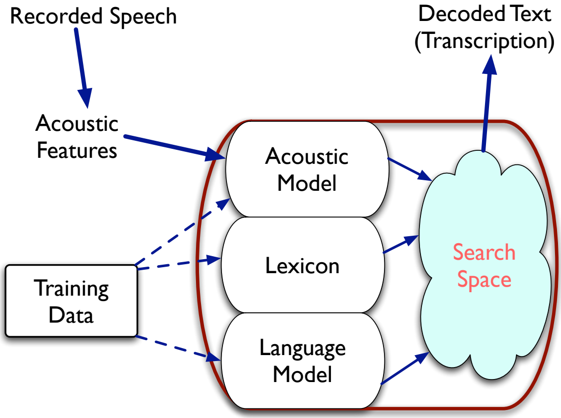
# Decoding, Alignment, and WFSTs

Steve Renals

Automatic Speech Recognition – ASR Lecture 12
26 March 2018

# HMM Speech Recognition



# The Search Problem in ASR

- Find the most probable word sequence $\hat{W} = w_1, w_2, \ldots, w_M$ given the acoustic observations $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$:

$$\begin{aligned}\hat{W} &= \arg\max_W P(W|\mathbf{X}) \\ &= \arg\max_W \underbrace{p(\mathbf{X} \mid W)}_{\text{acoustic model}} \underbrace{P(W)}_{\text{language model}}\end{aligned}$$

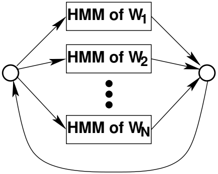
- Words are composed of state sequences so this problem corresponds to finding the most probable allowable state sequence (given the constraints of pronunciation lexicon and language model) - **Viterbi decoding**
- In a large vocabulary task evaluating all possible word sequences in infeasible (even using an efficient exact algorithm)
  - Reduce the size of the search space through pruning unlikely hypotheses
  - Eliminate repeated computations

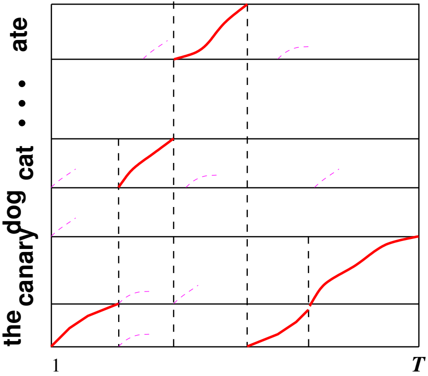
# Connected Word Recognition

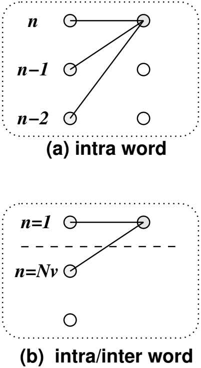
- The number of words in the utterance is not known
- Word boundaries are not known: $V$ words may potentially start at each frame

# Connected Word Recognition

- The number of words in the utterance is not known
- Word boundaries are not known: $V$ words may potentially start at each frame

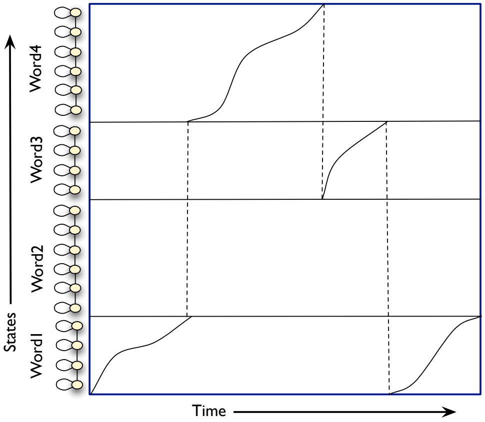


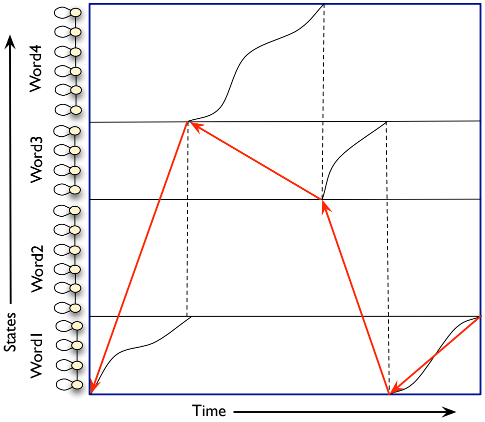




speech: "the cat ate the canary"

# Time Alignment Path

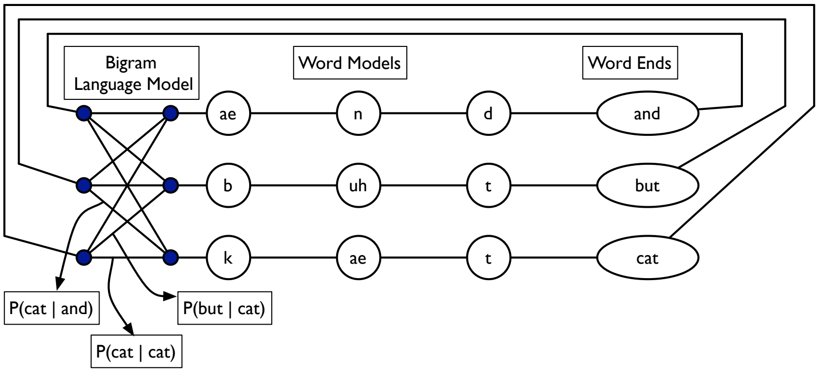


# Backtrace to Obtain Word Sequence




- Backpointer array keeps track of word sequence for a path:
  backpointer[word][wordStartFrame] = (prevWord, prevWordStartFrame)
- Backtrace through backpointer array to obtain the word sequence for a path

# Incorporating a bigram language model




Trigram or longer span models require a word history.

# Computational Issues

- Viterbi decoding performs an exact search in an efficient manner
- Exact search is not possible for large vocabulary tasks
  - Cross-word triphones need to be handled carefully since the acoustic score of a word-final phone depends on the initial phone of the next word
  - Long-span language models (eg trigrams) greatly increase the size of the search space
- Solutions:
  - Beam search (prune low probability hypotheses)
  - Dynamic search structures
  - Multipass search ($\rightarrow$ two-stage decoding)
  - Best-first search ($\rightarrow$ stack decoding / $A^*$ search)

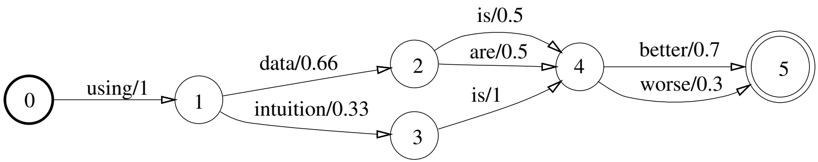
# Computational Issues

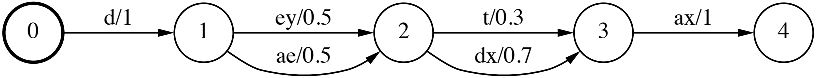
- Viterbi decoding performs an exact search in an efficient manner
- Exact search is not possible for large vocabulary tasks
  - Cross-word triphones need to be handled carefully since the acoustic score of a word-final phone depends on the initial phone of the next word
  - Long-span language models (eg trigrams) greatly increase the size of the search space
- Solutions:
  - Beam search (prune low probability hypotheses)
  - Dynamic search structures
  - Multipass search ($\rightarrow$ two-stage decoding)
  - Best-first search ($\rightarrow$ stack decoding / $A^*$ search)
- An alternative approach: Weighted Finite State Transducers (WFST)

# Weighted Finite State Transducers

- Used by Kaldi
- Weighted finite state automaton that transduces an input sequence to an output sequence (Mohri et al 2008)
- States connected by transitions. Each transition has
  - input label
  - output label
  - weight

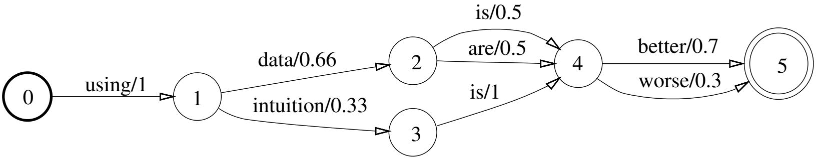
# Weighted Finite State Acceptors



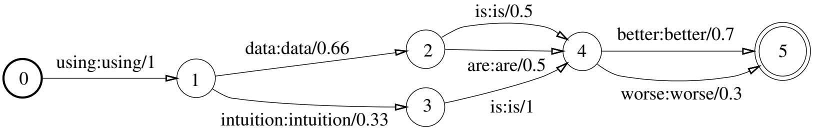



# Weighted Finite State Transducers

## Acceptor

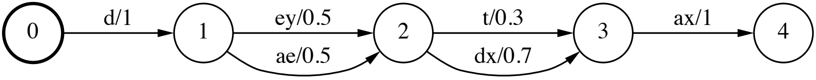



## Transducer

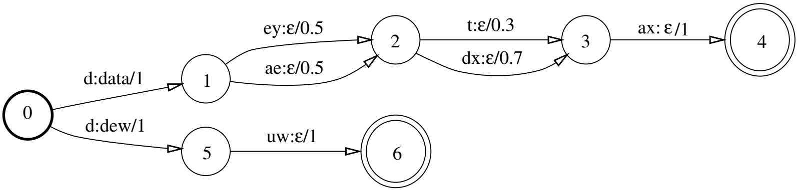


# Weighted Finite State Transducers

Acceptor




Transducer



# WFST Algorithms

**Composition** Combine transducers at different levels. For example if $G$ is a finite state grammar and $L$ is a pronunciation dictionary then $L \circ G$ transduces a phone string to word strings allowed by the grammar

**Determinisation** Ensure that each state has no more than a single output transition for a given input label

**Minimisation** transforms a transducer to an equivalent transducer with the fewest possible states and transitions

# Applying WFSTs to speech recognition

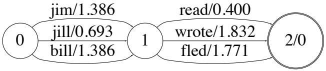
- Represent the following components as WFSTs

| | transducer | input sequence | output sequence |
|---|---|---|---|
| G | word-level grammar | words | words |
| L | pronunciation lexicon | phones | words |
| C | context-dependency | CD phones | phones |
| H | HMM | HMM states | CD phones |

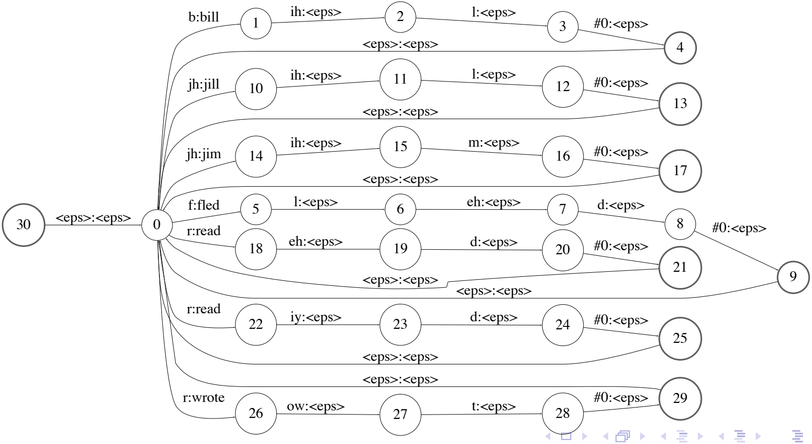
- Composing $L$ and $G$ results in a transducer $L \circ G$ that maps a phone sequence to a word sequence
- $H \circ C \circ L \circ G$ results in a transducer that maps from HMM states to a word sequence

# L, G

## G

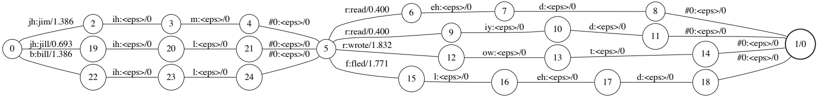

jim/1.386
jill/0.693
bill/1.386
read/0.400
wrote/1.832
fled/1.771
0
1
2/0

## L

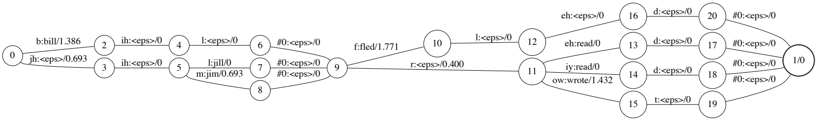

30
<eps>:<eps>
0
b:bill
1
ih:<eps>
2
l:<eps>
3
#0:<eps>
4
<eps>:<eps>
jh:jill
10
ih:<eps>
11
l:<eps>
12
#0:<eps>
13
<eps>:<eps>
jh:jim
14
ih:<eps>
15
m:<eps>
16
#0:<eps>
17
<eps>:<eps>
f:fled
5
l:<eps>
6
eh:<eps>
7
d:<eps>
8
#0:<eps>
9
r:read
18
eh:<eps>
19
d:<eps>
20
#0:<eps>
21
<eps>:<eps>
<eps>:<eps>
r:read
22
iy:<eps>
23
d:<eps>
24
#0:<eps>
25
<eps>:<eps>
<eps>:<eps>
r:wrote
26
ow:<eps>
27
t:<eps>
28
#0:<eps>
29

# $L \circ G$, det($L \circ G$), min(det($L \circ G$))

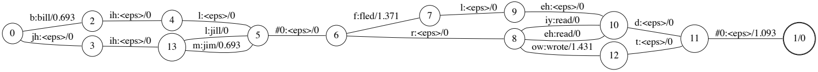
## $L \circ G$




## det($L \circ G$)

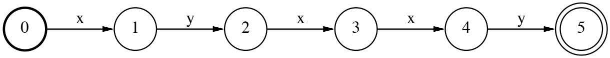



## min(det($L \circ G$))

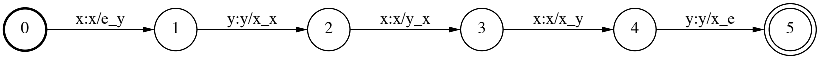


# Context dependency transducer $C$

Context-independent "string"




Context-dependency transducer (weights not shown)




(`x/e_y` – `x` with left context `e` (start/end) and right context `y`)

# Decoding using WFSTs

- We can represent the HMM acoustic model, pronunciation lexicon and n-gram language model as four transducers: H, C, L, G
- Combining the transducers gives an overall "decoding graph" for our ASR system – but minimisation and determination means it is much smaller than naively combining the transducers
- But it is important in which order the algorithms are combined otherwise the transducers may "blow-up" – basically after each composition, first determinise then minimise
- In Kaldi, ignoring one or two details

$$HCLG = \min(\det(H \circ \min(\det(C \circ \min(\det(L \circ G))))))$$

# Alignment

- Alignment is the task of matching a recording to a transcript
- In many circumstances the available transcript differs from a verbatim transcript: for example, captions/subtitles for a TV programme may not include every word spoken, or may include paraphrasing
- Performing alignment using such transcripts is of great practical use
    - time-aligning subtitles to the broadcast
    - using the data for speech recognition training (*lightly supervised training*)
- In lightly supervised training we need to use the alignment to identify reliable labels and learn from them – without also learning from unreliable labels, or past mistakes

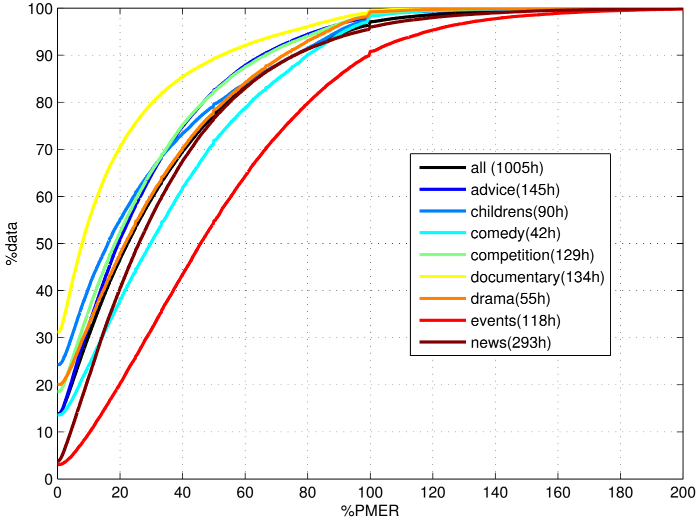
# Alignment using a biased language model

- Basic idea - transcribe the recording using a language model biased towards the transcript
  1. Train a *biased* language model on the supplied transcript, interpolated (smoothed) with a background LM

$$p(w_t|h_t) = \lambda p_{bias}(w_t|h_t) + (1 - \lambda) p_{bg}(w_t|h_t)$$

  2. Decode the training data with a pre-existing acoustic model, and the biased LM
  3. Align the captions with the ASR output
- For lightly supervised training – select utterances where there is a good match between the captions and the automatic output

# Data selection from subtitled TV recordings



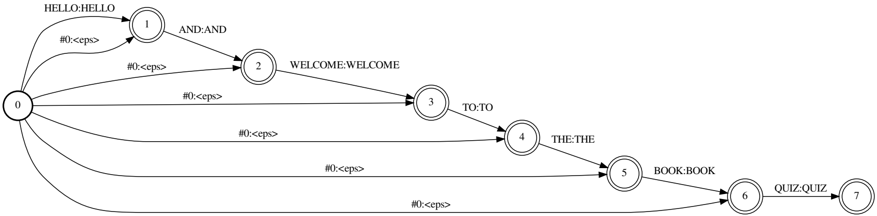

MGB Challenge 1 training data (BBC TV)

# An alternative alignment method

- The biased LM approach is quite computationally costly; it can also lead to bias towards data that we can already recognise well
- We have used an alternative approach based on constructing weighted finite state transducers for each utterance
- This allows us to use much stronger constraints – based on the captions – at decoding time

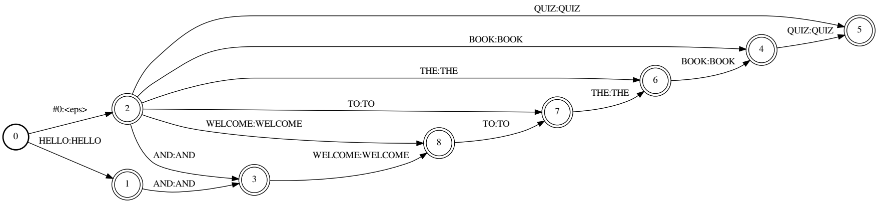
# Alignment with WFSTs

A $G$ transducer that allows any substring of the original captions – known as a *factor transducer*



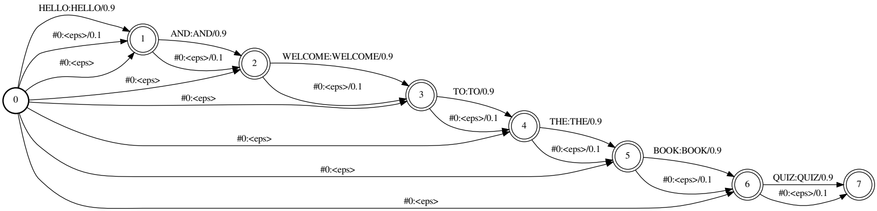
# Alignment with WFSTs

A determinized version of the $G$ transducer



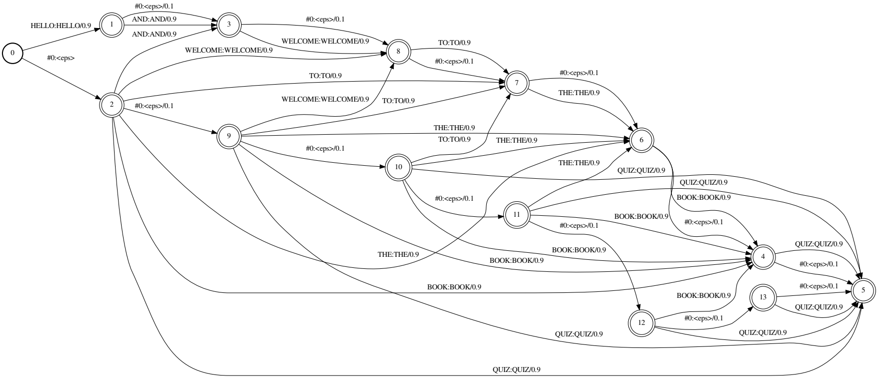
# Alignment with WFSTs

What about when word appears in the captions that was not actually spoken? We need to alter the design to be robust to this by allowing deletions (at a cost)



# Alignment with WFSTs

A determinized version



# The complete alignment process

1. Decode with a factor-transducer for the each programme
2. Align the output to the original captions
3. Re-segment the data, to potentially include missed speech
4. Decode again with utterance-specific factor transducers, allowing word-skips

# Evaluating alignment

- To evaluate we need gold-standard (verbatim) transcriptions as well as the captions to be aligned
- Evaluate the alignment of the captions with respect to a forced alignment of the gold-standard verbatim transcription
- Words spoken but not in the captions are ignored
- For words in both, systems judged correct if supplied timings are correct within a 100ms window
- Evaluated in terms of f-score

$$P = \frac{N_{match}}{N_{hyp}}, R = \frac{N_{match}}{N_{ref}}, F = 2 \times \frac{P \times R}{P + R}$$

# Alignment results on MGB

| System | Precision | Recall | F-score |
|---|---|---|---|
| Preliminary DNN AMs | | | |
| Pass 1 FT | 0.8816 | 0.7629 | 0.8180 |
| + force align | 0.8290 | 0.7855 | 0.8066 |
| Pass 2 FT+skip | 0.8679 | 0.8563 | 0.8620 |
| Final DNN AMs | | | |
| Pass 1 | 0.9009 | 0.8128 | 0.8546 |
| Pass 2 FT+skip | *0.8856* | *0.9013* | *0.8934* |

# Summary

- Search (decoding) in ASR involves finding the correct word sequence given a sample recording
- Weighted finite state transducer (WFST) framework – provides a well-justified way to combine models at different levels
- WFST algorithms - composition, determinisation, minimisation
- Kaldi represents a speech recogniser as an HCLG transducer – combining 4 transducers to map from HMM states to word sequences
- WFSTs provide a way to represent various problems in speech recognition, eg alignment

# Reading


- Mohri et al (2008). "Speech recognition with weighted finite-state transducers." In Springer Handbook of Speech Processing, pp. 559-584. Springer. http://www.cs.nyu.edu/~mohri/pub/hbka.pdf
- WFSTs in Kaldi. http://danielpovey.com/files/Lecture4.pdf
- Bell and Renals (2015), "A system for automatic alignment of broadcast media captions using weighted finite-state transducers," *ASRU*. https://doi.org/10.1109/ASRU.2015.7404861
- Moreno and Alberti (2009), "A factor automaton approach for the forced alignment of long speech recordings," *ICASSP*. https://doi.org/10.1109/ICASSP.2009.4960722
- Braunschweiler et al (2010), "Lightly supervised recognition for automatic alignment of large coherent speech recordings," *Interspeech*. http://www.isca-speech.org/archive/interspeech_2010/i10_2222.html