Large-Scale Random Forest Language Models for Speech Recognition

Yi Su Fred Jelinek Sanjeev Khudanpur

Center for Language and Speech Processing
Department of Electrical and Computer Engineering
Johns Hopkins University

Aug 29, 2007 / Interspeech

Outline

Decision Tree Language Models

- Language modeling as equivalence classification of histories
- N-gram language models
  - Markovian assumption

$$P(w|h) \approx P(w|\Phi(h)) = P(w|w_{i-n+1}^{i-1}),$$

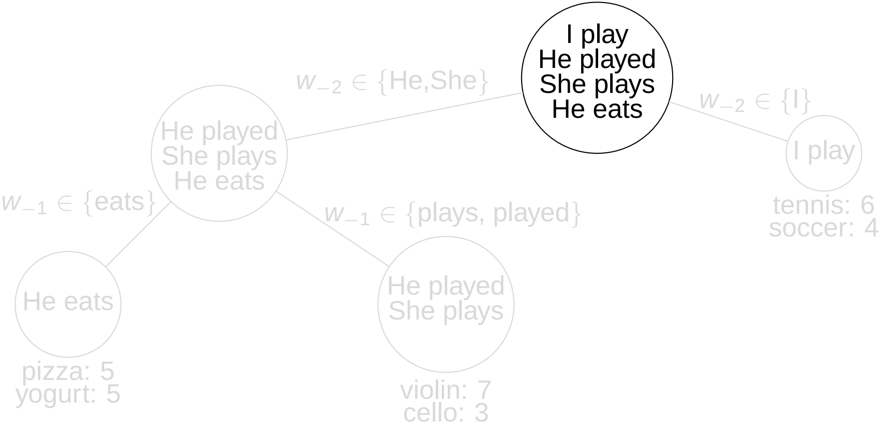
  where $h = w_1, \ldots, w_{i-1} = w_1^{i-1}$.
- Decision tree language models (Bahl et al., 1989)
  - Decision tree classifier as equivalence mapping

$$P(w|h) \approx P(w|\Phi(h)) = P(w|\Phi_{DT}(h)).$$

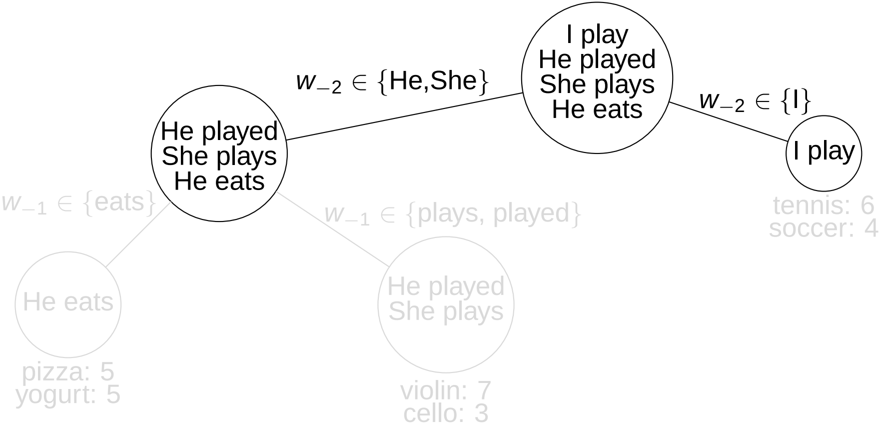
Decision Tree Training and Testing

- Growing (Top-down)
  - Start from the top node, which contains all *n*-gram histories in the training text;
  - Recursively split every node to increase the likelihood of the training text by an exchange algorithm (Martin et al., 1998);
  - Until splitting can no longer increase the likelihood.
- Pruning (Bottom-up)
  - Define the potential of a node as the gain in heldout data likelihood by growing it into a sub-tree
  - Prune away nodes whose potentials fall below a threshold.

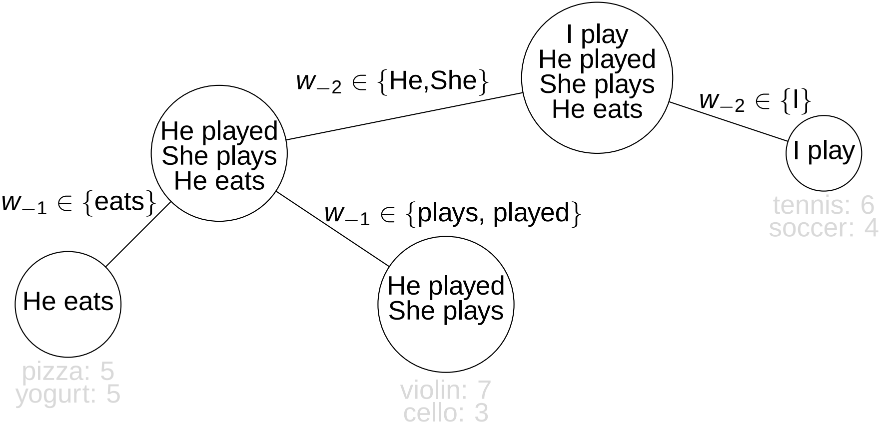
Decision Tree Language Models: Training



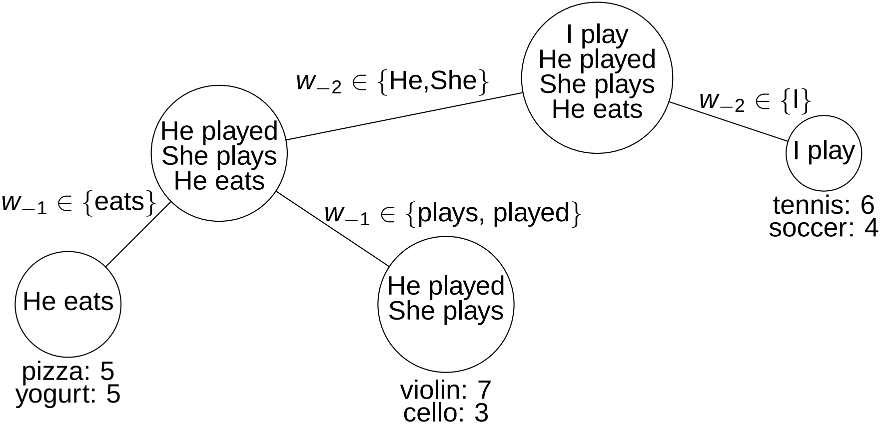
Decision Tree Language Models: Training



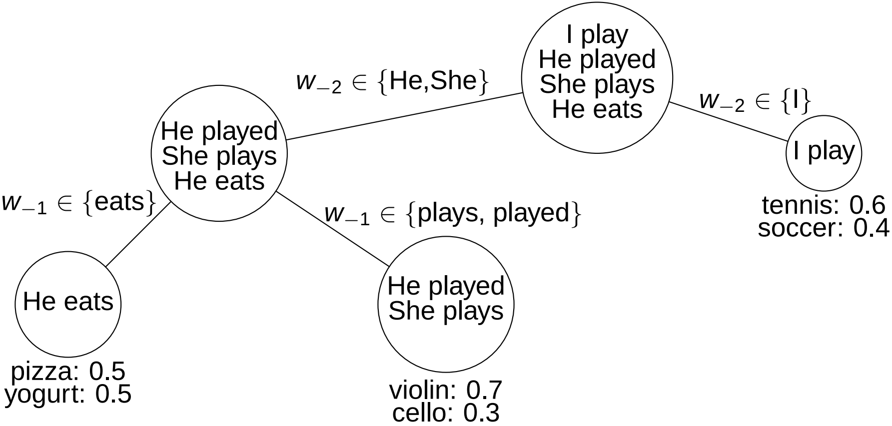
Decision Tree Language Models: Training



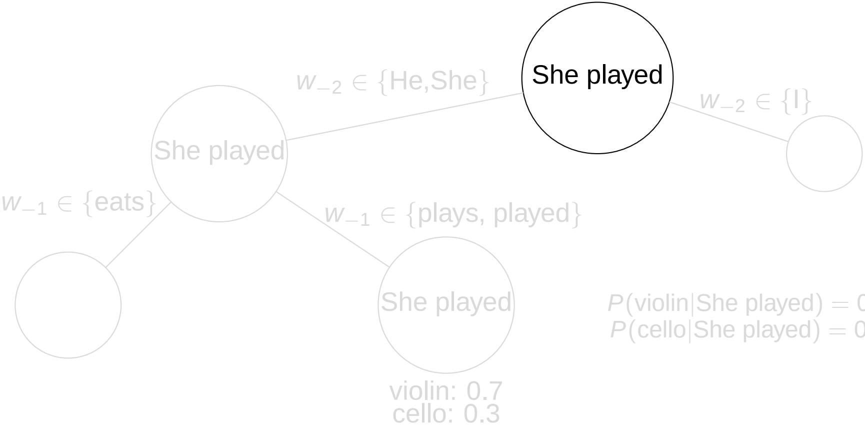
Decision Tree Language Models: Training



Decision Tree Language Models: Training

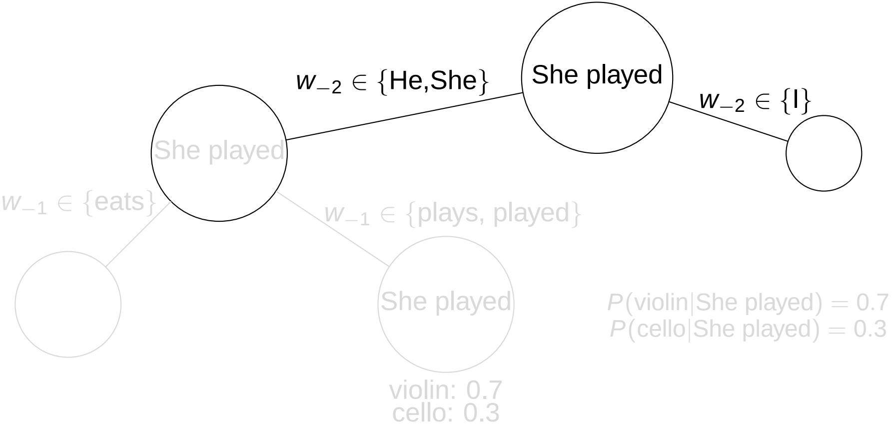


Decision Tree Language Models: Testing

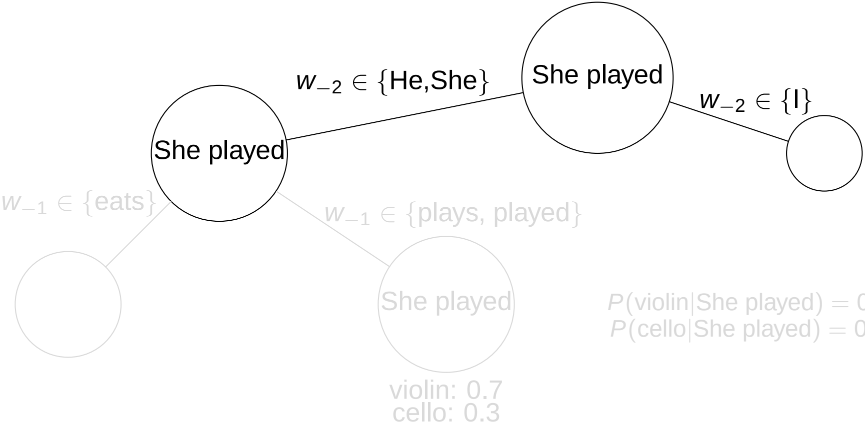


$P(\text{violin}|\text{She played}) = 0.7$
$P(\text{cello}|\text{She played}) = 0.3$

Decision Tree Language Models: Testing

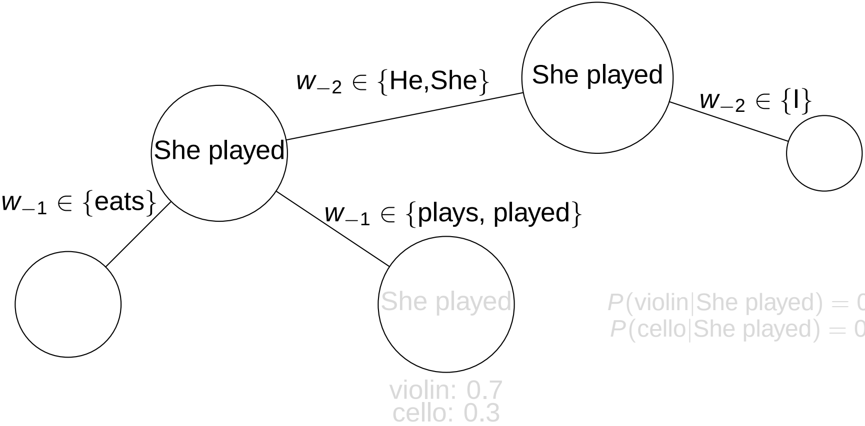


Decision Tree Language Models: Testing



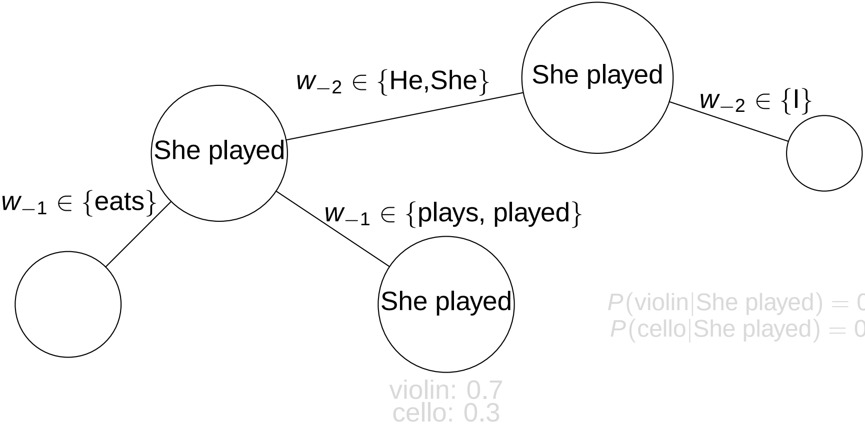
$P(\text{violin}|\text{She played}) = 0.7$
$P(\text{cello}|\text{She played}) = 0.3$

Decision Tree Language Models: Testing



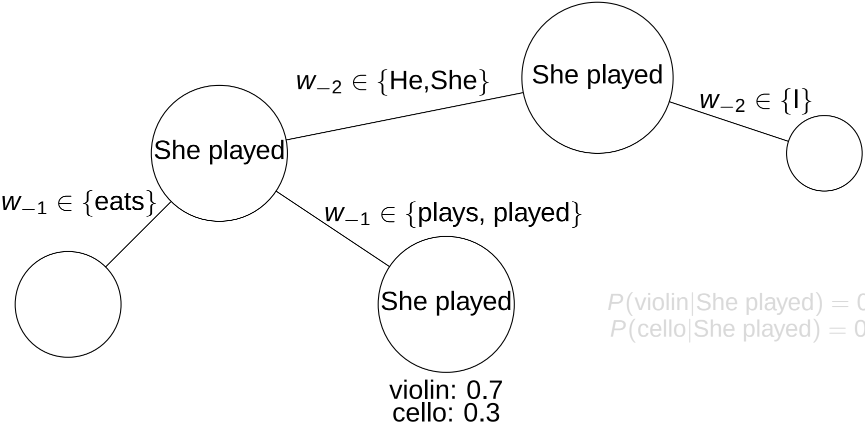
$$P(\text{violin}|\text{She played}) = 0.7$$
$$P(\text{cello}|\text{She played}) = 0.3$$

Decision Tree Language Models: Testing



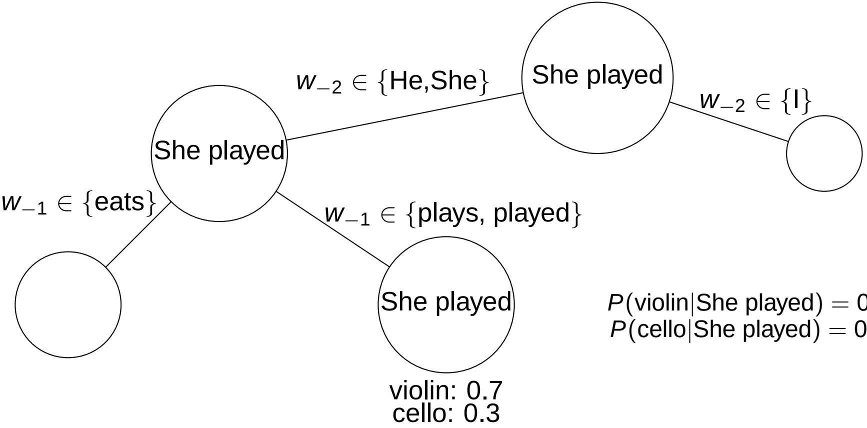
$P(\text{violin}|\text{She played}) = 0.7$
$P(\text{cello}|\text{She played}) = 0.3$

Decision Tree Language Models: Testing



$P(\text{violin}|\text{She played}) = 0.7$
$P(\text{cello}|\text{She played}) = 0.3$

Decision Tree Language Models: Testing



$$P(\text{violin}|\text{She played}) = 0.7$$
$$P(\text{cello}|\text{She played}) = 0.3$$

Decision Tree Language Models

- Failed to improve upon *n*-gram language models (Potamianos and Jelinek, 1998)
  - Without efficient search algorithm, greedy tree building procedure can't find a good tree
- Random forest (Breiman, 2001)
  - A collection of randomized decision trees
  - Final decision by voting
  - Good results in many classification tasks

Decision Tree Language Models

- Failed to improve upon *n*-gram language models (Potamianos and Jelinek, 1998)
  - Without efficient search algorithm, greedy tree building procedure can't find a good tree
- Random forest (Breiman, 2001)
  - A collection of randomized decision trees
  - Final decision by voting
  - Good results in many classification tasks

Random Forest Language Models

- A collection of randomized decision tree language models or an i.i.d. sample of decision trees (Xu and Jelinek, 2004)
- Probability via averaging

$$P(w|h) = \frac{1}{M}\sum_{j=1}^{M} P(w|\Phi_{DT_j}(h)).$$

- Superior to *n*-gram language models in terms of perplexity and word error rate on small size corpora (Xu and Mangu, 2005)

Training Randomization

- Random selection of questions
  - Set membership of a word in a history position j.

$$q_{(j,S)}(w_1^{i-1}) = \begin{cases} \text{true} & \text{if } w_j \in S; \\ \text{false} & \text{otherwise,} \end{cases}$$

    where $1 \leq j \leq i-1$ and $S \subset V$.
  - Randomly choose a subset of history positions to investigate.
- Random initialization of the exchange algorithm
  - Combat local maximum problem caused by greediness of exchange algorithm.
- Random sampling of the training data

Smoothing

- Kneser-Ney-style smoothing

$$\begin{aligned} P(w_i | w_{i-n+1}^{i-1}) &= \frac{\max(C(w_i, \Phi(w_{i-n+1}^{i-1})) - D, 0)}{C(\Phi(w_{i-n+1}^{i-1}))} \\ &+ \lambda(\Phi(w_{i-n+1}^{i-1})) P_{KN}(w_i | w_{i-n+2}^{i-1}) \end{aligned}$$

- Can be improved by Modified Kneser-Ney smoothing (Chen and Goodman, 1999)
  - Used in all experiments henceforth.

Why N-gram Language Models Work

- "There is no data like more data."
  - Performance of a statistical model depends on the amount of training data
- Simplicity implies scalability
  - N-gram language models outperform complex language models by using more data

Large-Scale Training and Testing

- Problem: Straightforward implementation quickly uses up addressable space.
  - Memory requirement grows as tree grows
- Solution: an efficient disk swapping algorithm exploiting
  - Recursive structure of binary decision tree
    - Compact representation for fast reading and writing
  - Local access property of tree-growing algorithm
    - Node-splitting depends only on the data it contains
- Achieving I/O overhead linear to the size of training *n*-gram types.

Experimental Results: Perplexity Learning Curve

- Always keeping a $>$ 10% lead over *n*-gram LM
- Which translates to significant gain in WER

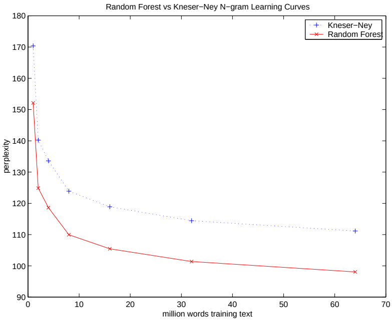


Figure: Learning curves

Experimental Results: Word Error Rate

- IBM GALE Mandarin acoustic model
  - 585 hrs, 107K vocab, PLP+VTLN+fMLLR+fMPE
- Random forest language model
  - 700M wds, 4-gram, 7∗50 trees per forest

Character Error Rate (%)	All	BN	BC
Baseline	18.9	14.2	24.8
RFLM	18.3	13.4	24.4

Table: Lattice rescoring for IBM GALE Mandarin ASR

Conclusions

- Random forest language modeling without tears.
    - Efficient disk swapping algorithm for large-scale RFLMs
    - Significant improvement in IBM GALE Mandarin system

Conclusions

- Random forest language modeling without tears.
  - Efficient disk swapping algorithm for large-scale RFLMs
  - Significant improvement in IBM GALE Mandarin system

Acknowledgments

- Many thanks to Peng Xu, Lidia Mangu, Yong Qin, Richard Sproat and Damianos Karakos!