

---

# Handling Complexity of Synchronous Grammars for Machine Translation

Hao Zhang

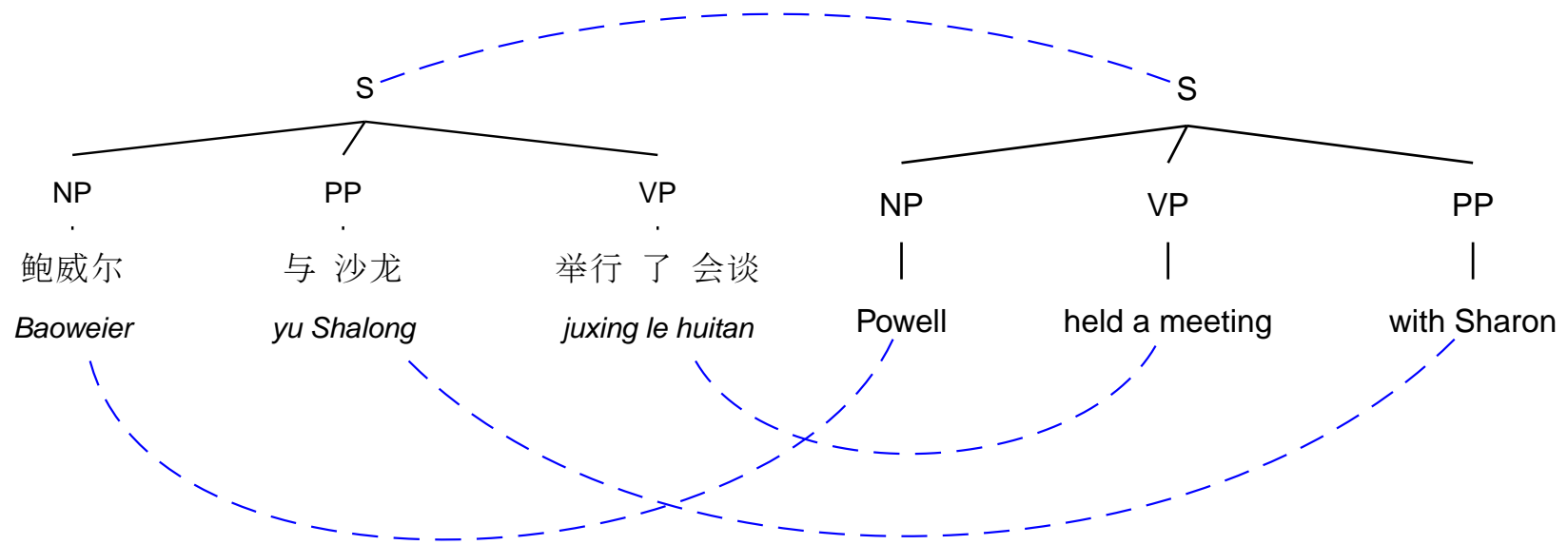
Computer Science Department  
University of Rochester

---

# Introduction

---

## Synchronous CFG Example



---

## Notions for SCFG

- A generic n-ary SCFG rule is written as

$$X \rightarrow X_1^{(1)} \dots X_n^{(n)}, X_{\pi(1)}^{(\pi(1))} \dots X_{\pi(n)}^{(\pi(n))}$$

where each  $X_i$  is a variable which can take the value of any nonterminal in the grammar.

- For example, the 3-ary rule  $S \rightarrow$

$$\left[ \begin{array}{c|c|c} & PP & \\ \hline & & VP \\ \hline NP & & \end{array} \right] \text{ can be}$$

written as

$$S \rightarrow NP^{(1)} PP^{(2)} VP^{(3)}, NP^{(1)} VP^{(3)} PP^{(2)}$$

where  $\pi = (1, 3, 2)$ .

---

## Problems in Practice

- Learning
  - Grammar induction (obtaining rules)
  - Parameter estimation (obtaining probabilities)
- Search
  - Synchronous parsing (alignment)
  - Decoding (translation)

---

# Publications

Efficient Multi-pass Decoding for Synchronous Context Free Grammars . Hao Zhang and Daniel Gildea. *In ACL-08:HLT*.

Bayesian Learning of Non-compositional Phrases with Synchronous Parsing . Hao Zhang, Chris Quirk, Robert C. Moore and Daniel Gildea. *In ACL-08:HLT*.

Enumeration of Factorizable Multi-Dimensional Permutations . Hao Zhang and Daniel Gildea. *Journal of Integer Sequences*, Article 07.5.8, 2007. (Contributed Integer Sequence A133262 in the On-line Encyclopedia of Integer Sequences)

Factorization of Synchronous Context-Free Grammars in Linear Time . Hao Zhang and Daniel Gildea. *In NAACL Workshop on Syntax and Structure in Statistical Translation (SSST)*, 2007.

Efficient Search for Inversion Transduction Grammar . Hao Zhang and Daniel Gildea. *In EMNLP-06*.

Factoring Synchronous Grammars by Sorting . Daniel Gildea, Giorgio Satta, and Hao Zhang. *In COLING/ACL-06*.

Inducing Word Alignments with Bilexical Synchronous Trees . Hao Zhang and Daniel Gildea. *In COLING/ACL-06*.

Synchronous Binarization for Machine Translation . Hao Zhang, Liang Huang, Daniel Gildea and Kevin Knight. *In HLT/NAACL-06*.

Machine Translation as Lexicalized Parsing with Hooks . Liang Huang, Hao Zhang and Daniel Gildea. *In Proceedings of the 9th International Workshop on Parsing Technologies (IWPT-05)*.

Stochastic Lexicalized Inversion Transduction Grammar for Alignment . Hao Zhang and Daniel Gildea. *In ACL-05*.

Syntax-Based Alignment: Supervised or Unsupervised? . Hao Zhang and Daniel Gildea. *In COLING-04*.

---

## Outline

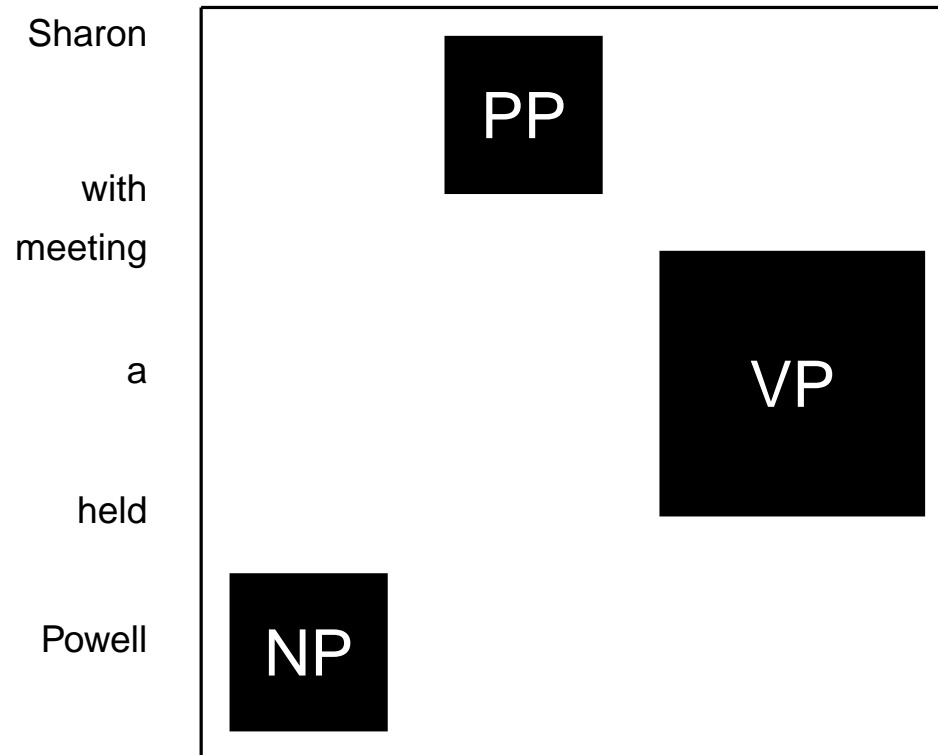
- **Part I**: Reducing SCFG Complexity through Factorization
- **Part II**: Efficient Multi-pass Decoding for SCFG
- **Part III**: Bayesian Learning of Phrases with Synchronous Parsing

---

# Part I: Grammar Factorization

---

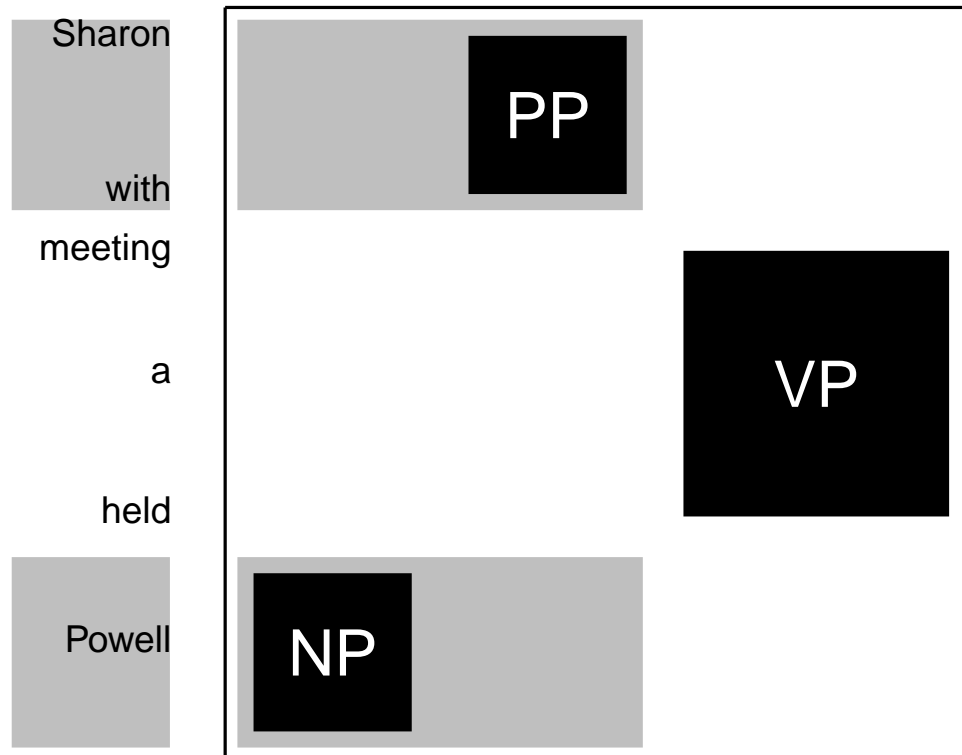
## Synchronous Binarization



鲍威尔      与 沙龙      举行 了 会谈

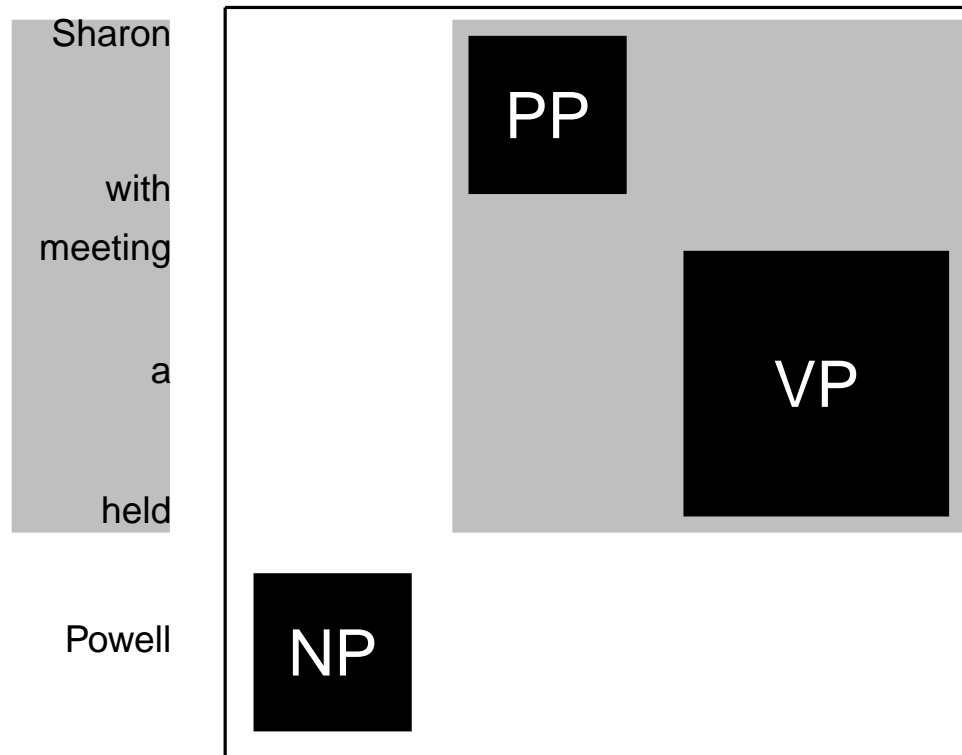
*Baoweier      yu Shalong      juxing le huitan*

## Synchronous Binarization



鲍威尔 与 沙龙 举行 了 会谈  
*Baoweier yu Shalong juxing le huitan*

## Synchronous Binarization



鲍威尔      与 沙龙      举行 了 会谈  
*Baoweier*      *yu Shalong*      *juxing le huitan*

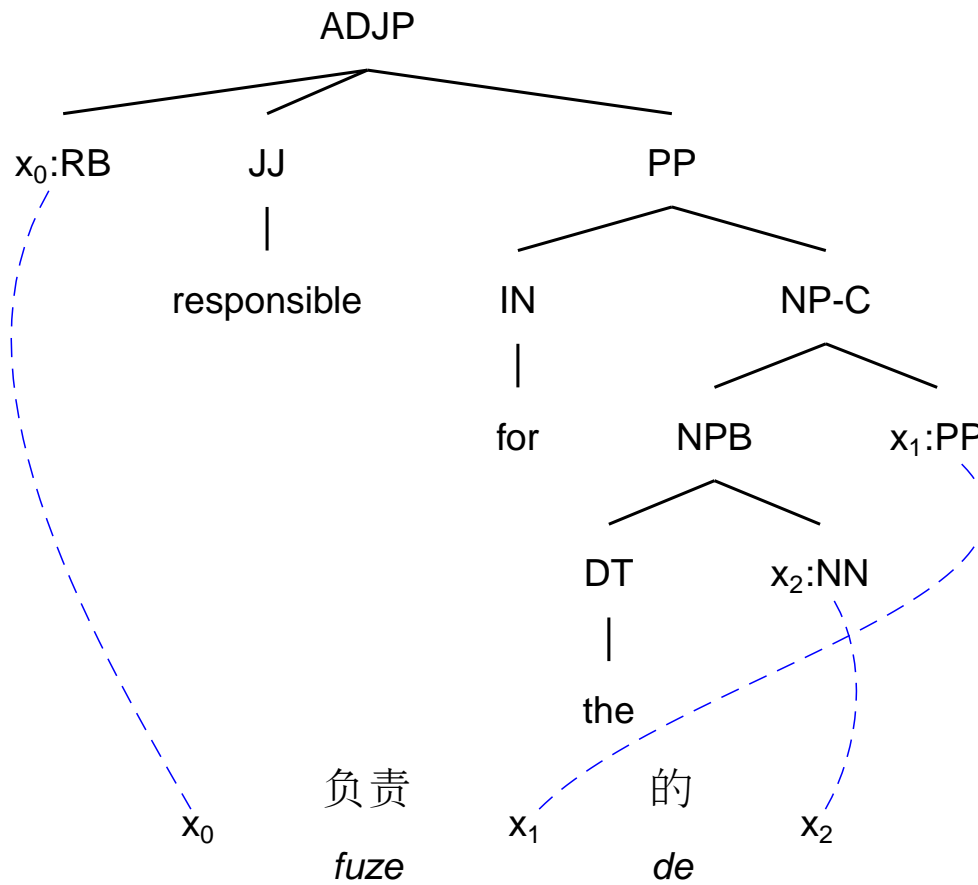
---

## Parsing Complexity of n-ary SCFG

Assume the sentence length (on either side) is  $O(|w|)$ .

- $O(|w|^{2n+2})$ , without any factorization.
- $O(|w|^{n+1+3}) = O(|w|^{n+4})$ , with binarization on one side.
- $O(|w|^{3+3}) = O(|w|^6)$ , with synchronous binarization maintaining continuous spans on both sides.

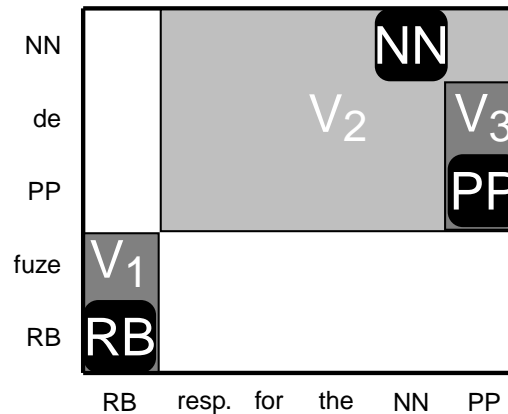
## Example from a Real System



(Galley et al., 2006)

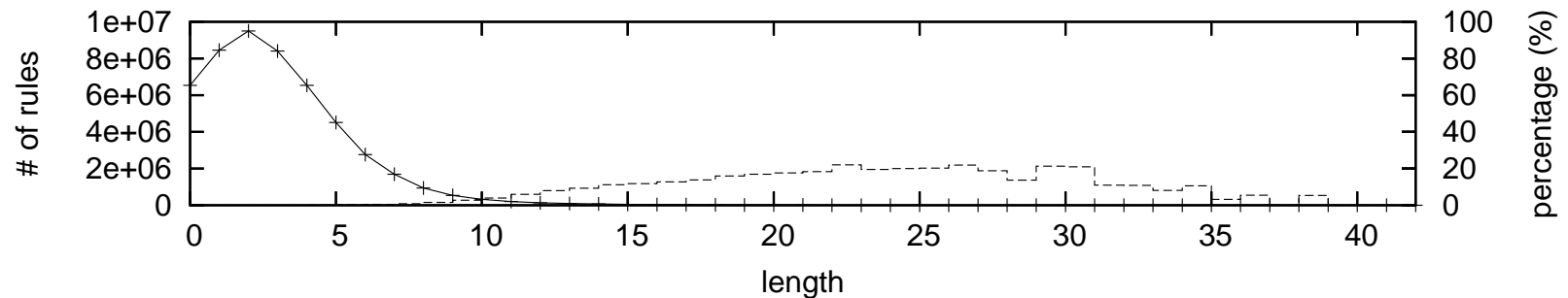
## Binarization Example for Tree Transducer

ADJP	→ T <sup>(1)</sup> ,	T <sup>(1)</sup>
T	→ RB <sup>(1)</sup> <i>fuze</i> PP <sup>(2)</sup> <i>de</i> NN <sup>(3)</sup> ,	RB <sup>(1)</sup> resp. for the NN <sup>(3)</sup> PP <sup>(2)</sup>
ADJP	→ T <sup>(1)</sup> ,	T <sup>(1)</sup>
T	→ V <sub>1</sub> <sup>(1)</sup> V <sub>2</sub> <sup>(2)</sup> ,	V <sub>1</sub> <sup>(1)</sup> V <sub>2</sub> <sup>(2)</sup>
V <sub>1</sub>	→ RB <sup>(1)</sup> ,	RB <sup>(1)</sup> <i>fuze</i>
V <sub>2</sub>	→ resp. for the NN <sup>(1)</sup> V <sub>3</sub> <sup>(2)</sup> ,	V <sub>3</sub> <sup>(2)</sup> NN <sup>(1)</sup>
V <sub>3</sub>	→ PP <sup>(1)</sup> ,	PP <sup>(1)</sup> <i>de</i>



---

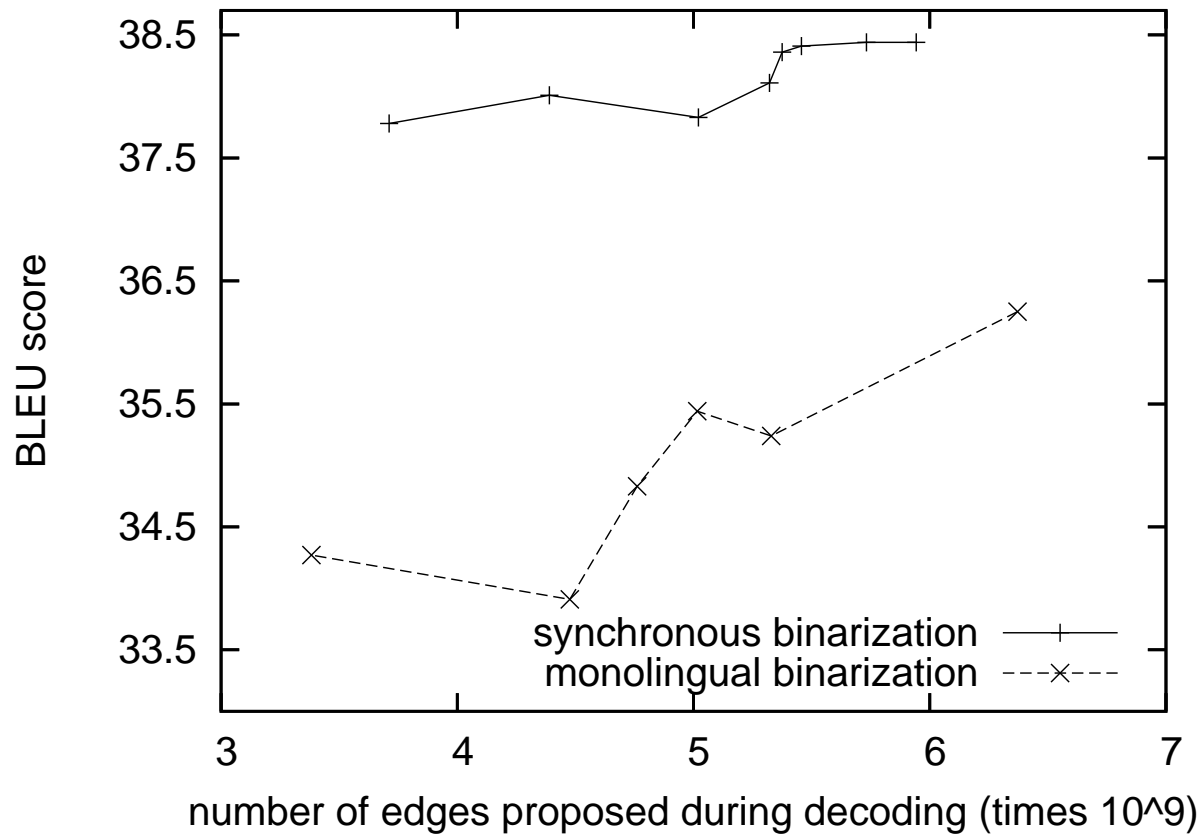
## How Many Rules Are Binarizable?



- 50,879,242 rules, among which 99.7% are binarizable.
- The solid-line: the distribution of all rules against permutation lengths.
- The dashed-line: the percentage of non-binarizable rules at varying lengths.

---

## Faster and More Accurate Decoding



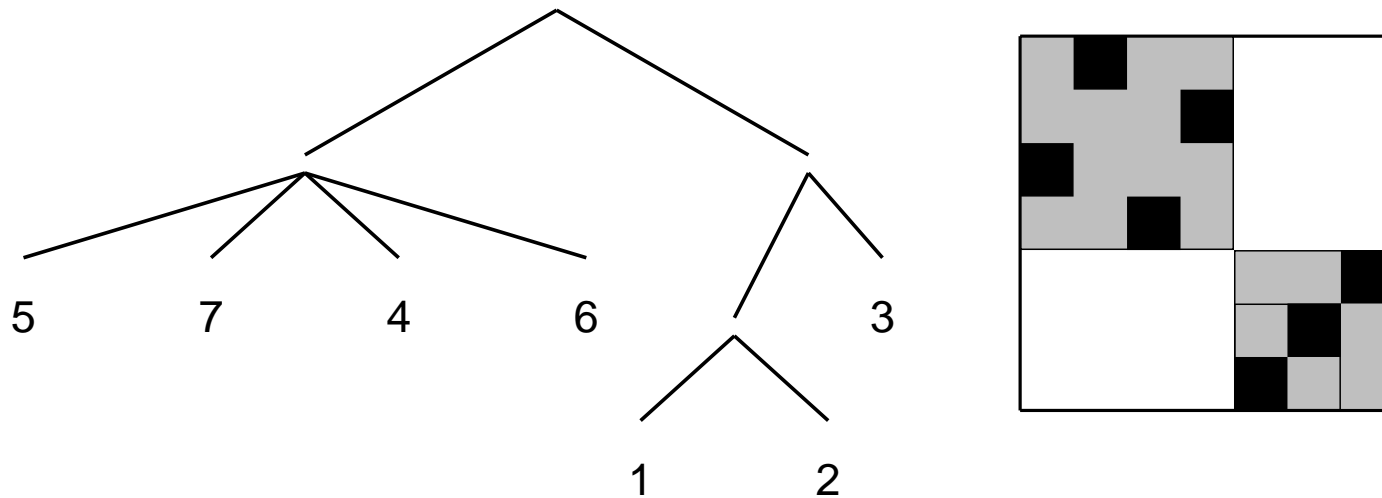
(Zhang, Huang, Gildea and Knight, 2006)

The synchronous binarizer is a part of the ISI system which ranked top among all participants in the Chinese-to-English track of the 2006 NIST Machine Translation Evaluation.

---

## Beyond Binarizability

For example, (5, 7, 4, 6, 1, 2, 3) is decomposable:



So we can reduce long SCFG rules into shorter ones. If the longest SCFG rule after such factorization is  $k$ , we can parse sentences with the grammar in  $O(|w|^{k+4})$ .

---

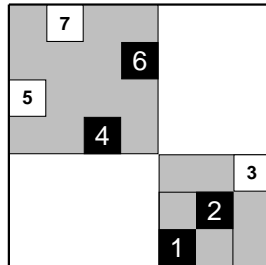
## Asymptotic Results

- $H_k(n)$ , the number of  $(n - k)$ -ary permutations, grows factorially.  
$$H_k(n)/n! \approx \frac{2^k}{e^2 \cdot k!}.$$
  - Percentages: 13.5%, 27.1%, 27.1%, 18.0%, 9.0%, 3.6%, . . .
- The number of  $k$ -ary permutations grows roughly exponentially.
  - Bases: 5.83, 5.83, 6.87, 7.33, 7.82, 8.26 . . .

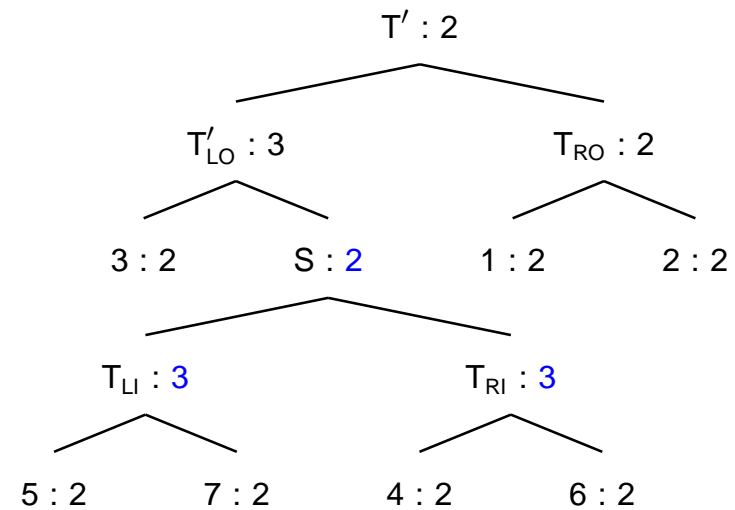
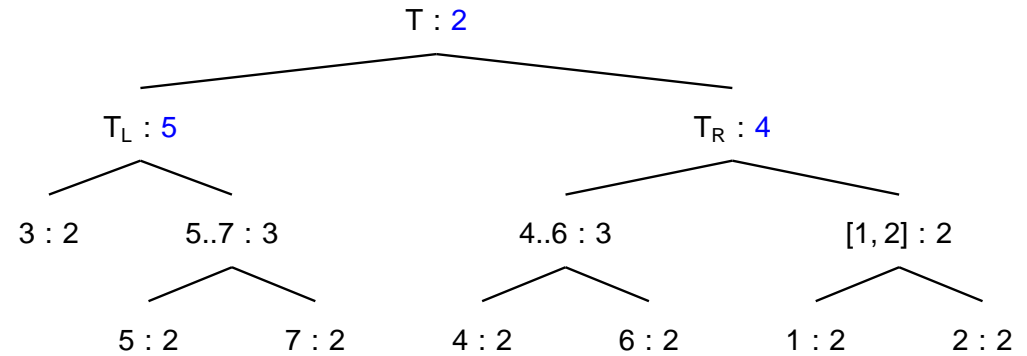
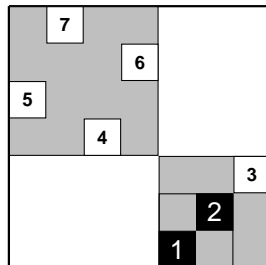
(Zhang and Gildea, 2007)

# K-arization and Optimal Parsing

$O(|w|^{11})$



$O(|w|^8)$



---

## Part II: Efficient Decoding

---

## Language Model Integrated Decoding for SCFG

- LM-integrated states are represented as  $X[i, j, u_{1,\dots,n-1}, v_{1,\dots,n-1}]$ .
  - A trigram state:  $X[3, 6, \text{held}, a, a, \text{meeting}]$
  - A bigram state:  $X[3, 6, \text{held}, \text{meeting}]$
- Complexity of decoding is determined by the number of variables at each DP step.
  - $X_1[i, j, u, v] + X_2[j, k, u', v']$
- $O(|w|^{3+4(n-1)})$  for binary SCFG with n-gram language model.
- Can be optimized to  $O(|w|^{3+3(n-1)})$ , using the “hook” trick.  
(Huang, Zhang and Gildea, 2005)

---

## Progressive LM-Integrated Decoding

- A bigram-integrated state  $[X, i, j, u, v]$  is said to be a coarse-level state of a trigram-integrated state  $[X, i, j, u, u', v', v]$ .
- Gradually augment the LM-integrated states from lower orders to higher orders.
- Coarse-to-fine decoding in the  $A^*$  framework.
- The key: using the actual outside cost of a bigram state as a heuristic estimate of the outside cost of a refined trigram state.

---

## Inside/Outside Parsing for Coarse-level Decoding

- An algorithm similar to inside/outside parsing is applied to get the outside costs of the coarse-level LM-integrated states.
- Bottom-up first, then top-down.
- Many algorithmic choices for the coarse-level bottom-up pass, such as CKY, agenda-based.

---

## Heuristics for Fine-grained Decoding

We prioritize decoding states according to

$$\begin{aligned} & \beta(X[i, j, u_1, u_2, v_1, v_2]) \\ & \quad + \alpha(X[i, j, u_1, v_2]) \\ & \quad + h_{\text{BestBorder}}(u_1, u_2, v_1, v_2) \end{aligned}$$

where

$$\begin{aligned} & h_{\text{BestBorder}}(u_1, u_2, v_1, v_2) \\ & = \left[ \max_s P_{\text{Im}}(u_2 \mid s, u_1) \right] \\ & \quad \cdot \left[ \max_s P_{\text{Im}}(s \mid v_1, v_2) \right] \end{aligned}$$

---

## Experimental Setup

- Data set: LDC 2002 MT evaluation data set, with a length limit of 20 on the Chinese sentences. 371 sentences, each having 10 references.
- We vary the decoding strategies and beam settings and keep the model unchanged in the experiments.

---

## Decoding Strategies

- *CYK*: standard bottom-up ITG decoder (Wu, 1996; Chiang, 2005), with standard beam pruning.
- *Agenda*: agenda-based ITG decoder (Zhang and Gildea, 2006), with standard beam pruning.
- *Lazy\_kbest*: top-down ITG decoder, treating beam pruning as k-best selection (Huang and Chiang, 2007).
- *Bitri\_cyk*: two-pass ITG decoder, first pass as *cyk* with beam pruning, second pass as *agenda* (Zhang and Gildea, 2008).

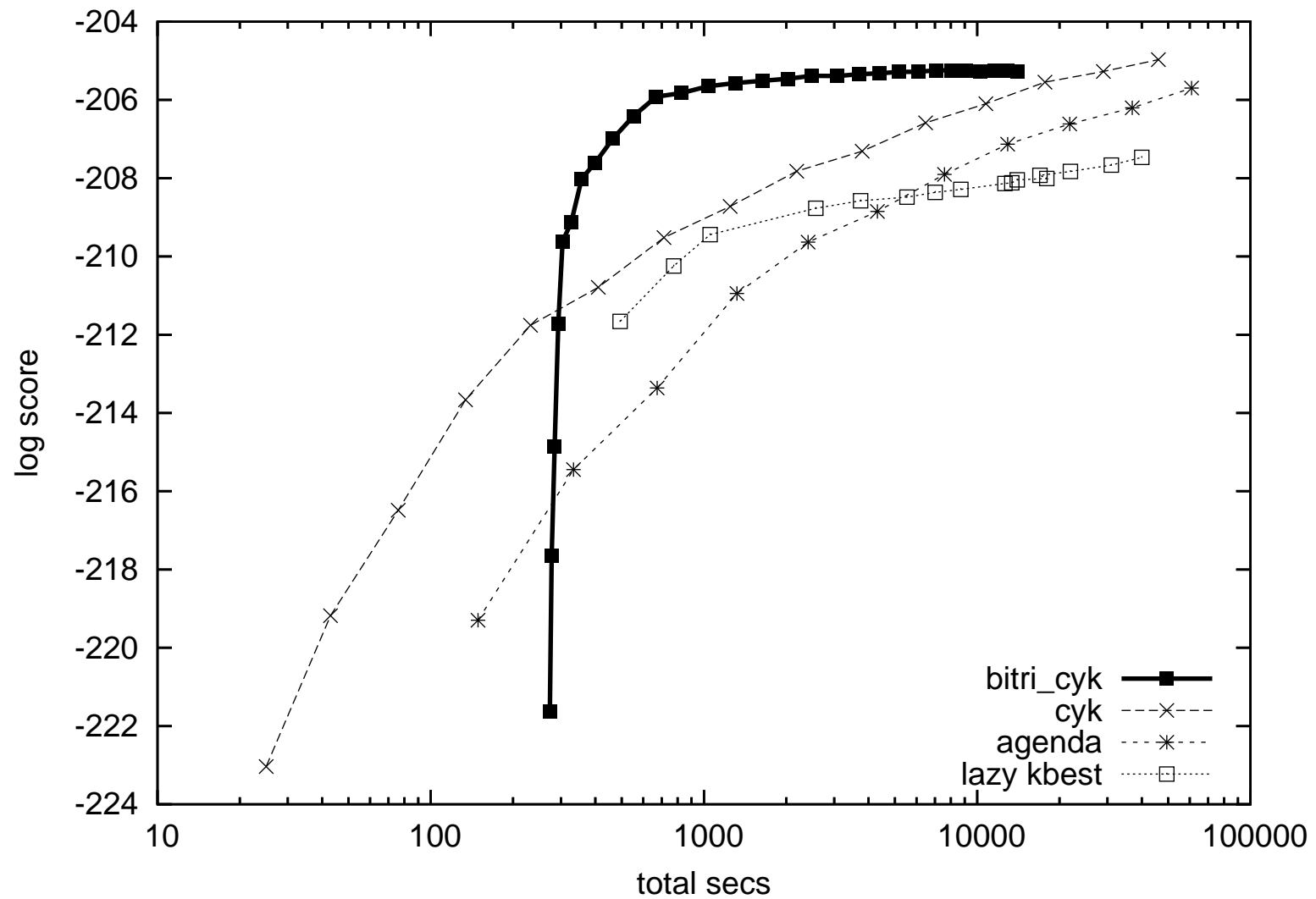
---

## Bigram-pass Outside Cost as Trigram-pass Outside Estimate

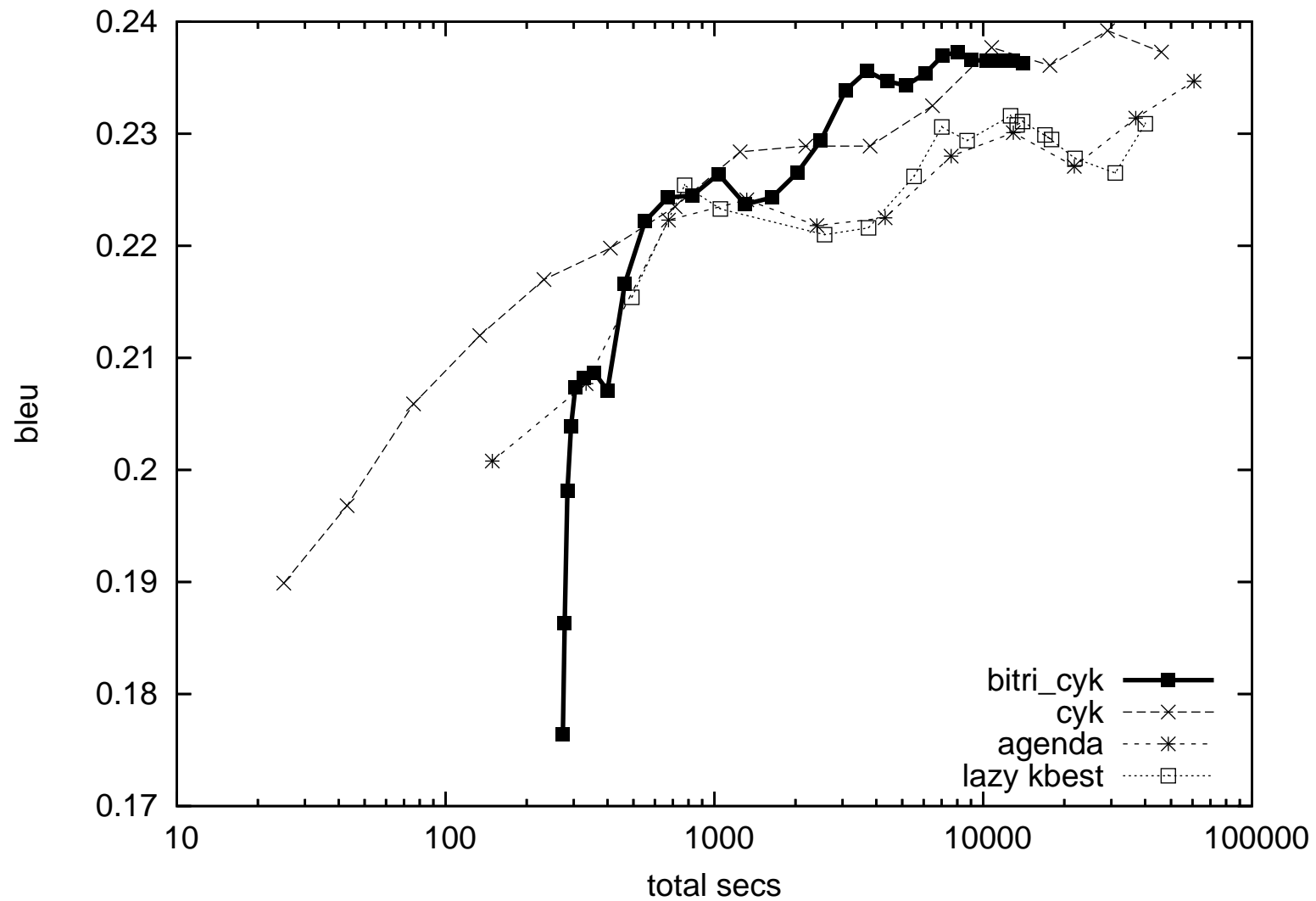
<i>Decoding Method</i>	Avg. Hyper-edges	BLEU
Bigram Pass	167K	21.77
Trigram Pass		
Uniform Outside Cost	–	–
Bigram Outside Cost(BO)	+ 629.7K=796.7K	23.56
<i>BO+Best Border Heuristic</i>	<i>+2.7K =169.7K</i>	<i>23.46</i>
Trigram One-pass, with Beam	6401K	23.47

(Zhang and Gildea, 2008)

## Two-pass decoding versus One-pass decoding



## Two-pass decoding versus One-pass decoding



---

## Decoding to Maximize BLEU

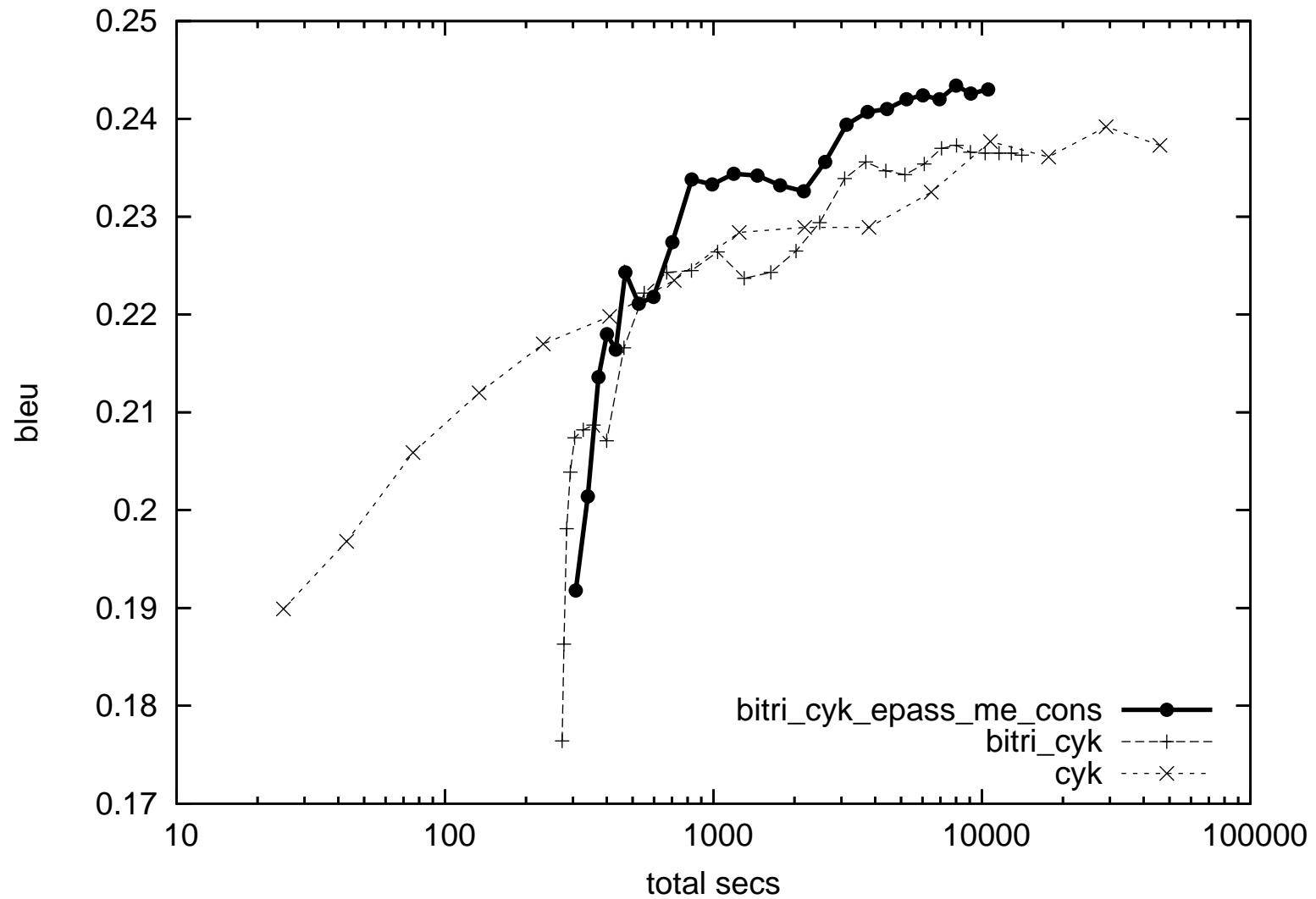
- Model scores and BLEU scores do not linearly correlate.
- Solution: another decoding pass that maximizes the expected count of translation hypotheses.
- This approximately maximizes the number of matching n-grams.

$$\begin{aligned} EC([X, i, j, u, u', v', v]) \\ = \alpha([X, i, j, u, u', v', v]) \cdot \beta([X, i, j, u, u', v', v]) \end{aligned}$$

- Similar to Goodman (1996)'s parsing algorithm for maximizing the expected labeled recall.

$$- \max_T \sum_{[X, i, j, u, u', v', v]} EC([X, i, j, u, u', v', v])$$

# Maximizing BLEU



---

## Summarization of Decoding Results

<i>Decoder</i>	Time	BLEU	Model Score
One-pass agenda	4317s	22.25	-208.849
One-pass CYK	3793s	22.89	-207.309
<hr/>			
<i>Multi-pass, CYK first</i>			
<i>agenda second pass</i>	3689s	23.56	-205.344
<hr/>			
<b>MEC third pass</b>	<b>3749s</b>	<b>24.07</b>	<b>-203.878</b>
<hr/>			
Lazy-cube-pruning	3746s	22.16	-208.575

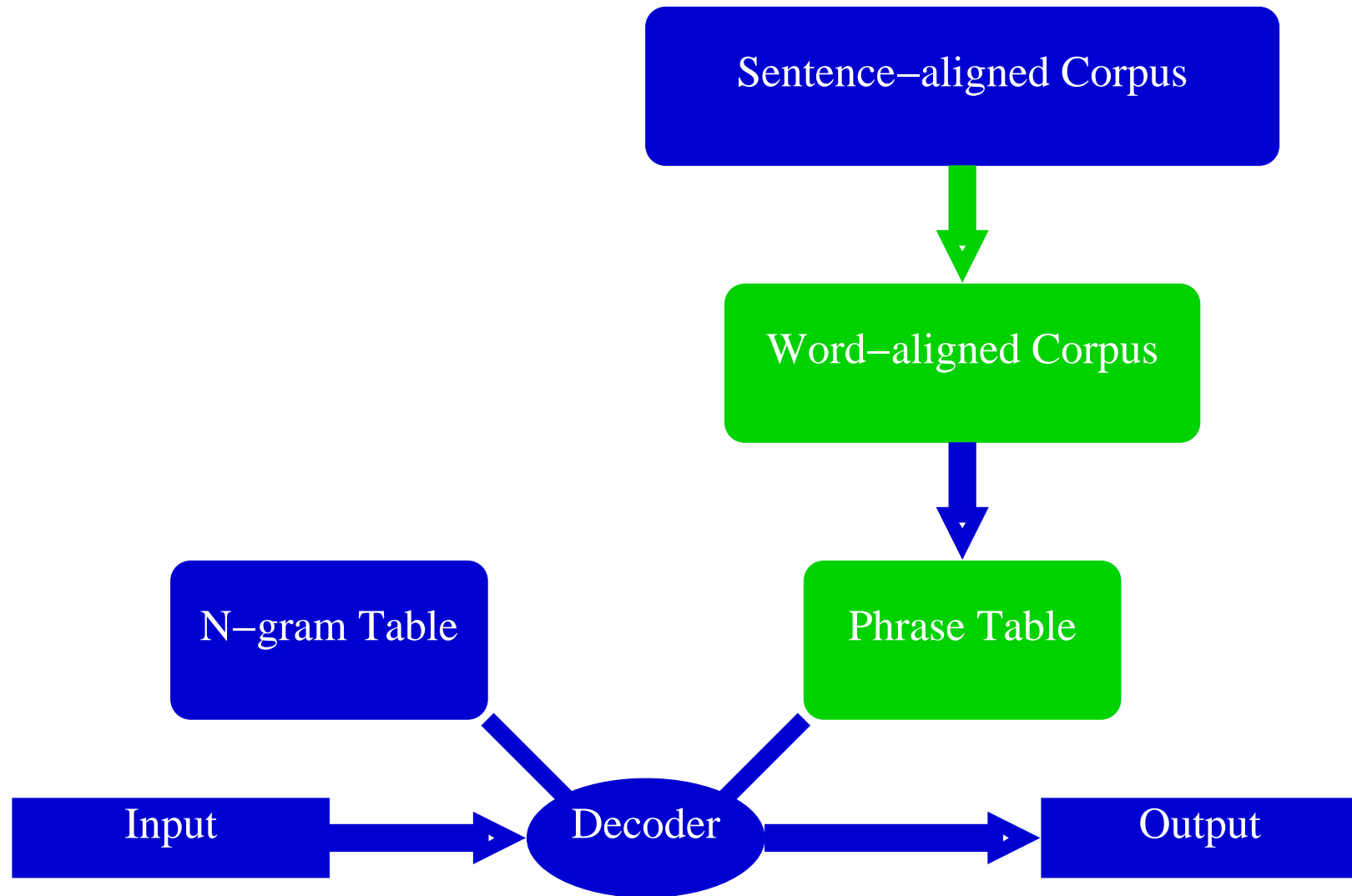
Summarization of different trigram decoding strategies, using about the same time (10 seconds per sentence).

---

## **Part III: Learning Phrases**

---

## Pipeline of Phrase-based Systems



---

## Difficulty of Learning Phrases

- Phrase pairs are substring pairs appearing in sentence pairs.
- The space of all phrase pairs is prohibitively large.
- Traditional EM faces overfitting: data memorization.

---

## Dirichlet Prior for Phrasal Inversion Transduction Grammar

$$\theta_x \mid \alpha_x \sim \text{Dir}(\alpha_x),$$

$$\theta_c \mid \alpha_c \sim \text{Dir}(\alpha_c),$$

$$\begin{array}{l|l} [X \ X] & \\ \langle X \ X \rangle & X \sim \text{Multi}(\theta_x). \\ C & \\ \mathbf{e/f} & C \sim \text{Multi}(\theta_c). \end{array}$$

---

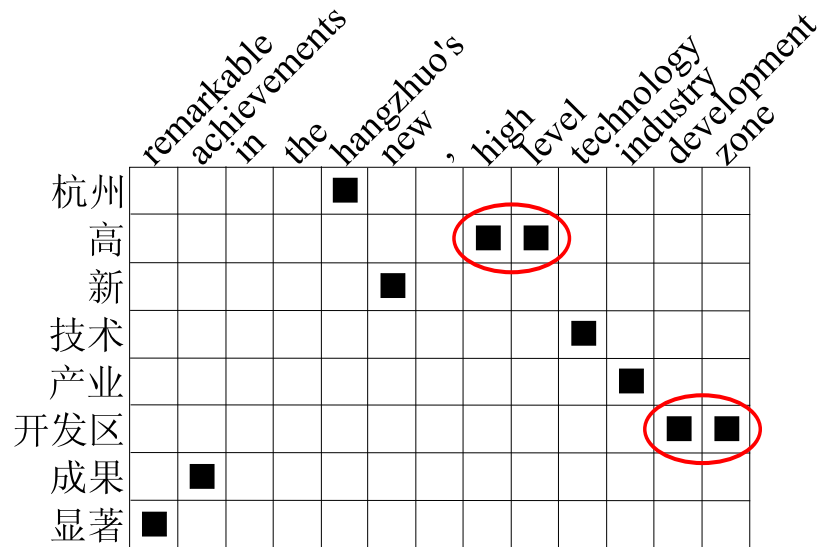
## Variational Bayesian Inference for ITG

We iteratively update the variational parameters to indirectly maximize the posterior probability given the data. The update rule for the phrase pairs is:

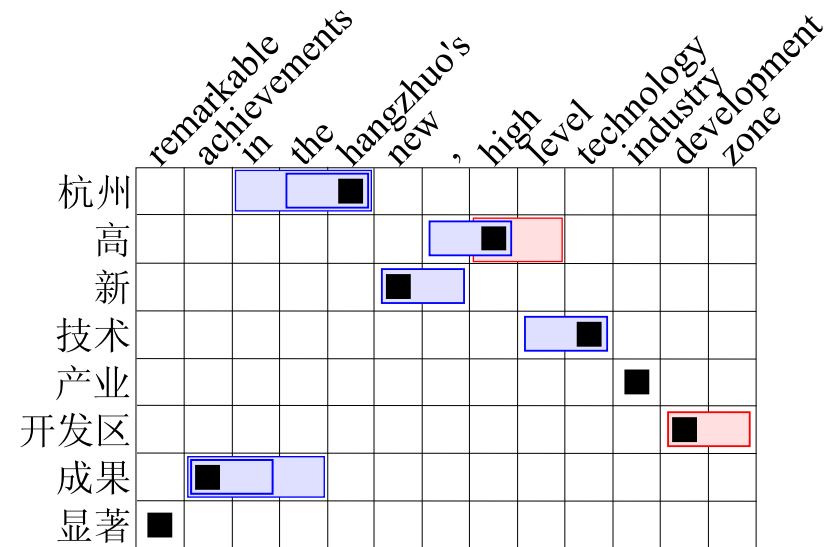
$$\tilde{P}^{(l+1)}(\mathbf{e}/\mathbf{f}) = \frac{\exp(\psi(E(\mathbf{e}/\mathbf{f}) + \alpha_C))}{\exp(\psi(E(C) + m\alpha_C))},$$

- Raw fractional counts  $c$  are replaced by  $\exp(\psi(c + \alpha))$ .
- The effect is to penalize infrequent pairs more.

## Hard Constraint: Non-compositional Phrases



(a)



(b)

(Cherry, 2007)

---

## End-to-end Evaluation

	<i>Development</i>	<i>Test</i>
GIZA++	37.46	28.24
ITG-word	35.47	26.55
ITG-mwm (VB)	39.21	<b>29.02</b>
ITG-mwm (EM)	39.15	28.47

Translation results on Chinese-English, using the subset of training data (141K sentence pairs) that have length limit 35 on both sides. (No length limit in translation. )

(Zhang, Quirk, Moore and Gildea, 2008)

---

## **Conclusions and Future Work**

---

## Conclusions

- Binarizable SCFGs are expressive and efficient.
- The marriage of coarse-to-fine and A\* decoding produces fast SCFG decoding.
- Dealing with metrics of MT evaluation in decoding achieves good translations.
- The combination of Bayesian learning and ITG constraint guides learning of phrases.

---

## Future Work

- Micro-scoping the space of Binarizable SCFGs to look for SCFGs that are optimal in trading off efficiency and expressiveness.
- Even faster decoding algorithms.
- Efficient unsupervised learning for SCFGs.
- More mathematically principled way of learning phrase pairs.
- An end-to-end SCFG system.

---

**Thank You**