \sum_{\boldsymbol{w},\boldsymbol{y}} \Pr(\boldsymbol{W} = \boldsymbol{w}, \boldsymbol{Y} = \boldsymbol{y}) \log \frac{\Pr_m(\boldsymbol{W} = \boldsymbol{w}, \boldsymbol{Y} = \boldsymbol{y})}{\Pr_m(\boldsymbol{W} = \boldsymbol{w}) \Pr_m(\boldsymbol{Y} = \boldsymbol{y})}. \tag{2.13}$$

语音识别求解问题

Suppose that a language model, ℓ, is given. We would like to choose a vector of acoustic parameters, $\boldsymbol{\theta}$, to maximize

$$I_{\ell,\theta}(\boldsymbol{W};\boldsymbol{Y}) = \sum_{\boldsymbol{w},\boldsymbol{y}} \Pr(\boldsymbol{W}=\boldsymbol{w},\boldsymbol{Y}=\boldsymbol{y}) \log\left(\frac{\Pr_{\ell,\theta}(\boldsymbol{W}=\boldsymbol{w},\boldsymbol{Y}=\boldsymbol{y})}{\Pr_{\ell}(\boldsymbol{W}=\boldsymbol{w})\Pr_{\ell,\theta}(\boldsymbol{Y}=\boldsymbol{y})}\right), \qquad (2.19)$$

where the ℓ and $\boldsymbol{\theta}$ subscripts indicate which models and parameters are used in the computation of the subscripted probabilities. Since we do not know $\Pr(\boldsymbol{W}=\boldsymbol{w},\boldsymbol{Y}=\boldsymbol{y})$, we must instead assume that our sample $(\boldsymbol{w},\boldsymbol{y})$ is representative and choose $\boldsymbol{\theta}$ to maximize

MMi推导

$$f_{mmi}(\theta)=\sum_{w,y} log \frac{Pr_{l,\theta}(W=w,Y=y)}{Pr_{l,\theta}(W=w)Pr_{l,\theta}(Y=y)}$$

$$=\sum_{w,y}\left(\log \frac{Pr_{l,\theta}(W=w,Y=y)}{Pr_{l,\theta}(Y=y)} - \log Pr_{l,\theta}(W=w) \right)$$

when lm model is give, $Pr_{l,\theta}(W=w)$ *is constant*

$$f_{mmi}(\theta)=\sum_{w,y}\left(\log \frac{Pr_{l,\theta}(W=w,Y=y)}{Pr_{l,\theta}(Y=y)} \right)$$

$$=\sum_{w,y} \log \frac{Pr_{l,\theta}(Y=y|W=w)*Pr_{l,\theta}(W=w)}{\sum_{w'} Pr_{l,\theta}(Y=y|W=w')*Pr_{l,\theta}(W=w')}$$

Objective functions: MPE

Minimum Phone Error (MPE) is the summed "raw phone accuracy" (#correct - #ins) times the posterior sentence prob:

$$\mathcal{F}_{\text{MPE}}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa \text{RawPhoneAccuracy}(s, s_r)}{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}$$

$$= \sum_{r=1}^{R} \sum_s P^\kappa (s_r|\mathcal{O}_r, \lambda) \text{RawPhoneAccuracy}(s, s_r) \qquad (3)$$

Equals the expected phone accuracy of a sentence drawn randomly from the possible transcriptions (proportional to scaled probability).

MPE： 将phone准确率考虑到loss function中，（当然也可以直接用word error rate，但这个实现起来复复杂，所以一般大家用的MPE）
上下两部分合在一起，记数P(k)(sr|or, lambda)

SMBR

MPE/sMBR

MBR(minimum Bayes risk)的目标函数是最小化各种粒度指标的错误，比如MPE是最小化phone级别的错误，sMBR最小化状态的错误。目标函数如下：

$$J_{MBR}(\theta; S) = \sum_{m=1}^{M} J_{MBR}(\theta; o^m, w^m) = \sum_{m=1}^{M} \sum_{w} P(w|o^m) A(w, w^m)$$

$$= \sum_{m=1}^{M} \frac{\sum_{w} P(o^m|s^w)^k P(w) A(w, w^m)}{\sum_{w'} P(o^m|s^{w'})^k P(w')}$$

Objective functions: Simple example

- Suppose correct sentence is "a", only alternative is "b".
- Let $a = p_\lambda(\mathcal{O}|\text{"a"})P(\text{"b"})$ (acoustic & LM likelihood), b is same for "b".
- ML objective function $= \log(a) +$ other training files.
- MMI objective function $= \log(\frac{a}{a+b}) +$ other training files.
- MPE objective function $= \frac{a\times 1 + b\times 0}{a+b} +$ other training files.

这里提一个问题：MPE和MMI看起来是一样的，主要是因为这里面只有两个侯选句子,a的准确率是1，b的准确率是0，然后实际情况确并不是这样的

Challenges of MMI

- Frame-level models are good, but sequence-level models are poor $\Rightarrow$ need to operate at the sequence level.
- It's hard to estimate the denominator probabilities over a complete sequence

$$\sum_{W} p_{\lambda}(X|W)P(W)$$

for anything beyond small tasks

follows:

$$\mathcal{F}_{\mathrm{MMI}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{O}_r|s_r)^\kappa P(s_r)^\kappa}{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa} \tag{2.2}$$

早期的做法N-best,但是对于长句子，效率非常低，因类存了许多冗余的信息，所以后面改用lattice，复用一些路径。

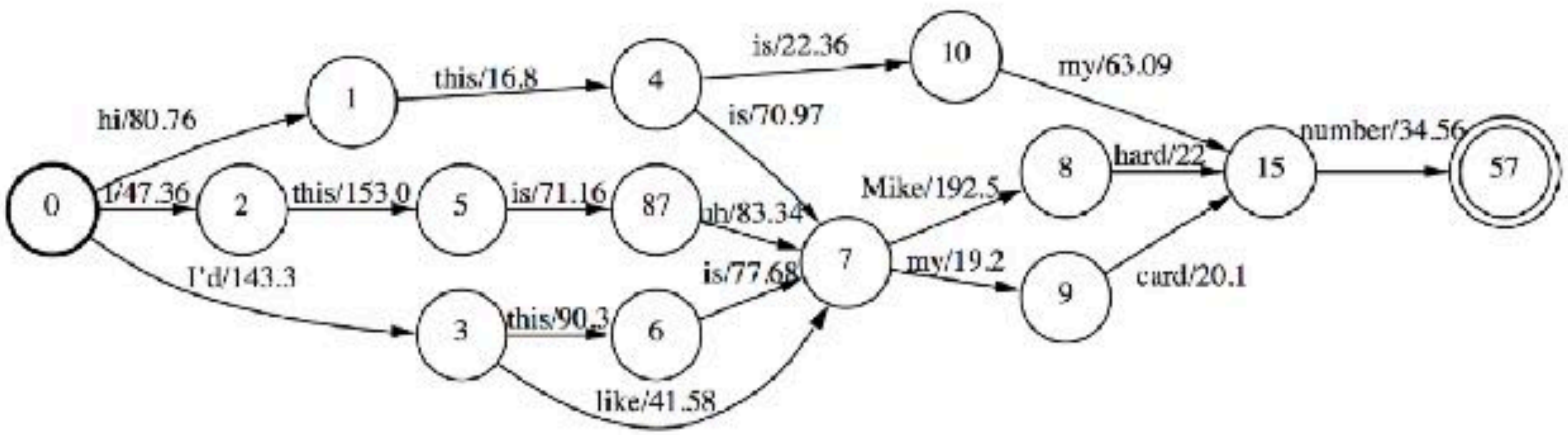
4.1.1 Need for lat

The MMI objective function can be expressed as a difference of HMM likelihoods. For R training files, this can be written

$$\mathcal{F}_{\mathrm{MMI}}(\lambda) = \sum_{r=1}^{R} \log p_\lambda^\kappa(\mathcal{O}_r|\mathcal{M}_r^{\mathrm{num}}) - \log p_\lambda^\kappa(\mathcal{O}|\mathcal{M}_r^{\mathrm{den}}), \tag{4.1}$$

Lattice-based MMI

- Approximate $\sum_W$ with a sum over a lattice
- Generate lattice for each utterance using an initial model
- Use a weak language model
- But attempt to minimise the size of the lattice
- Derive phone arcs from the lattice



1.用lattice估计全量的W
2.用最初的模型生成lattice
3.使用弱的语言模型，通常用1-gram的语言模型
4.但是控制lattice大小，通过beam控制，以减小计算的复杂度
5.继承解码后的边对应的phone

Lattices and MMI/MPE optimisation

- Lattices are generated once and used for a number of iterations of optimisation
- 2 sets of lattices-
  - Numerator lattice (= alignment of correct sentence)
  - Denominator lattice (from recognition). [Needs to be big, e.g beam > 125]
- Lattices need time-marked phone boundaries:
- Can't do unconstrained forward-backward because:
  i) slow and ii) interferes with the probability scaling which is done at whole-model level

Sequence training of hybrid HMM/DNN systems

- Can we train DNN systems with an MMI-type objective function? – **Yes**
- Forward- and back-propagation equations are structurally similar to forward and backward recursions in HMM training
- Initially train DNN framewise using cross-entropy (CE) error function
  - Use CE-trained model to generate alignments and lattices for sequence training
  - Use CE-trained weights to initialise weights for sequence training
- Train using back-propagation with sequence training objective function (e.g. MMI)

Sequence training results on Switchboard (Kaldi)

Results on Switchboard "Hub 5 '00" test set, trained on 300h training set, comparing maximum likelihood (ML) and discriminative (BMMI) trained GMMs with framewise cross-entropy (CE) and sequence trained (MMI) DNNs. GMM systems use speaker adaptive training (SAT).
All systems had 8859 tied triphone states.
GMMs – 200k Gaussians
DNNs – 6 hidden layers each with 2048 hidden units

	SWB	CHE	Total
GMM ML (+SAT)	21.2	36.4	28.8
GMM BMMI (+SAT)	18.6	33.0	25.8
DNN CE	14.2	25.7	20.0
DNN MMI	12.9	24.6	18.8

Veseley et al, 2013.

大纲

- 知识点1：语音模型训练的最大似然损失
- 知识点2：语音模型训练的判别式损失函数(MMI,MPE,SMBR)
- 知识点3：Lattice free MMI

LF-MMI

"Purely sequence-trained models for ASR based on lattice-free MMI"

(Povey et al, 2016)

- Solves a fundamental problem – a practical method for computing HMM "true" state posteriors using a DNN acoustic model
- Uses this to train a properly normalised sequence model, trained with MMI right from the start
- Removes the need for an acoustic scaling fudge factor

Getting it to work...

The core idea

- Both numerator and denominator state sequences are represented as *HCLG* FSTs
- Parallelise denominator forward-backward computation on a GPU
- Replace word-level LM with a 4-gram phone LM for efficiency
- Reduce the frame rate
  - might be a good idea for other reasons...
- Changes to HMM topology motivated by CTC (see Lecture 15)

Getting it to work...

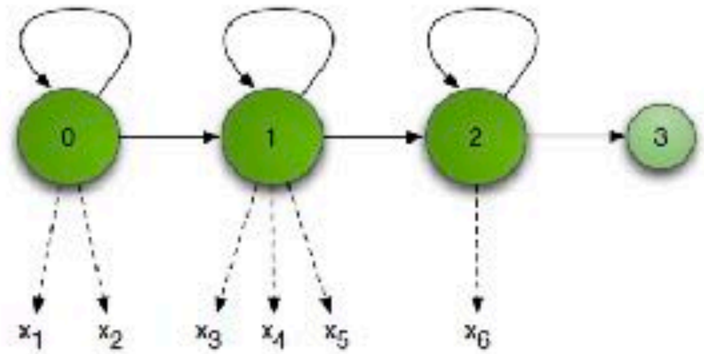
Extra tricks

- Train on small fixed-size chunks (1.5s)
  - probably enough to counter the flaws in the conditional independence assumption
- Careful optimisation of denominator FST to minimise the size
- Various types of regularisation

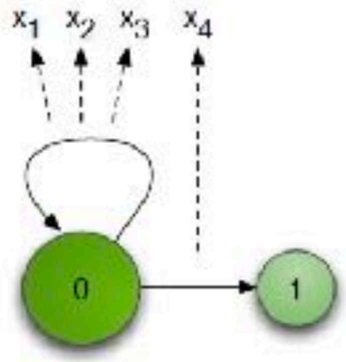
HMM topologies

Replace standard 3-state HMM with topology that can be traversed in a single frame

Standard topology



LF-MMI topology



Denominator FST

- LM is essentially a 3-gram phone LM
- No pruning and no backoff to minimise the size
  - Use of unpruned 3-grams means that there is always a 2-word history.
  - Minimises the size of the recognition graph when phonetic context is incorporated
- Addition of a fixed number of the most common 4-grams
- Conversion to *HCLG* FST in the normal way
- *HCLG* size reduced by a series of FST reversal, weight pushing and minimisation operations, followed by epsilon removal

生成分母FST代码：kaldi/src/chainbin/chain-make-den-fst.cc

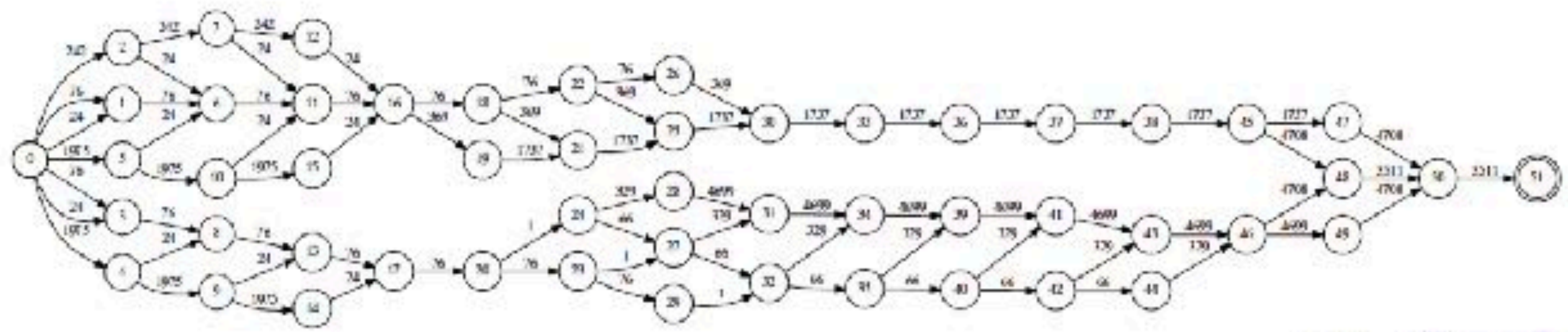
The normalisation FST

- The phone-LM assumes that we are starting at the beginning of an utterance $\rightarrow$ not suitable for use with 1.5s chunks
- Need to adjust the initial probabilities for each HMM state
- Iterate 100 times through the denominator FST to get better initial occupancy probabilities
$$\alpha_j^{(n)}(0) = \sum_i a_{ij} \alpha_i^{(n-1)}(0)$$
- Add a new initial state to the denominator FST that connects to each state with the new probabilities $\rightarrow$ the "normalisation FST"

Numerator FST

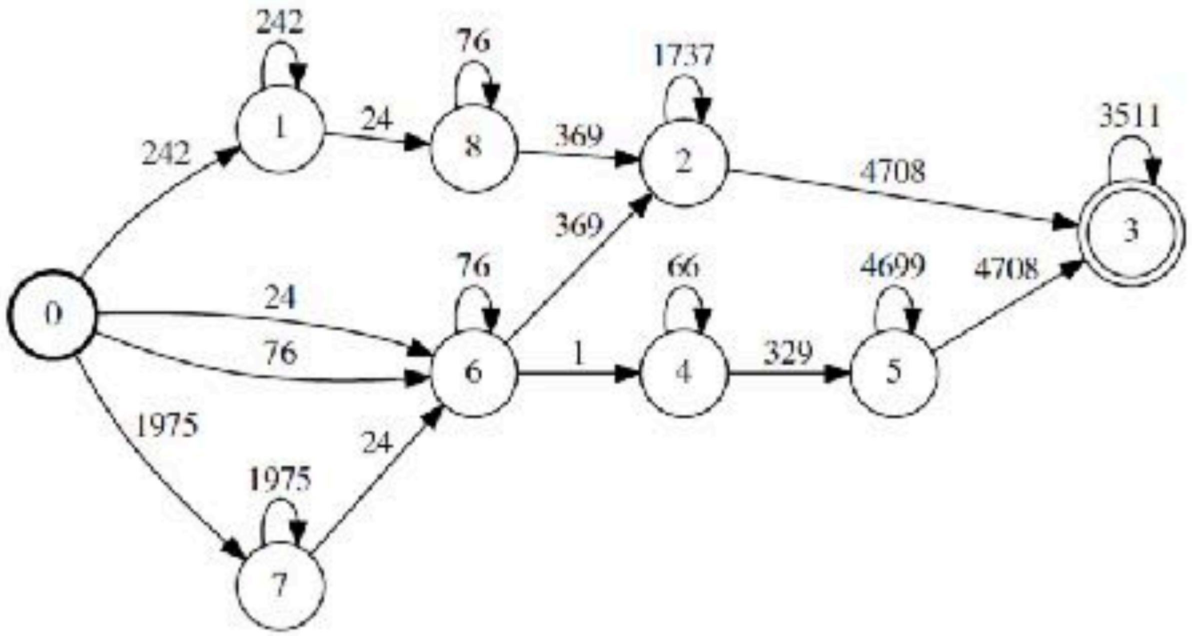
The original paper

- Used GMM system to generate lattices for training utterances, representing alternate pronunciations
- Lattice determines which phones are allowed to appear in which frames, with an additional tolerance factor
- Constraints encoded as an FST
- Compose with the normalisation FST to ensure that the logprob objective function is always < 0



Numerator FST

More recently, unconstrained numerator found to work better (Hadian, Povey et al, IEEE SLT, 2018)



2018 _ End-to-end speech recognition using lattice-free MMI - Dan Poveyhttps://www.danielpovey.com/files/2018_interspeech_end2end.pdf

Specialised forward-backward algorithm

- Work with probabilities rather than log-probabilities to avoid expensive log/exp operations
- Numeric overflow and underflow is a big problem
- Two specialisations:
  - re-normalise probabilities at every time step
  - the "leaky HMM" - gradual forgetting of context

Benefits of LF-MMI

- Models are typically faster during training and decoding than standard models
- Word error rates are generally lower
- Ability to properly compute state posterior probabilities over arbitrary state sequences also opens possibilities for
  - Semi-supervised training
  - Cross-model student-teacher training

  where sequence information is critical

LF-MMI results on Switchboard

Results on SWB portion of the Hub 5 2000 test set, trained on 300h training set. Results use speed perturbation and i-vector based speaker adaptation.

Objective	Model (size)	WER (%)
CE	TDNN-A (16.6M)	12.5
CE $\rightarrow$ sMBR	TDNN-A (16.6M)	11.4
LF-MMI	TDNN-A (9.8M)	10.7
	TDNN-B (9.9M)	10.4
	TDNN-C (11.2M)	10.2
LF-MMI $\rightarrow$ sMBR	TDNN-C (11.2M)	10.0

See Povey et al (2016) for more results

QA