Return to Main

Objectives

Review:
Lexical Trees
Factorization

Efficienct Search:
Memory Organization
Subtree Isomorphism
Sharing Tails
Exploiting Polymorphism

Examples:
Search Demonstration

On-Line Resources:
Odell: Context
Software Tools for NLP
Search Tool

LECTURE 38: OPTIMIZATION OF LEXICAL TREES

- Objectives:
  - Subtree Isomorphism
  - Sharing Trees
  - Polymorphism
  - Search Demo

This lecture follows the course textbook closely:

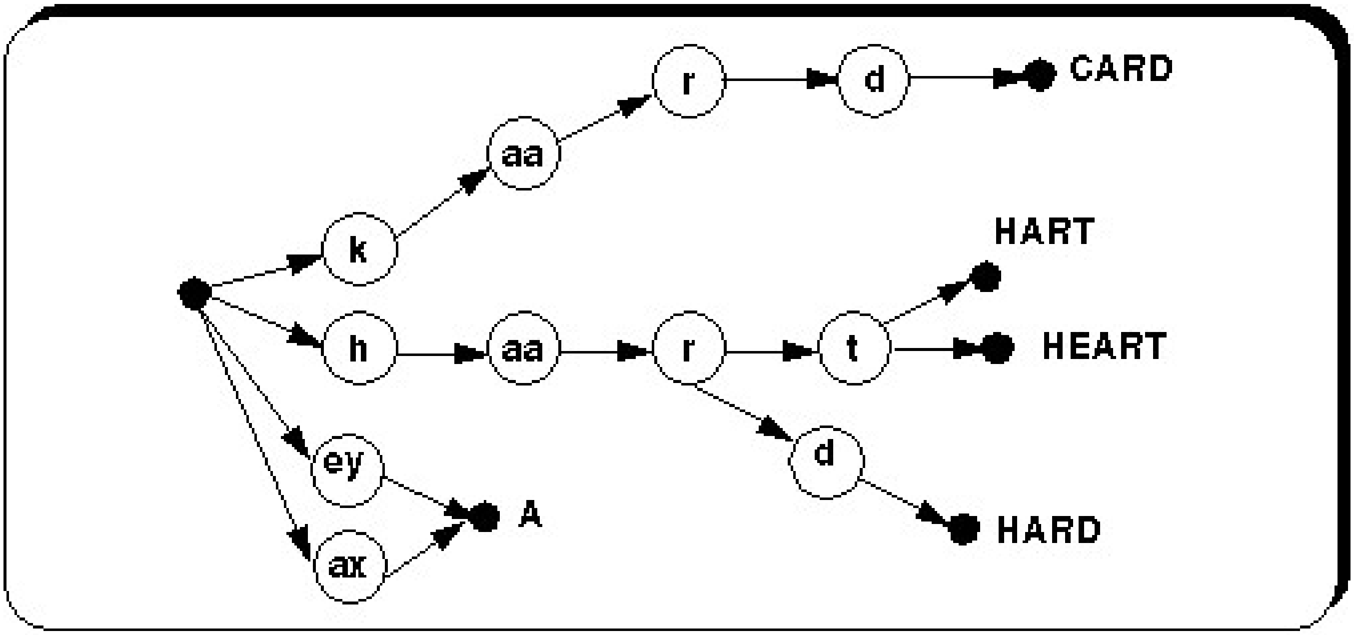
> X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development*, Prentice Hall, Upper Saddle River, New Jersey, USA, ISBN: 0-13-022616-5, 2001.

Another good source for some of this information is:

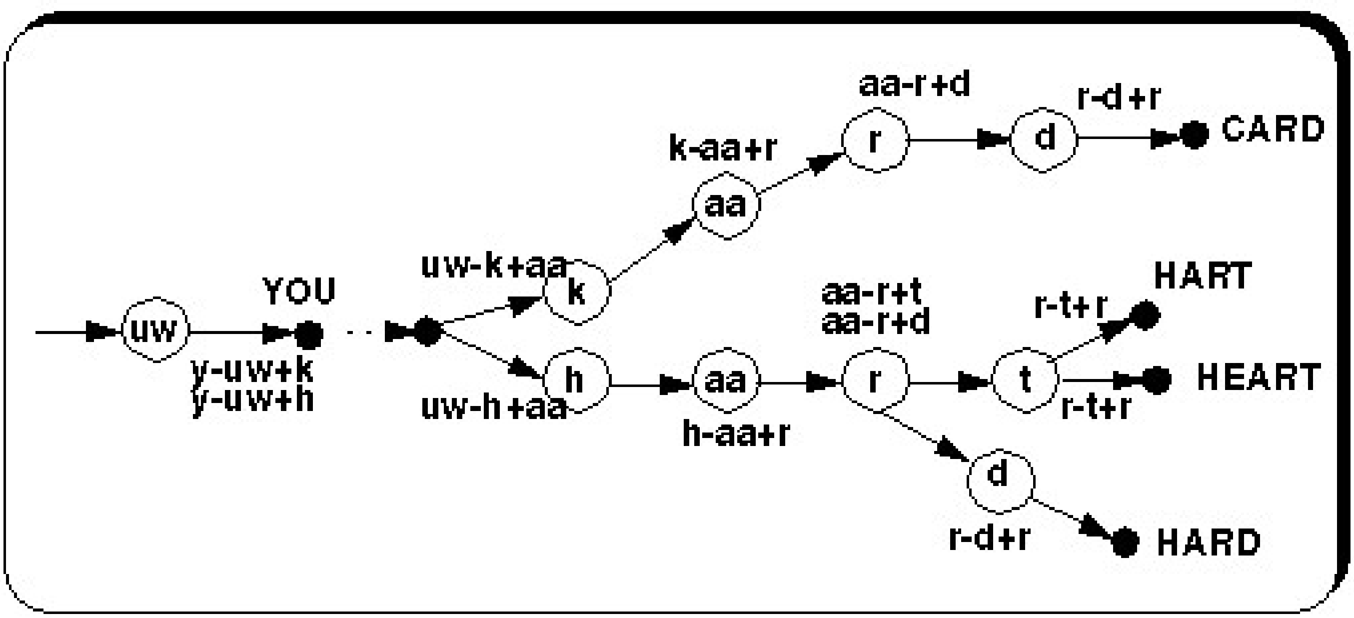
> N. Deshmukh, A. Ganapathiraju and J. Picone, "Hierarchical Search for Large Vocabulary Conversational Speech Recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 84-107, September 1999.

LEXICAL TREES

- A **lexical tree** is a data structure that allows us to share histories between words in the lexicon:



- Most systems use some form of dynamic context expansion:



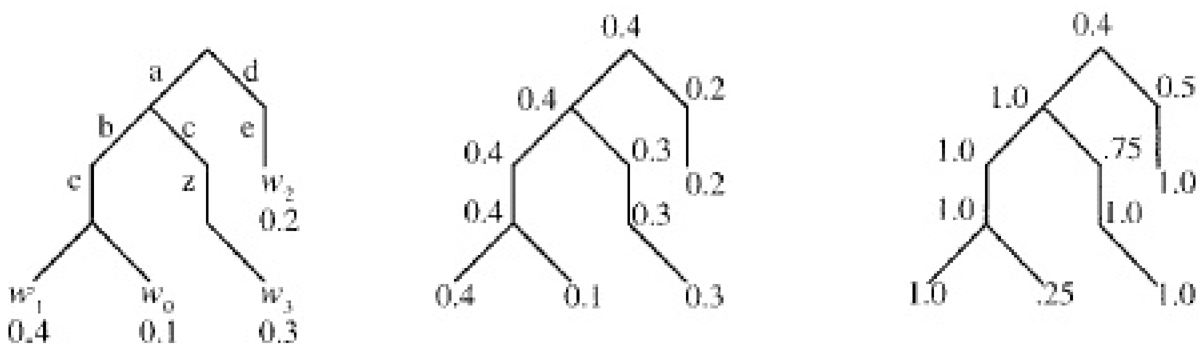
FACTORED LANGUAGE MODEL PROBABILITIES

- Premise: We can improve accuracy and minimize resources if we can apply the language model score as soon as possible.
- Factoring LM probabilities across a tree is one such idea:

$$P^*(n) = \max_{x \in child(n)} P(x)$$

$$F^*(n) = \frac{P^*(n)}{P^*(parent(n))}$$

- We can embed these probabilities in the pronunciation tree:



- In practice, such ideas require strong emphasis on the language model to be effective (such as WSJ or NAB). Applications such as conversational speech, in which acoustic ambiguity is extreme, and acoustic scores tend to dominate decoding, do not benefit as much from such approaches.

MEMORY ORGANIZATION AND EFFICIENCY

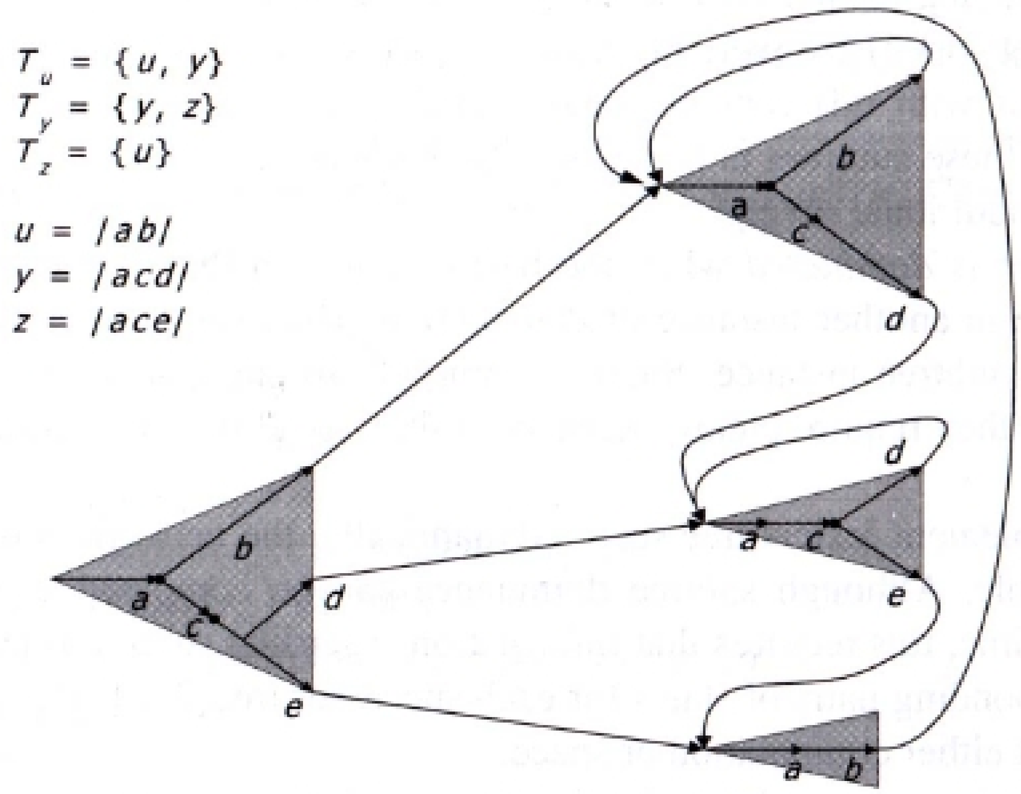
- A major drawback to the use of successor trees is the large memory overhead required.
- For example, the 1994 NAB LM contains 5M bigrams and over 70M bytes to store predecessor-dependent lexical trees.
- Need efficient ways to handle the multiple copies of lexical trees.
- Factorization of the LM scores pushes the application of probabilities earlier in the tree (before we leave the leaves and transition to the next word). Hence, we open the possibility to merge duplicated trees.
- These trees can be merged to avoid redundant state evaluation, thereby saving space and computation, with no loss of optimality.

SUBTREE ISOMORPHISM

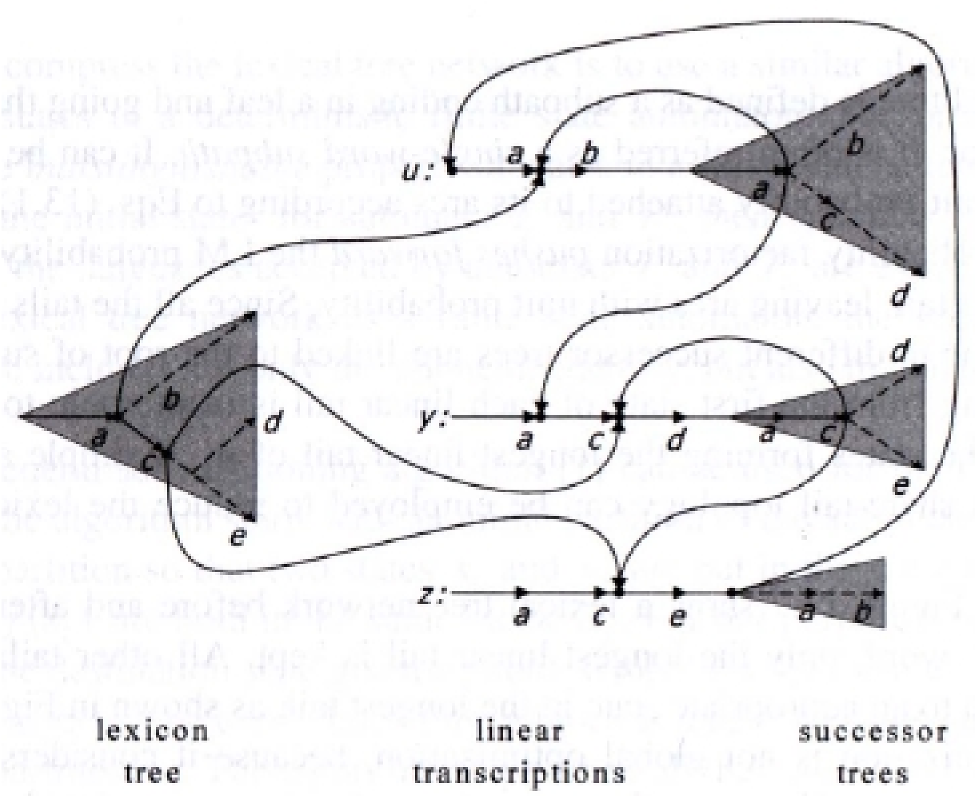
- One way we could reduce complexity is to optimize the number of states in a deterministic finite state automaton. If we exploit the tree structure of the graph, we can do more aggressive optimization.
- Two subtrees are said to be *isomorphic* to each other if they can be made equivalent by permuting the successors.
- Similarly, two states are indistinguishable if and only if their subtrees are isomorphic.
- We can merge subtrees that are isomorphic within a lexical tree. There are automated algorithms to do this.

SHARING TAILS OF TREES

- Assume a bigram language model.
- A *linear tail* in a lexical tree is defined as a subpath ending in a leaf node and going through states with a unique successor (also called a *single-word subpath*).
- LM factorization pushes forward the LM probability to the last arc of the linear tail.
- We can optimize a tree to take advantage of shared-tail optimization. Consider this tree before optimization:



and this tree after shared-tail optimization:



- What are the advantages of this approach?

EXPLOITING SUBTREE POLYMORPHISM

- The previous techniques we have discussed only eliminate identical subtrees.
- There are many subtrees that share the same nodes and topology, but have different LM scores. Can we avoid redundant acoustic model state evaluations for such trees?
- A subtree is *dominated* when the best outcome in that subtree is not better than the worst outcome in another instance of that tree. State evaluation is redundant and unnecessary for the dominated tree.
- A *polymorphic* linguistic context assignment to reduce redundancy is employed.
- Polymorphic context assignment involves keeping a single copy of the tree and allowing each state to assume the linguistic context of the most promising history (in essence, a pruning technique).
- Each node in the tree is evaluated once. However, this approach can introduce search errors.
- One approach (WHISPER) to mitigate the effects of this pruning is to keep a heap of the most promising contexts, and to delay the decision regarding which context is most promising.
- For trigram decoding, many of the approaches we have just described are still not practical.