

# Efficient Language Modelling for SCFG models

Ken Heafield

Tetsuo Kiso

Marcello Federico

Hieu Hoang

- Chart Decoder

- Expand hypotheses to the left

**Hypothesis 1:** .... he does not

**Hypothesis 1, extended:** someone said that he does not

- Optimization

- Save previous LM lookups
- Faster LM

- Recombination

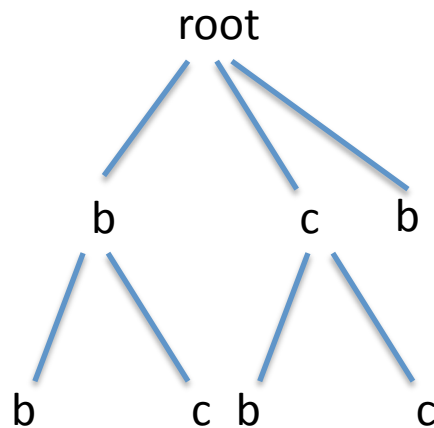
- More aggressive recombination  
→ better model score

# Optimization

Target string: b c d e

$$p(b) \times p(b \rightarrow c) \times p(bc \rightarrow d) \times p(cd \rightarrow e)$$

LM reverse trie:



# Optimization

Target string: b c d e

$p(b) \times p(b \rightarrow c) \times p(bc \rightarrow d) \times p(cd \rightarrow e)$

root

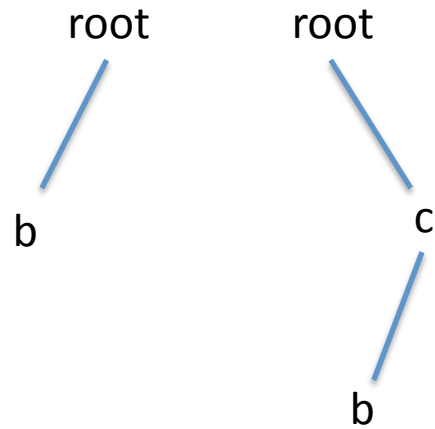
b



# Optimization

Target string: b c d e

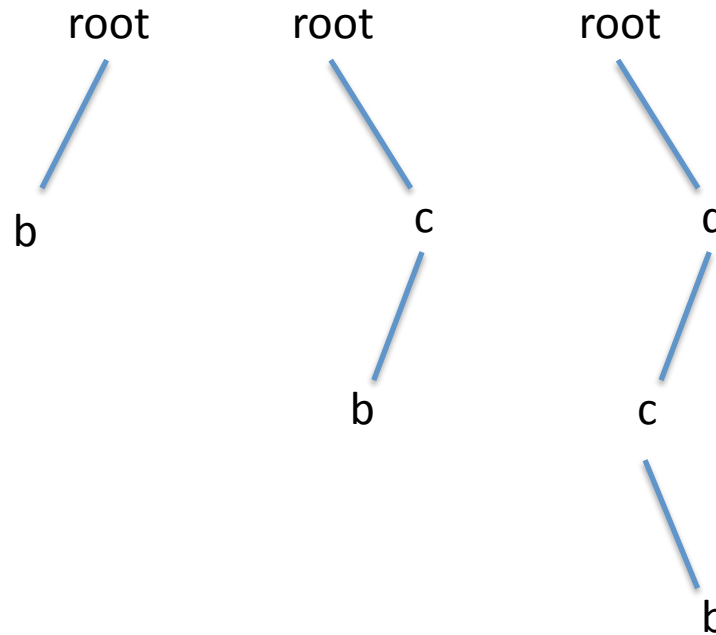
$p(b) \times p(b \rightarrow c) \times p(bc \rightarrow d) \times p(cd \rightarrow e)$



# Optimization

Target string: b c d e

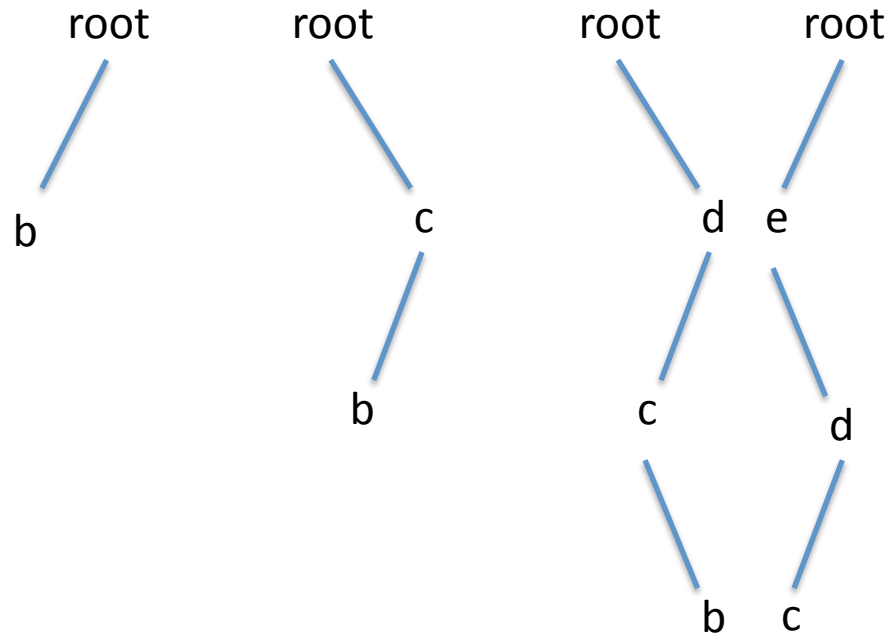
$p(b) \times p(b \rightarrow c) \times p(bc \rightarrow d) \times p(cd \rightarrow e)$



# Optimization

Target string: b c d e

$p(b) \times p(b \rightarrow c) \times p(bc \rightarrow d) \times p(cd \rightarrow e)$



# Optimization

Target string: **a** b c d e

**p(a)** x p(a→b) x p(b→c) ...

root



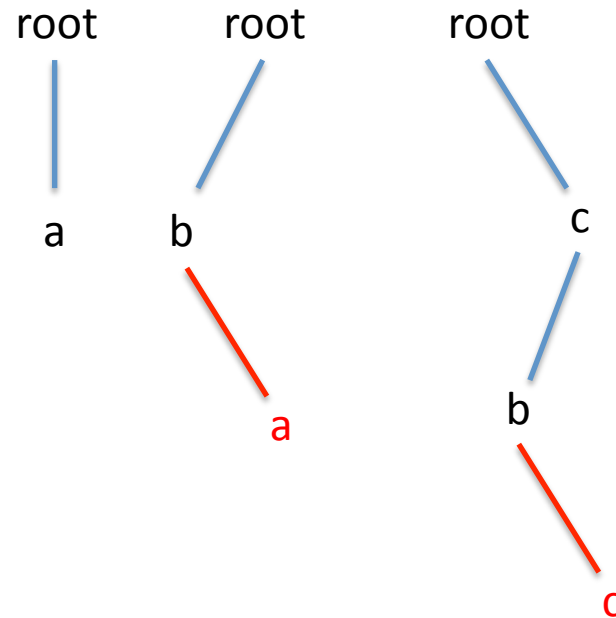
a



# Optimization

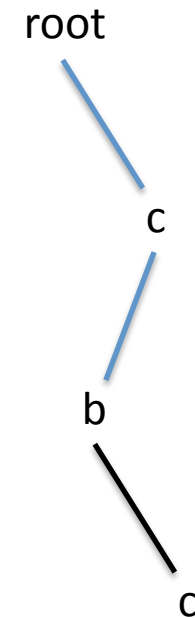
Target string: **a** b c d e

**p(a)** x p(a→b) x p(ab→c) ...



# Recombination

Target string: a b c d e     $p(a) \times p(a \rightarrow b) \times p(ab \rightarrow c) \dots$



# Changes in the decoder

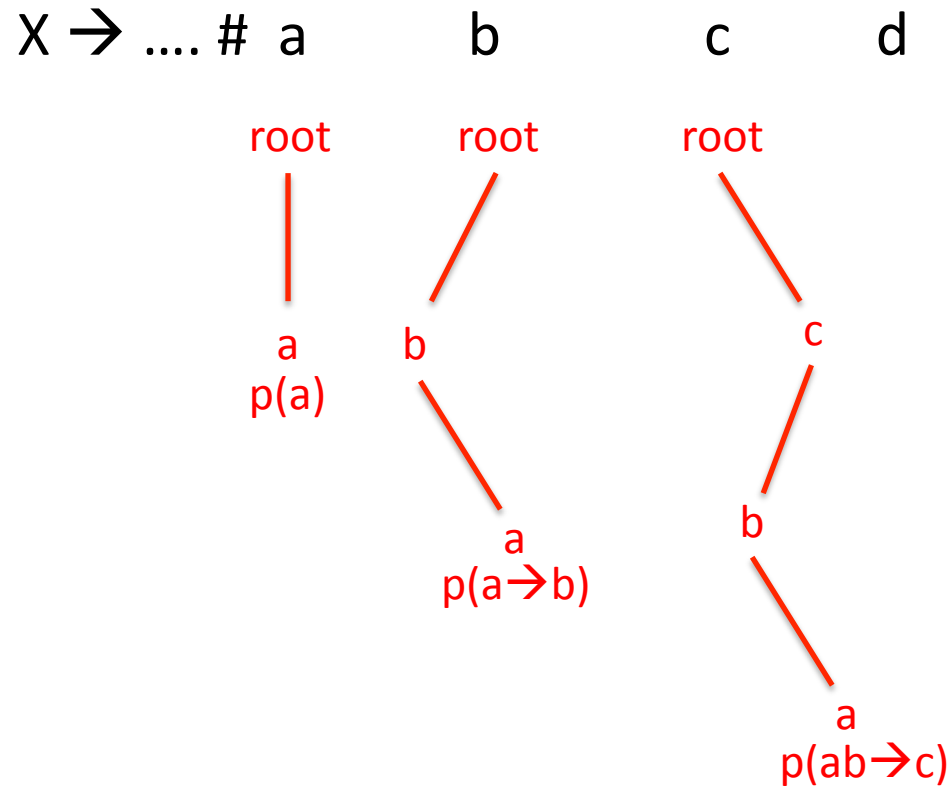
- State information
  - Hypothesis recombination
  - 1 object for each hypo, each LM
  - Same class for all LM
    - Aggressive recombination when appending words
    - Normal recombination for prepending words
      - Compares left-most (n-1) words

# Changes in the decoder

- Extension
  - Specific state information for KenLM
  - Aggressive recombination when prepending words
    - store trie-node pointers
  - Speed optimization
    - Lookup probabilities from stored trie-node pointers
- LM probability calculation
  - Original decoder
    - Generic LM calculation
    - Handled by decoder
  - New decoder
    - KenLM-specific LM calculation
    - Understands terminals / non-terminals

# Changes in KenLM

- Takes over LM scoring from decoder
  - Understand non-terminals



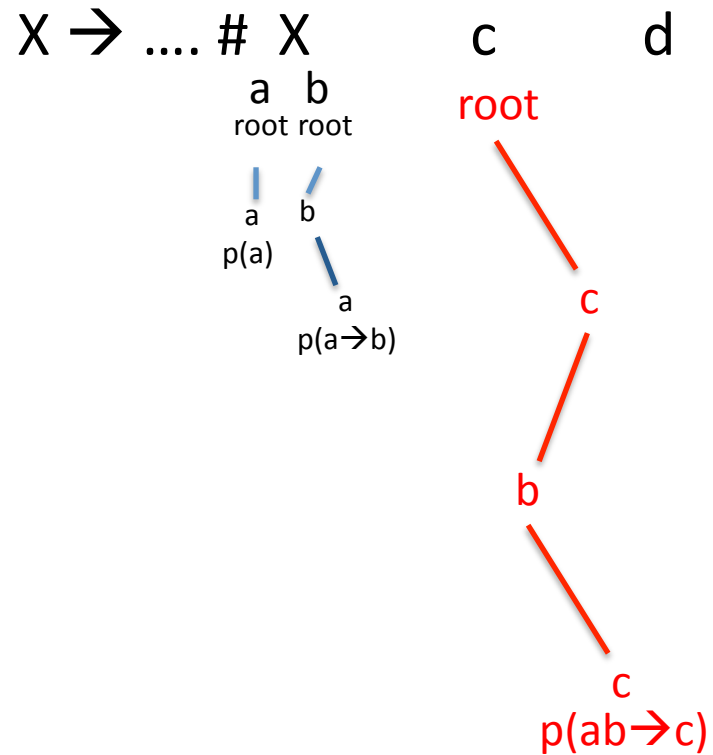
# Changes in KenLM

- Takes over LM scoring from decoder
  - Understand non-terminals

$X \rightarrow \dots \# X \quad c \quad d$   
                   $a \quad b$

# Changes in KenLM

- Takes over LM scoring from decoder
  - Understand non-terminals

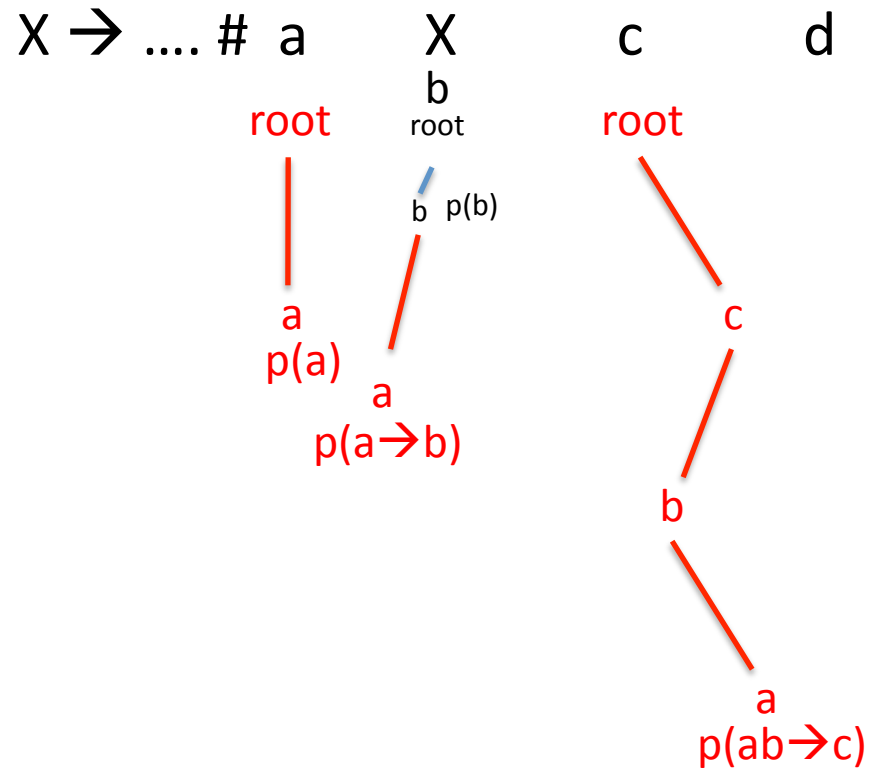






# Changes in KenLM

- Takes over LM scoring from decoder
  - Understand non-terminals



# Interface between LM and Decoder

Language model provides ChartState and RuleScore classes.

## Rule Application

1. Construct a RuleScore object
2. Decoder tells RuleScore about the RHS, going left-to-right
  - Terminal** Provide word id
  - Nonterminal** Provide probability and ChartState
3. RuleScore returns new probability and ChartState

## More information from KenLM

- ▶ Is the probability independent of more words to the left?
  - ▶ Matched a shorter  $n$ -gram than requested
  - ▶ Matched full  $n$ -gram but it won't extend left.
- ▶ Otherwise, trie pointer for further extension to the left.

## Remembering to charge backoff

A chart entry has words  $a\ b\ c\ d$ . We know  $a\ b\ c\ d$  does not extend left.

$$p(z\ a\ b\ c \rightarrow d) = p(a\ b\ c \rightarrow d) + \text{backoff}(z\ a\ b\ c)$$

### What this implies

State bit indicates if  $\text{backoff}(z\ a\ b\ c)$  should be charged.

# What's in ChartState?

- ▶ Right context: array of vocab ids and backoffs
- ▶ Left context: trie pointers for contexts that extend
- ▶ Left estimate: estimated probability of left context
- ▶ Charge backoff: flag indicates if backoff should be charged

# Status

## Done

- ▶ Changes to Moses
- ▶ More information from KenLM

## TODO

- ▶ Debug state minimization
- ▶ Use trie pointers for efficiency

# Regression test

---

- ▶ Compared Avg. model scores over 100 sentences using 5-gram LM:
  - ▶ language model scores
  - ▶ total model scores
- ▶ **Expectation: new chart decoder generates same or better scores**
- ▶ Results

Type	$\text{Score}_{\text{new}} - \text{Score}_{\text{current}}$
LM	-0.0225
Total	-0.0014