

Training

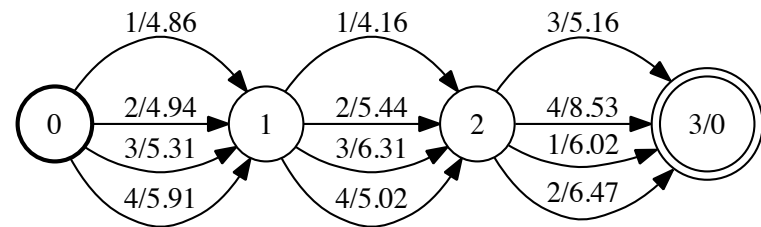


+ Lattices in Kaldi

- Representation of the alternative word-sequences that are "sufficiently likely" for a particular utterance
 - The lattice should have a path for every word sequence within α of the best-scoring one.
 - The scores and alignments in the lattice should be accurate.
 - The lattice should not contain duplicate paths with the same word sequence.
- We begin with a Weighted Finite State Transducer
$$\mathbf{HCLG} = \min(\det(\mathbf{H} \circ \mathbf{C} \circ \mathbf{L} \circ \mathbf{G}))$$
 - H: HMM
 - C: Context dependent phonemes
 - L: Lexicon
 - G: Grammar

+ WFSTs and Decoding

- An input utterance U is a set of feature vectors of length T
 - U: Utterance is a WFSA with T+1 states



- One arc for every combination time +state
- The search graph is defined as
$$S \equiv U \circ \text{HCLG}$$
 - S has approximately T+1 times as many states as HCLG
- Decoding is finding best path through S
 - In reality, searching through a subset of S that has been pruned

+ Operations on lattices

■ Pruning lattices

- Use a specified beam to remove states and arcs that are not on a path sufficiently close to the cost of the best path through the lattice.

■ Computing the best path

- outputs the corresponding input-symbol sequence (alignment) and output-symbol sequence (transcription) of the best path

■ Computing the N-best hypotheses

- Outputs a lattice with a new start state with (up to) n arcs, each starting a separate path that is within the top N scoring paths

+ Language model rescoring

- Lattice weights are a combination of language model + transition probabilities +pronunciation/silence probabilities.
- First need to subtract the original LM probabilities then add the new LM probabilities
 - `lattice-lmrescore --lm-scale=-1.0 ark:in.lats G_old.fst ark:nolm.lats`
 - `lattice-lmrescore --lm-scale=1.0 ark:nolm.lats G_new.fst ark:out.lats`
- NOTE: Lexicon has to be the same!

+ More Lattice Operations

■ Lattice Union

- Union on two lattices, most for discriminative MMI to ensure the correct transcription is in the lattice

■ Lattice Compose

- First project one set to get a word-to-word transducer, then compose
- Resulting lattice would have sum of the scores on its paths
- Uses:
 - Converting to a new lexicon (by first creating a phone lattice)
 - Adding in a “backoff” model

+ More lattice operations



- Lattice interpolation
 - Interpolates 2 lattices with a specified scaling factor
- Conversion of lattices to phones
 - Removes word labels and puts in phone labels (note: they won't be exactly time aligned)
- Computing oracle WERs from lattices
 - Takes two tables: the first of lattices, the second of transcriptions
 - Outputs the oracle word-sequence of the lattices, outputs the corresponding WERs