A Novel Discriminative Method for Pruning Pronunciation Dictionary Entries

Seppo Enarvi and Mikko Kurimo

Dept. of Signal Processing and Acoustics
School of Electrical Eng., Aalto University, Finland

One way to improve the modeling accuracy in speech recognition is by growing the dictionary...

- Non-native pronunciations
- Colloquial pronunciations
- Multiwords
- Rule-based generation
- Learning from a phone recognizer

Optimizing dictionary size

- A larger recognition vocabulary means a **higher probability of confusing** one entry with another.
- The task: prune pronunciations that are more likely to cause confusion than improve recognition.

ML pruning

- Each pronunciation variant is given a probability weight:

$$p(v_{lm}|w_l) = \frac{c(v_{lm})}{c(w_l)}$$

where v_{lm} is the pronunciation *m* of word w_l

and *c*(*v*) is the count of v in aligned training data.

- Prune the lowest probability pronunciation variants.
- Requires only forced alignments of the training data.

Minimum Classification Error

- Decoding formulated as a classification task: Assign utterance X to class W_k which has the largest discriminant function:

$$g_k(X) > g_j(X), j \neq k$$

- Misclassification measure reflects the probability that X is misrecognized:

$$d_i(X) = -g_i(X) + \log \sum_{j \neq i} \exp(g_j(X))$$

- Mapped to a smooth loss function for gradient-based optimization:

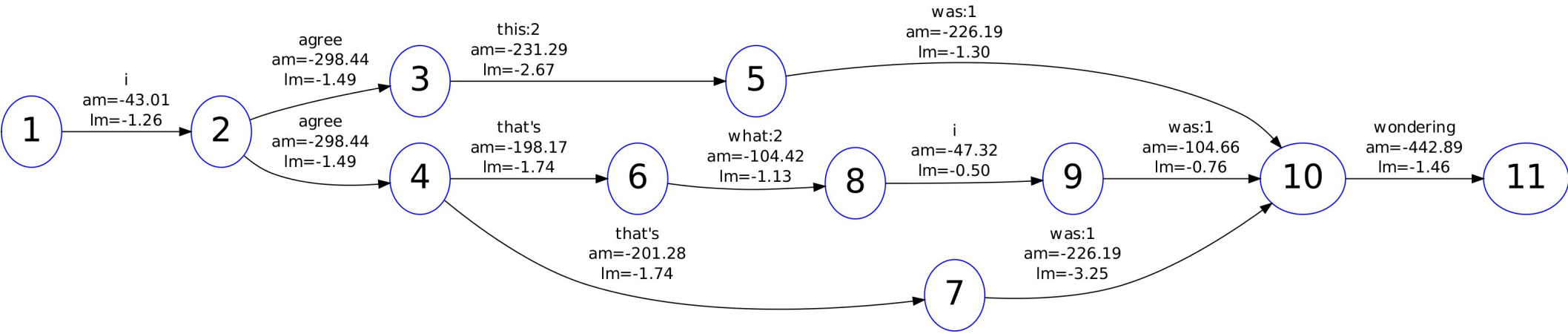
$$l_i(X) = \frac{1}{1 + \exp(-d_i(X))}$$

MCE pruning

- Greedy pruning: Compare the loss function value when an entry is included in the dictionary, and with the entry excluded.
- Requires one decoding pass to obtain a lattice or n-best list. Loss function value with different dictionaries can be computed using forced alignment.

Novel method: Pronunciation lattice

- Viterbi decoders are commonly able to produce a word lattice representing the search history.
- Include pronunciation variants as distinct arcs in the word lattice (e.g. “this:1”, “this:2”).



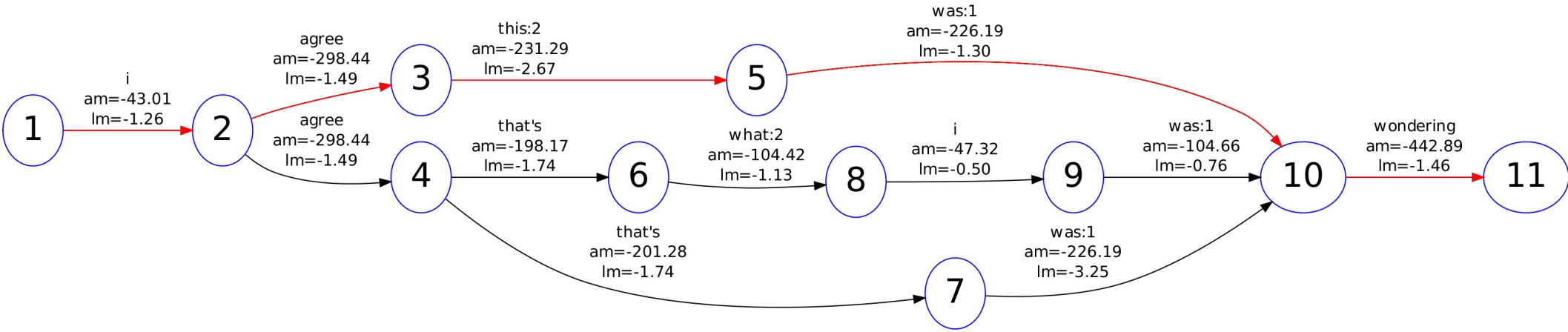
Pronunciation scoring

- Find the best path through a pronunciation lattice, and compute WER.
- For each word pronunciation in the best path, **remove the pronunciation** from the lattice and compute new WER.
- If WER increased, the word pronunciation is useful

  → increase score.
- If WER decreased, confusion introduced by including the pronunciation in the dictionary is harmful

  → decrease score.

Example utterance



- Best path through the lattice:

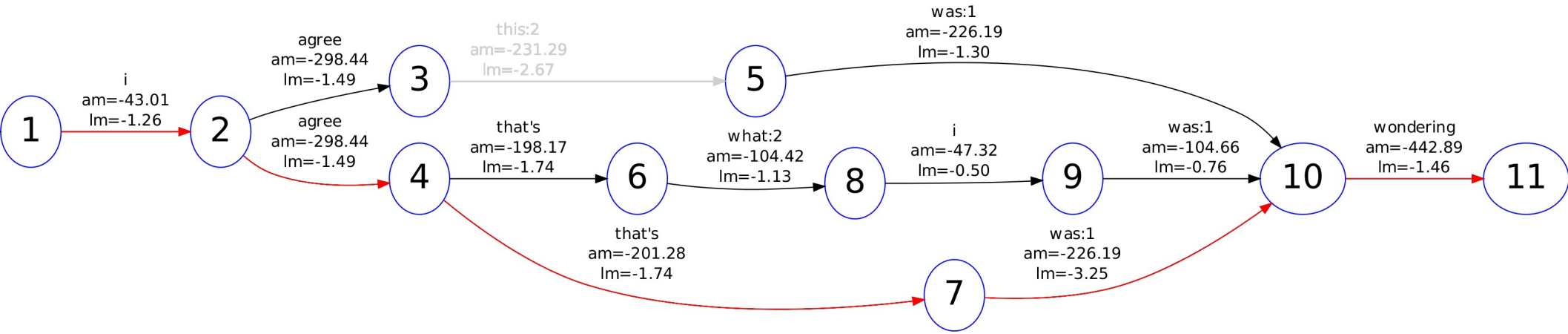
  i agree this:2 was:1 wondering

- Correct transcription:

  i agree that's what i was wondering

- Decoder has confused pronunciations *this:2* and *that's*.

Score for pronunciation *this:2*



- New best path through the lattice:

  i agree that's was:1 wondering

- Correct transcription:

  i agree that's what i was wondering

- Removal of pronunciation *this:2* has corrected one error

  → score for *this:2* will be reduced.

Experiments

- 108,000 training utterances from ICSI meeting corpus
- 84,000 word vocabulary, including multiword pronunciations added by a phonetician
- 2,000 utterance test set
- Aalto ASR system

Results

Probabilities (*	Pruning	Pruned Entries	WER
Unity	No	0	40.4 %
Unity	Discriminative	332	40.2 %
Frequency	No	0	39.9 %
Frequency	P < 0.016	193	39.9 %
Frequency	P < 0.1	2769	40.0 %
Frequency	Discriminative	190	39.8 %

*)

- Unity: Equal probability among pronunciation variants
- Frequency: Frequency in aligned training data

Conclusions

- Adding new pronunciation variants increases confusions
- A novel method to prune harmful pronunciation variants
- A discriminative way to optimize word error rate directly
- An experiment to improve a state-of-the-art English pronunciation dictionary for conversational speech recognition
- The improvements were small, probably because the dictionary was already quite good (handmade by a phonetician)
  - Might be more effective for pruning automatically added pronunciations?

Thank You!