Inducing Sentence Structure from Parallel Corpora for Reordering

John DeNero and Jakob Uszkoreit

Google research

Overview

Overview

- Translation is an end-task application for syntactic parsing

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation
  - Structural (hierarchical) reordering

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation
  - Structural (hierarchical) reordering

- Syntactic pre-ordering uses a supervised parser to predict structure

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation
  - Structural (hierarchical) reordering

- Syntactic pre-ordering uses a supervised parser to predict structure

- *This talk*: Unsupervised approach to predicting sentence structure

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation
  - Structural (hierarchical) reordering
- Syntactic pre-ordering uses a supervised parser to predict structure
- *This talk*: Unsupervised approach to predicting sentence structure
  - Pipeline of reordering-focused prediction problems

Overview

- Translation is an end-task application for syntactic parsing
  - Lexical disambiguation
  - Structural (hierarchical) reordering

- Syntactic pre-ordering uses a supervised parser to predict structure

- *This talk*: Unsupervised approach to predicting sentence structure
  - Pipeline of reordering-focused prediction problems
  - Learning signal comes from aligned parallel corpora

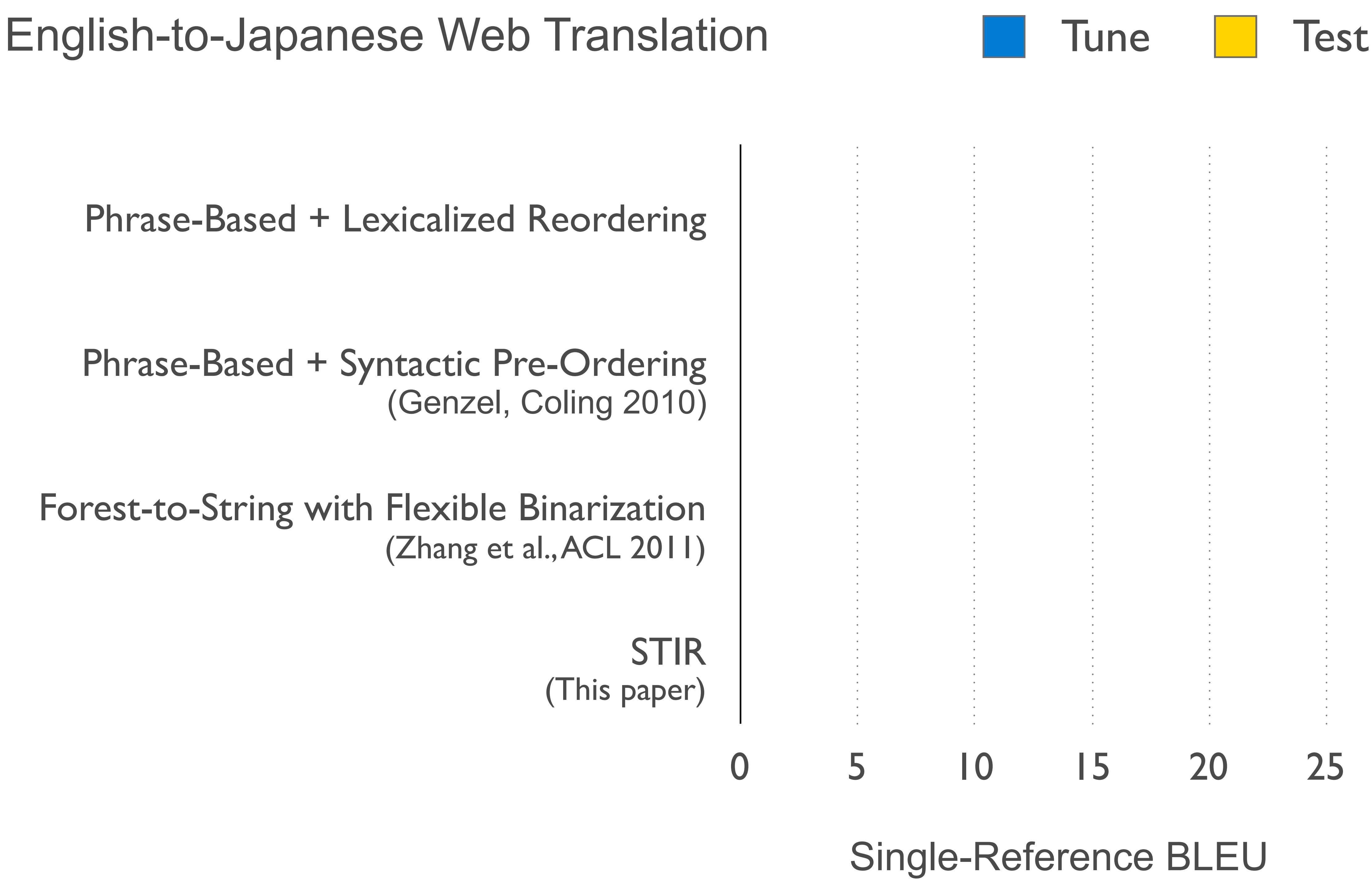
Syntax Matters for Hierarchical Reordering

English-to-Japanese Web Translation

Tune Test

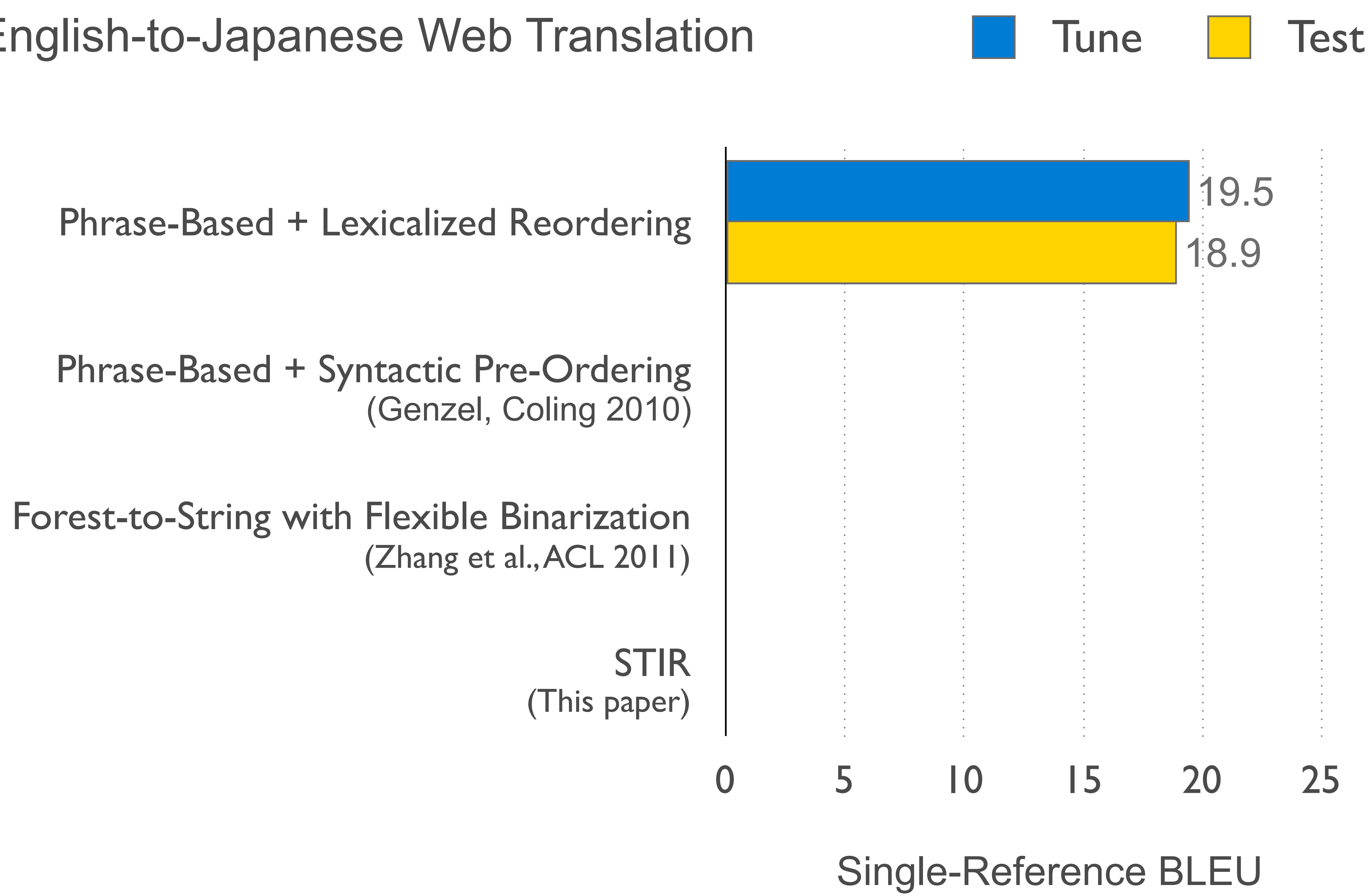
Single-Reference BLEU

Syntax Matters for Hierarchical Reordering

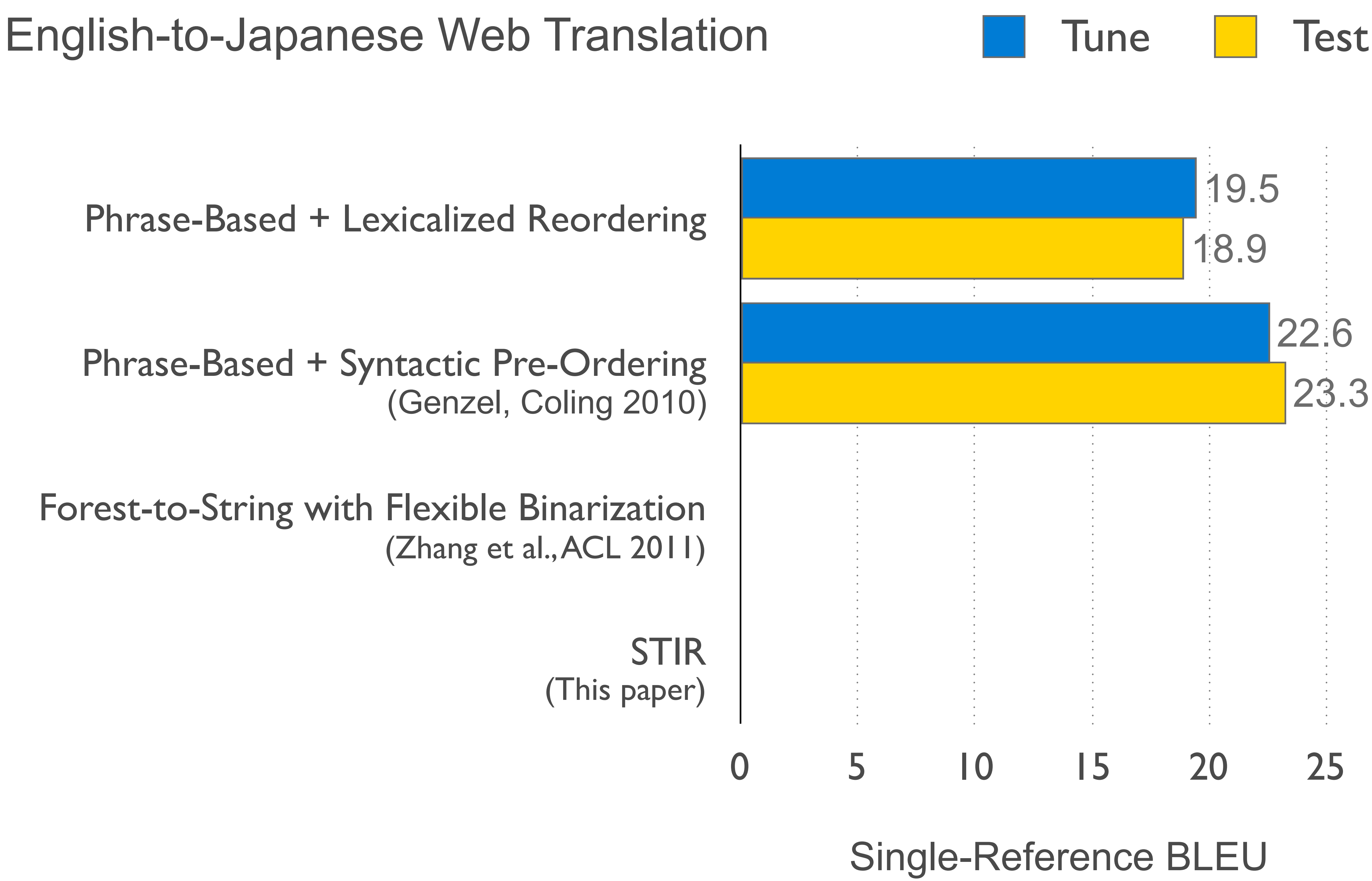


Syntax Matters for Hierarchical Reordering

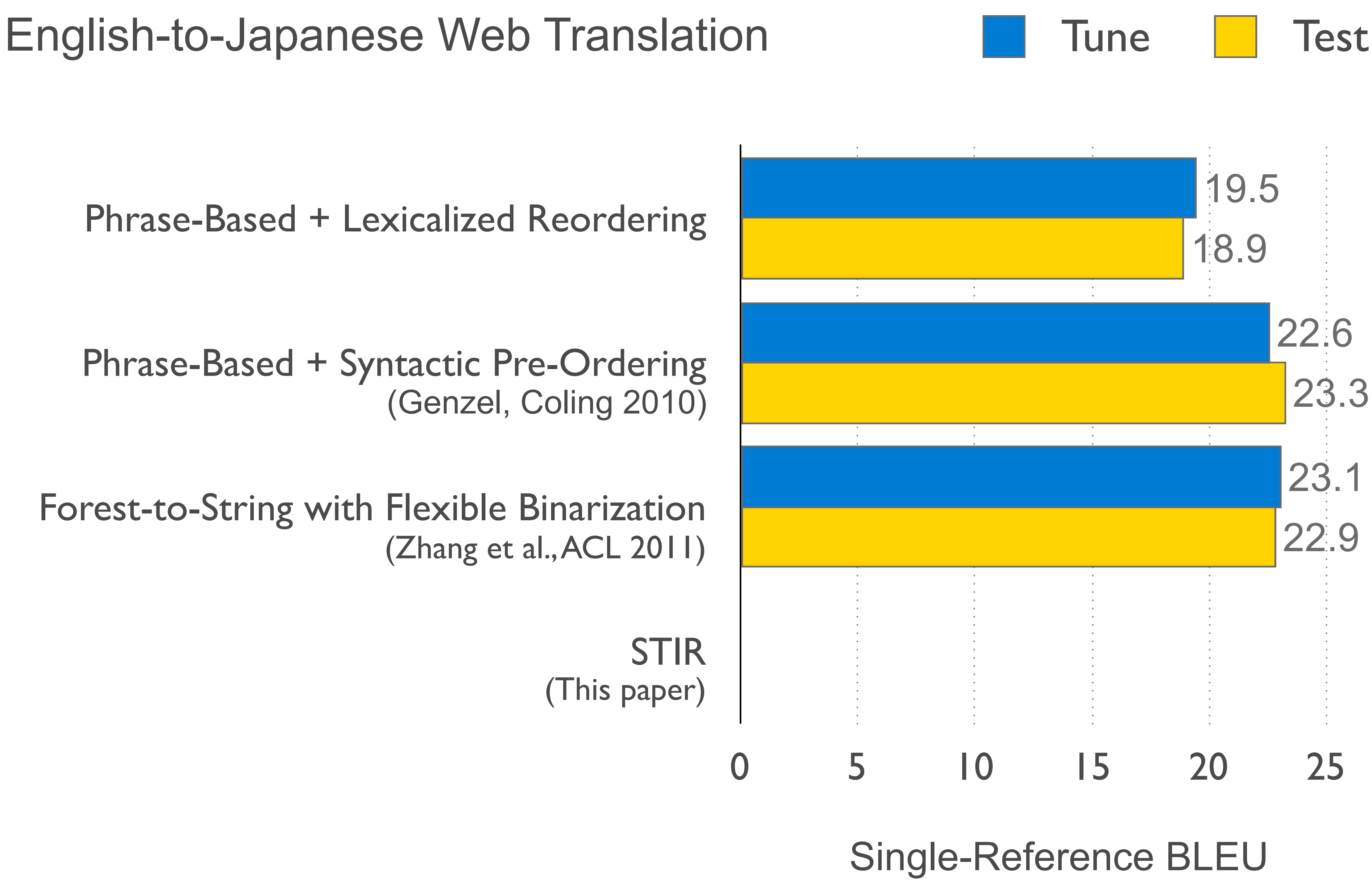
English-to-Japanese Web Translation



Syntax Matters for Hierarchical Reordering

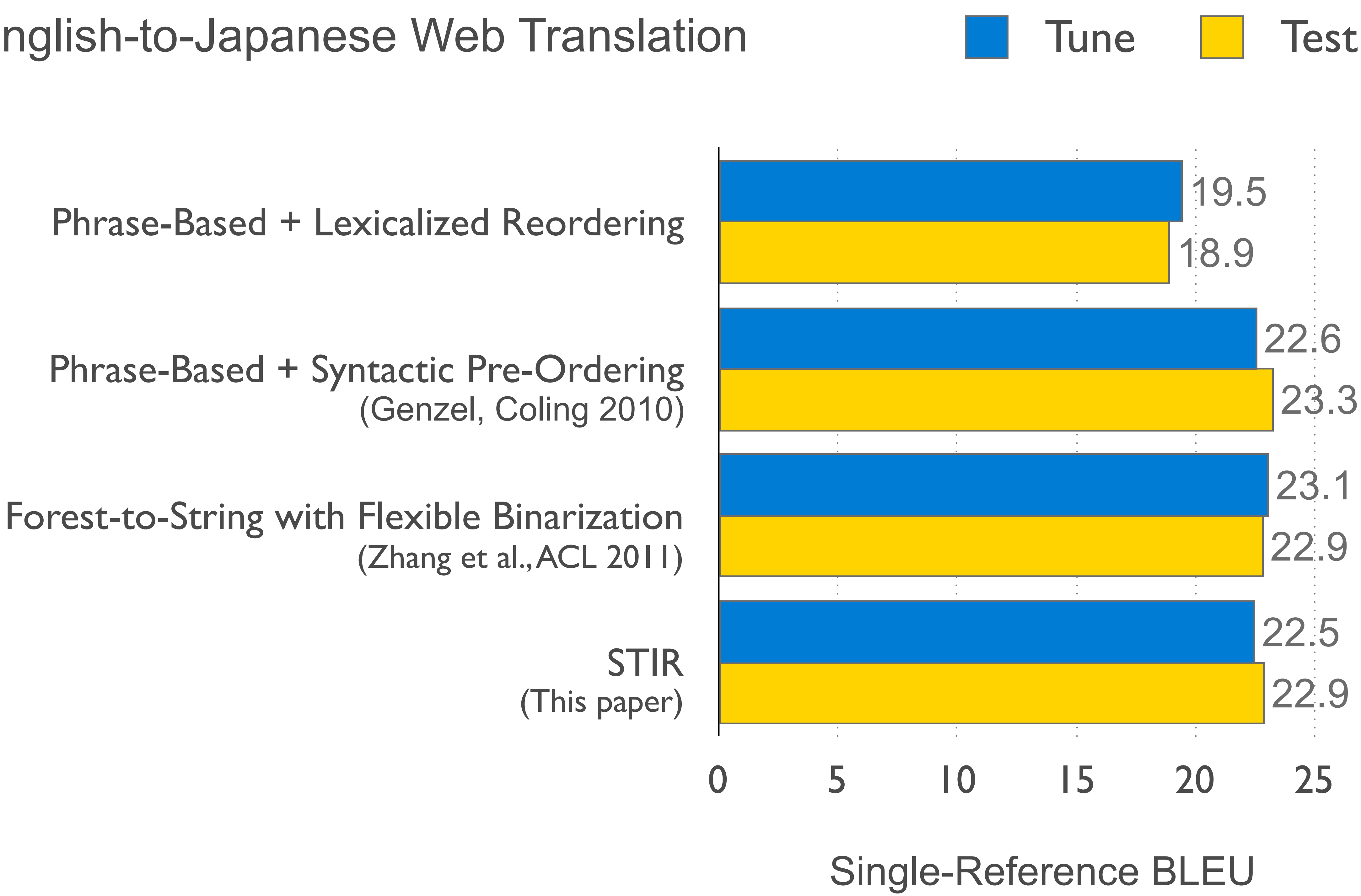


Syntax Matters for Hierarchical Reordering

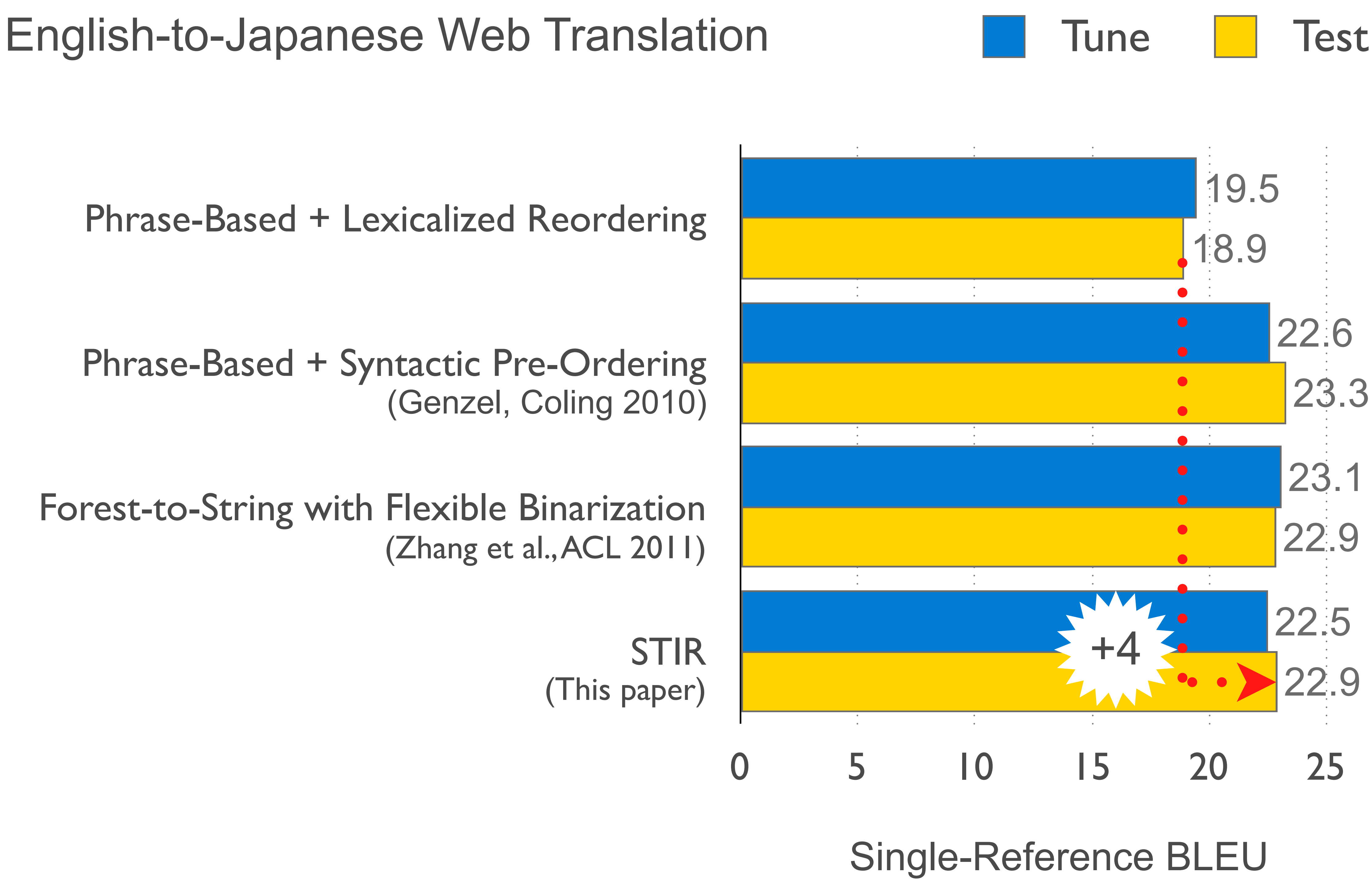


Syntax Matters for Hierarchical Reordering

English-to-Japanese Web Translation



Syntax Matters for Hierarchical Reordering



Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline

Translation Pipeline

Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline

Translation Pipeline

S V O

Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline

Translation Pipeline

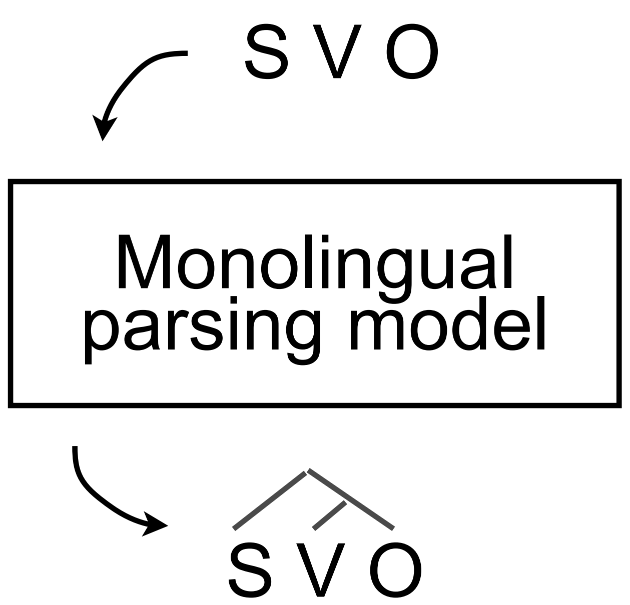
S V O

Monolingual parsing model

Syntactic Pre-Ordering in Phrase-Based Systems

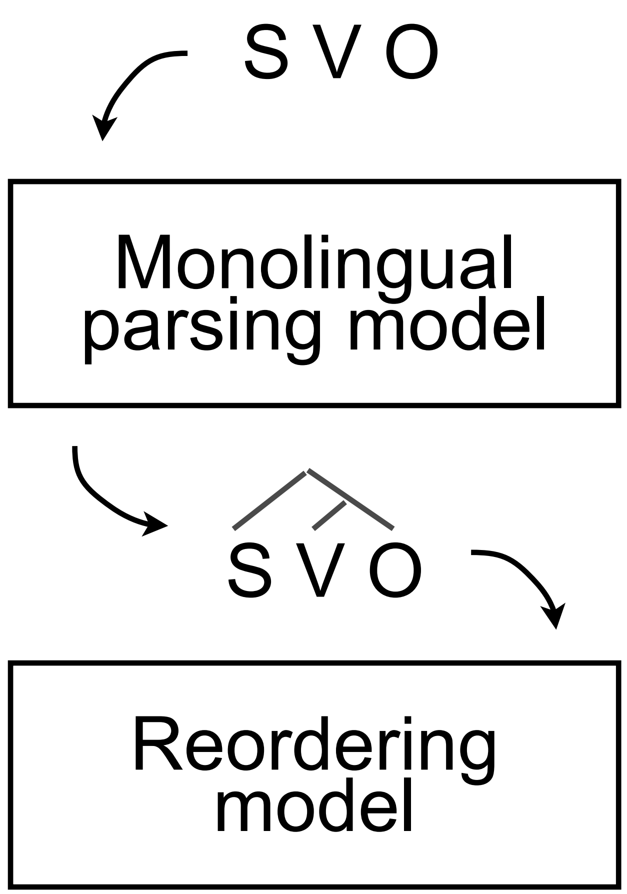
Pre-Ordering Pipeline

Translation Pipeline



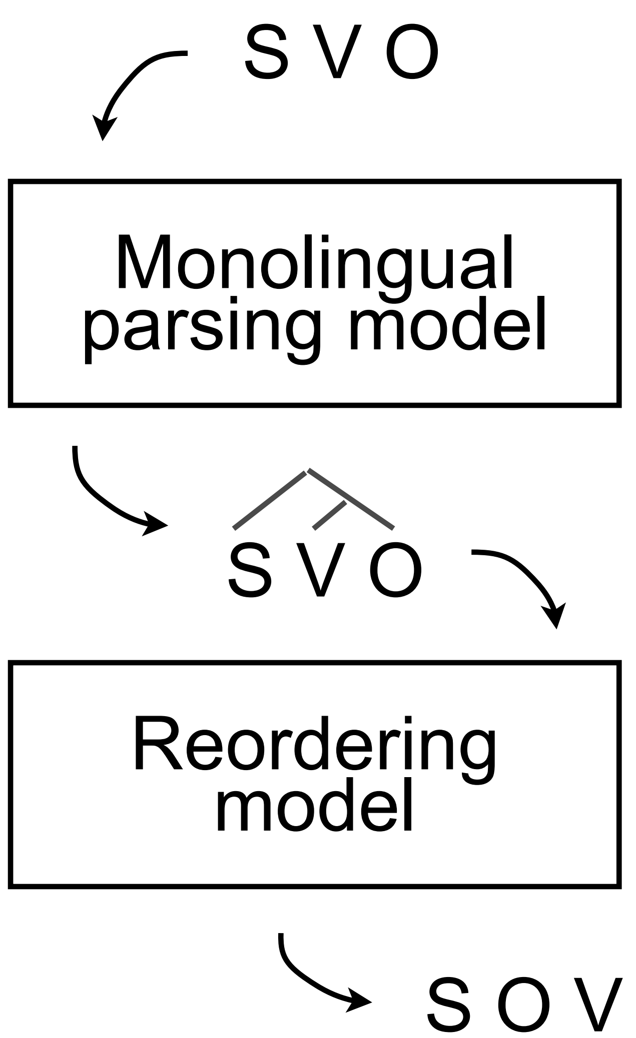
Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline | **Translation Pipeline**



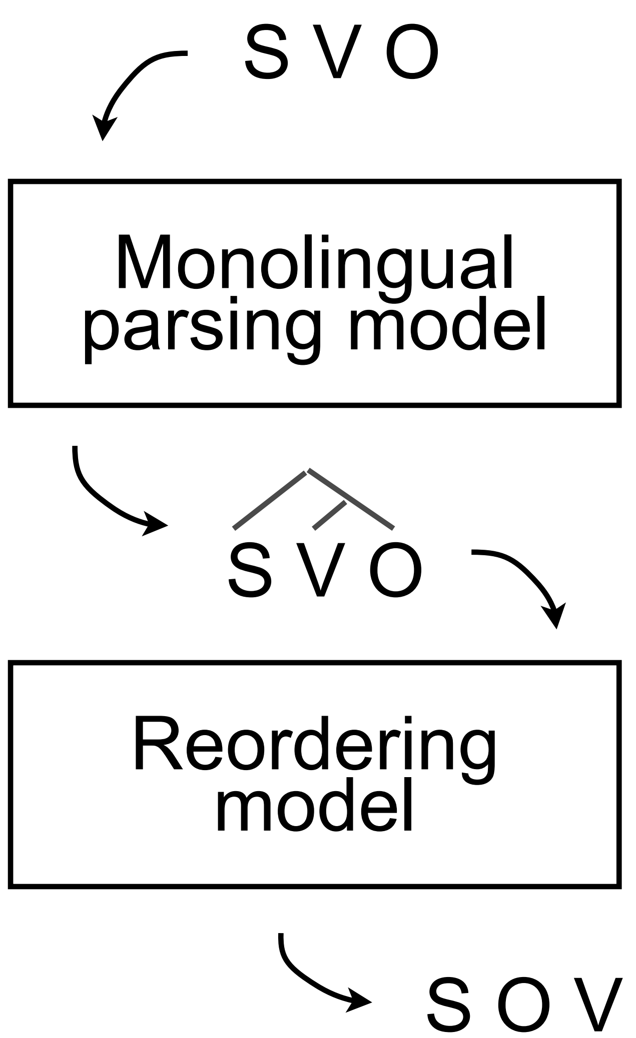
Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline **Translation Pipeline**



Syntactic Pre-Ordering in Phrase-Based Systems

Pre-Ordering Pipeline



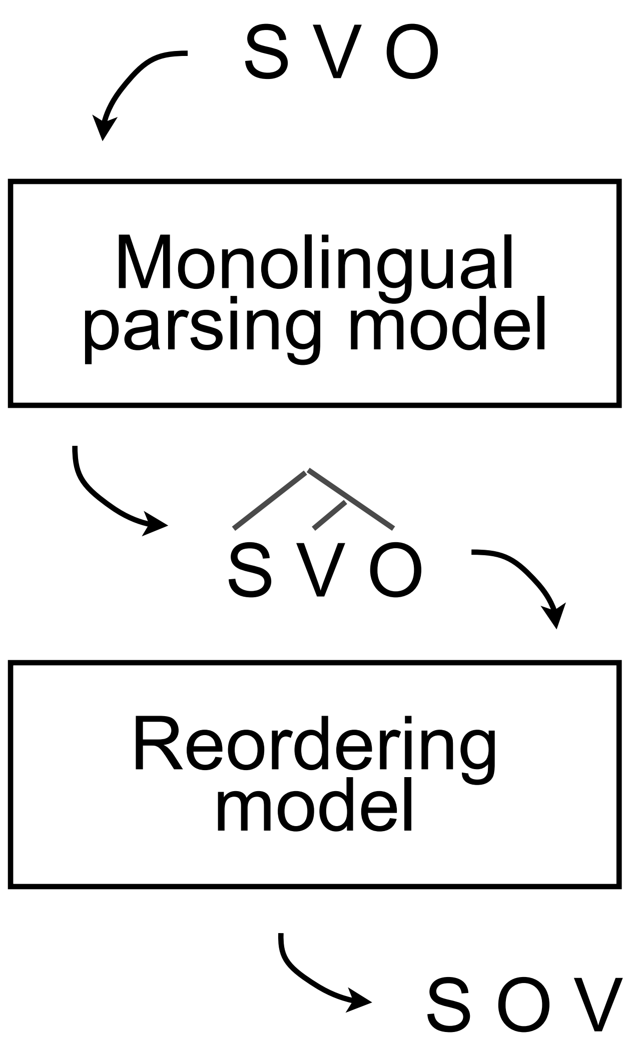
Translation Pipeline

Source training

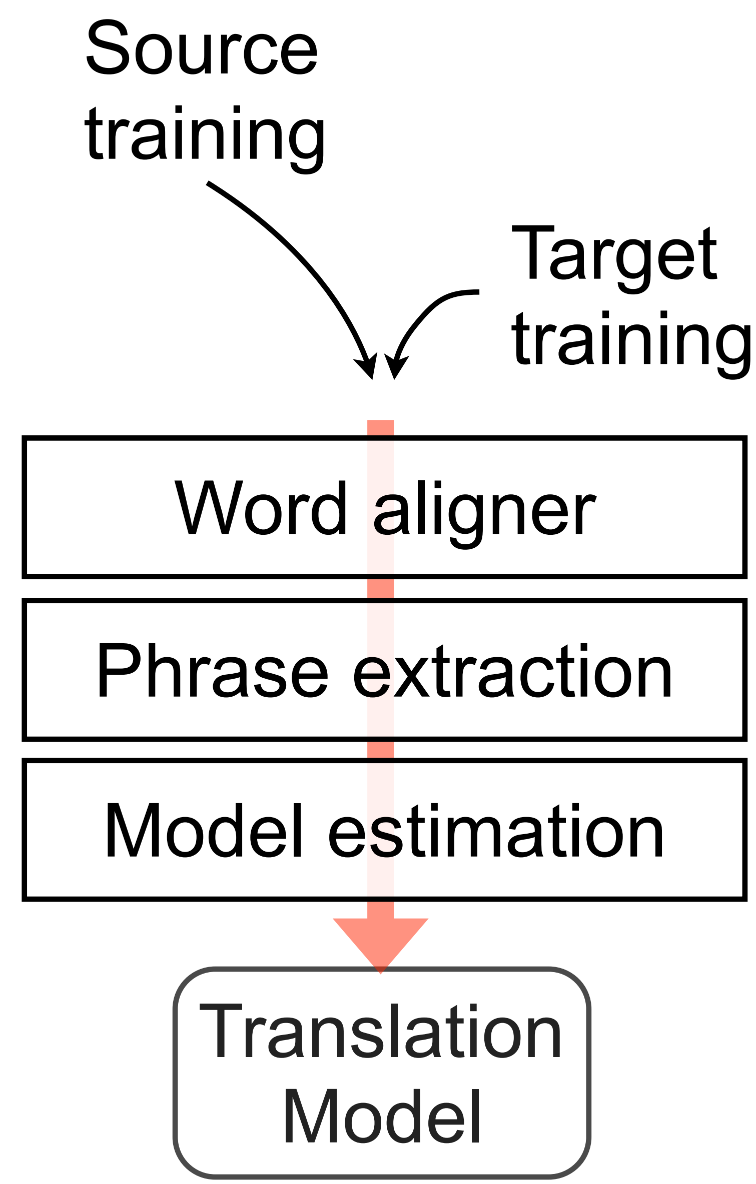
Target training

Syntactic Pre-Ordering in Phrase-Based Systems

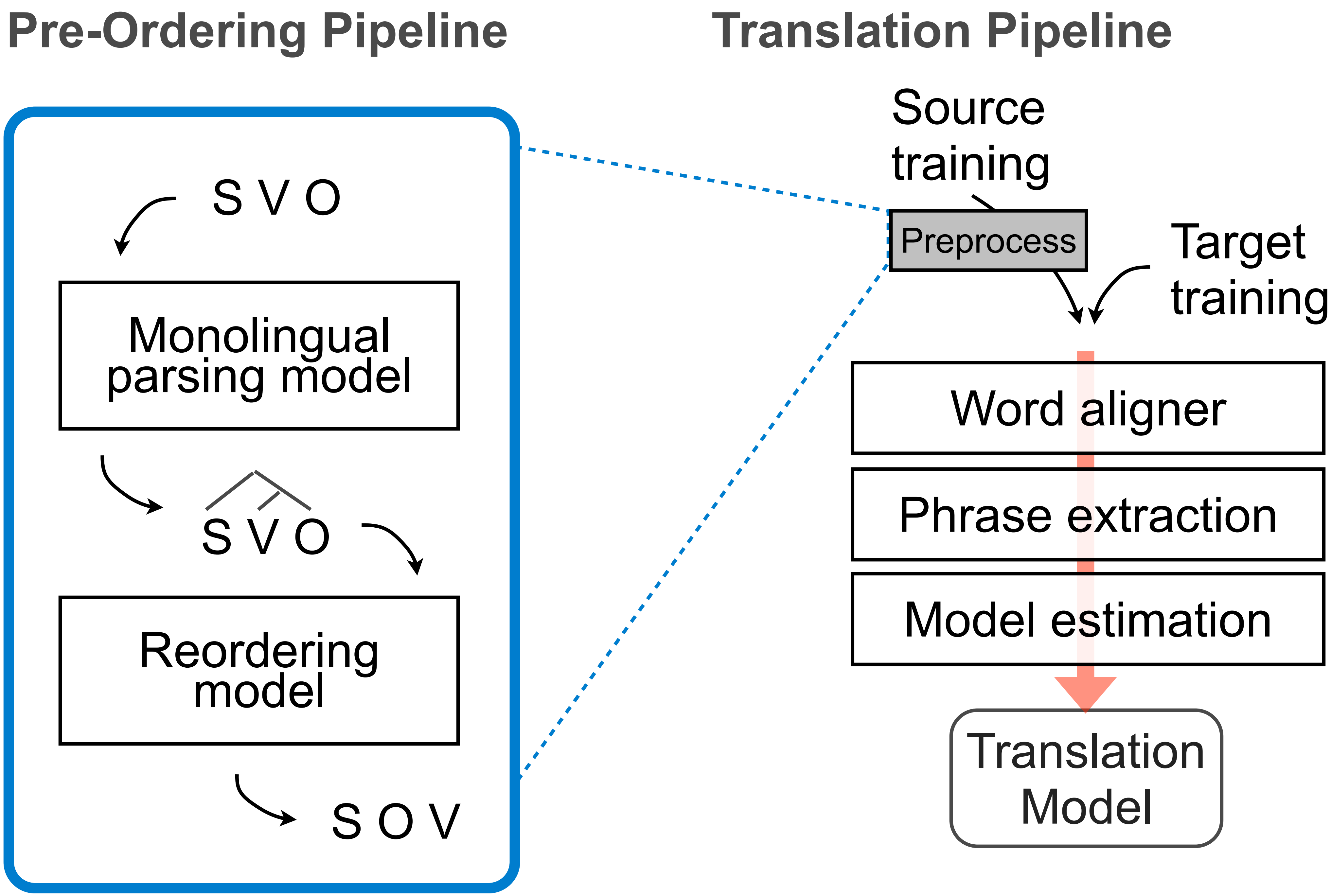
Pre-Ordering Pipeline



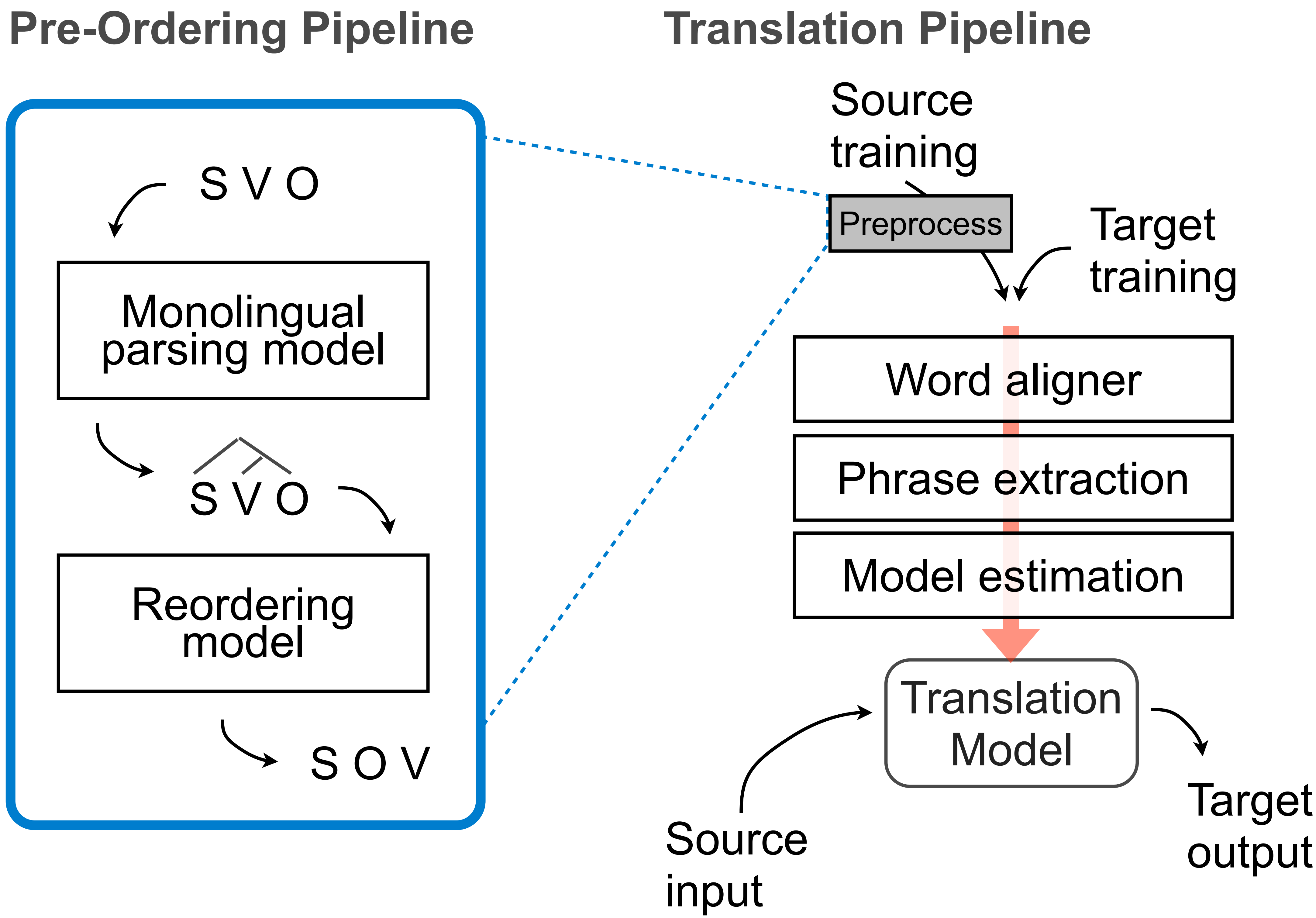
Translation Pipeline



Syntactic Pre-Ordering in Phrase-Based Systems

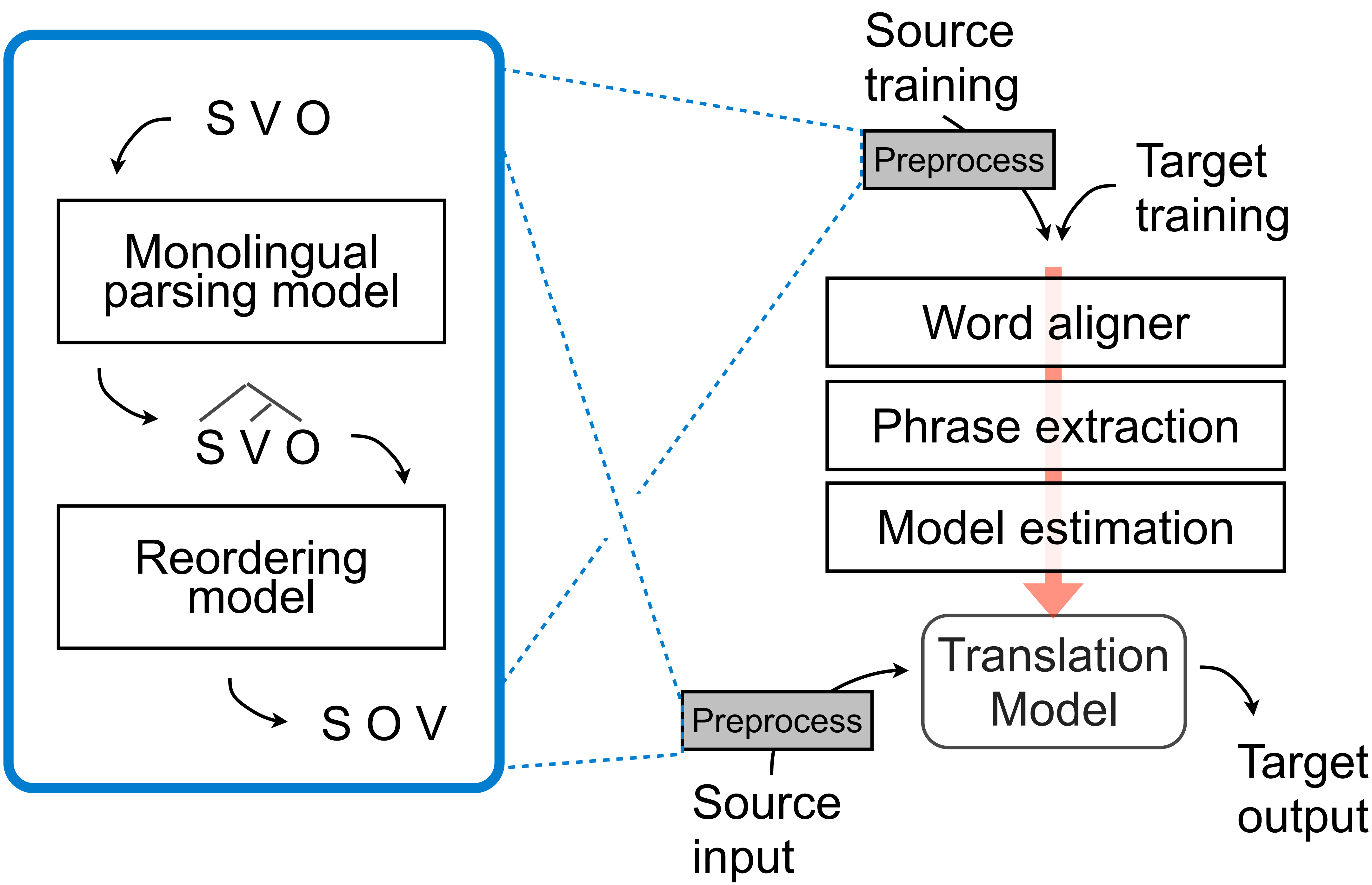


Syntactic Pre-Ordering in Phrase-Based Systems



Syntactic Pre-Ordering in Phrase-Based Systems

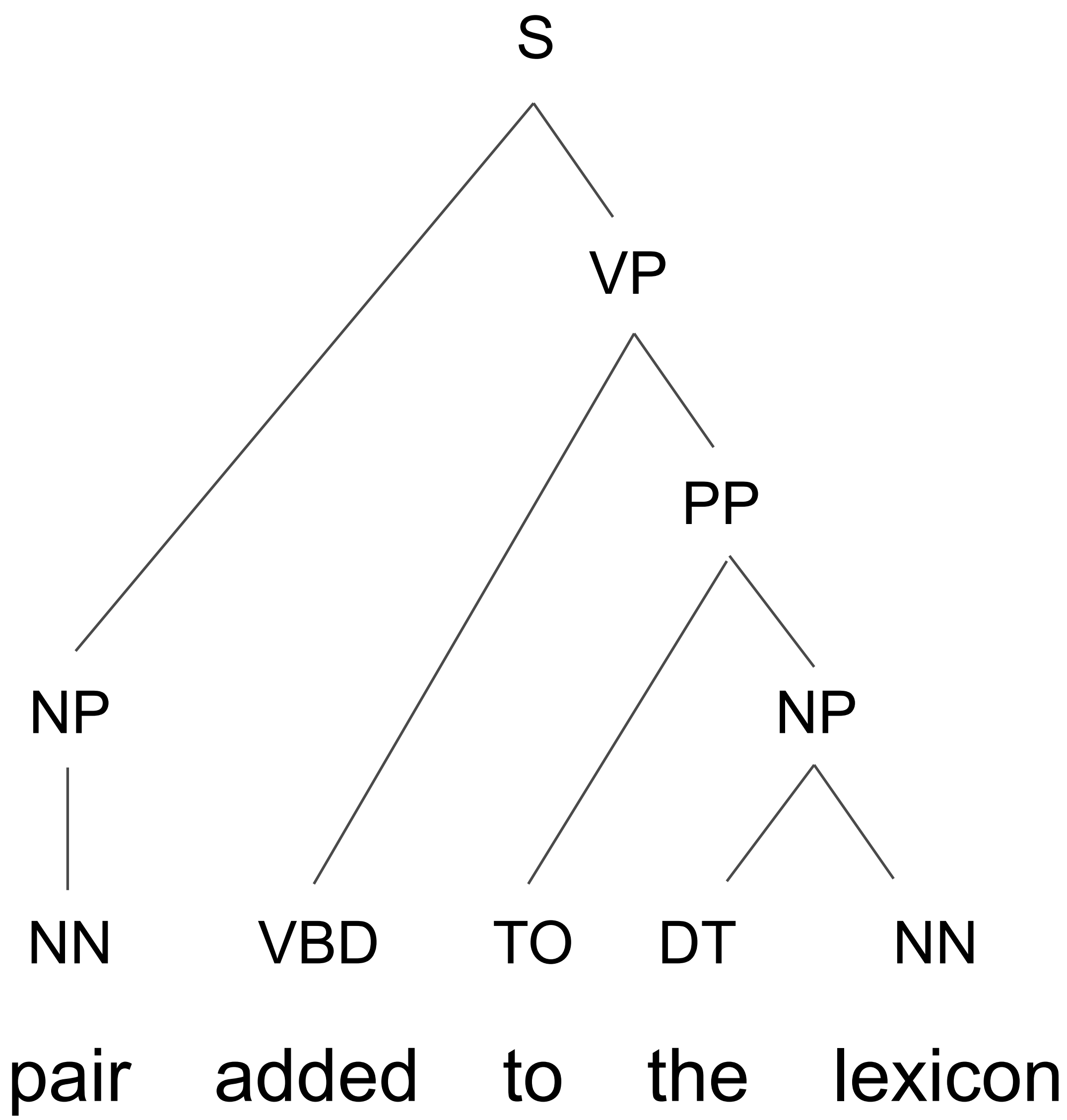
Pre-Ordering Pipeline **Translation Pipeline**



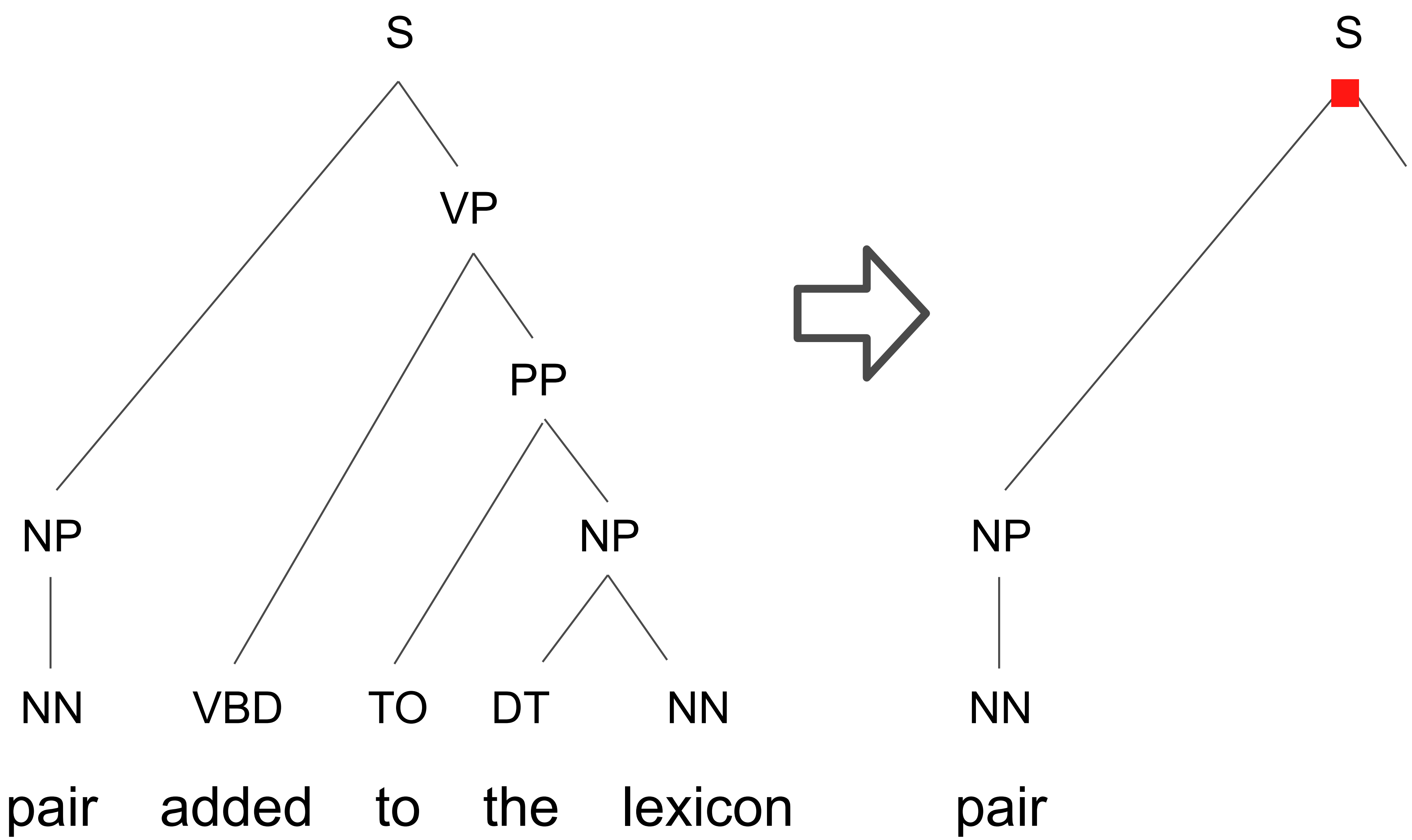
Syntactic Pre-Ordering Example

pair added to the lexicon

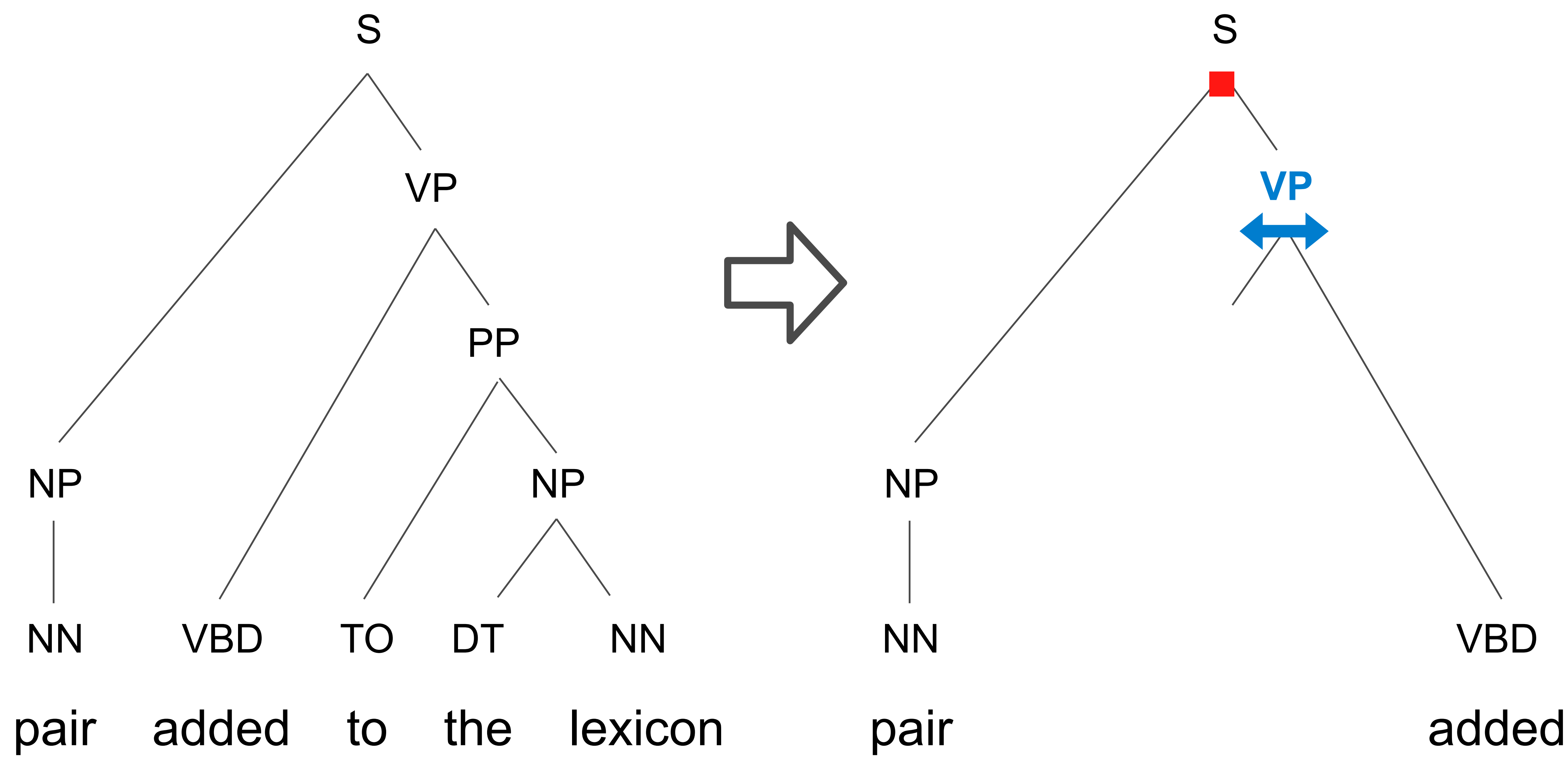
Syntactic Pre-Ordering Example



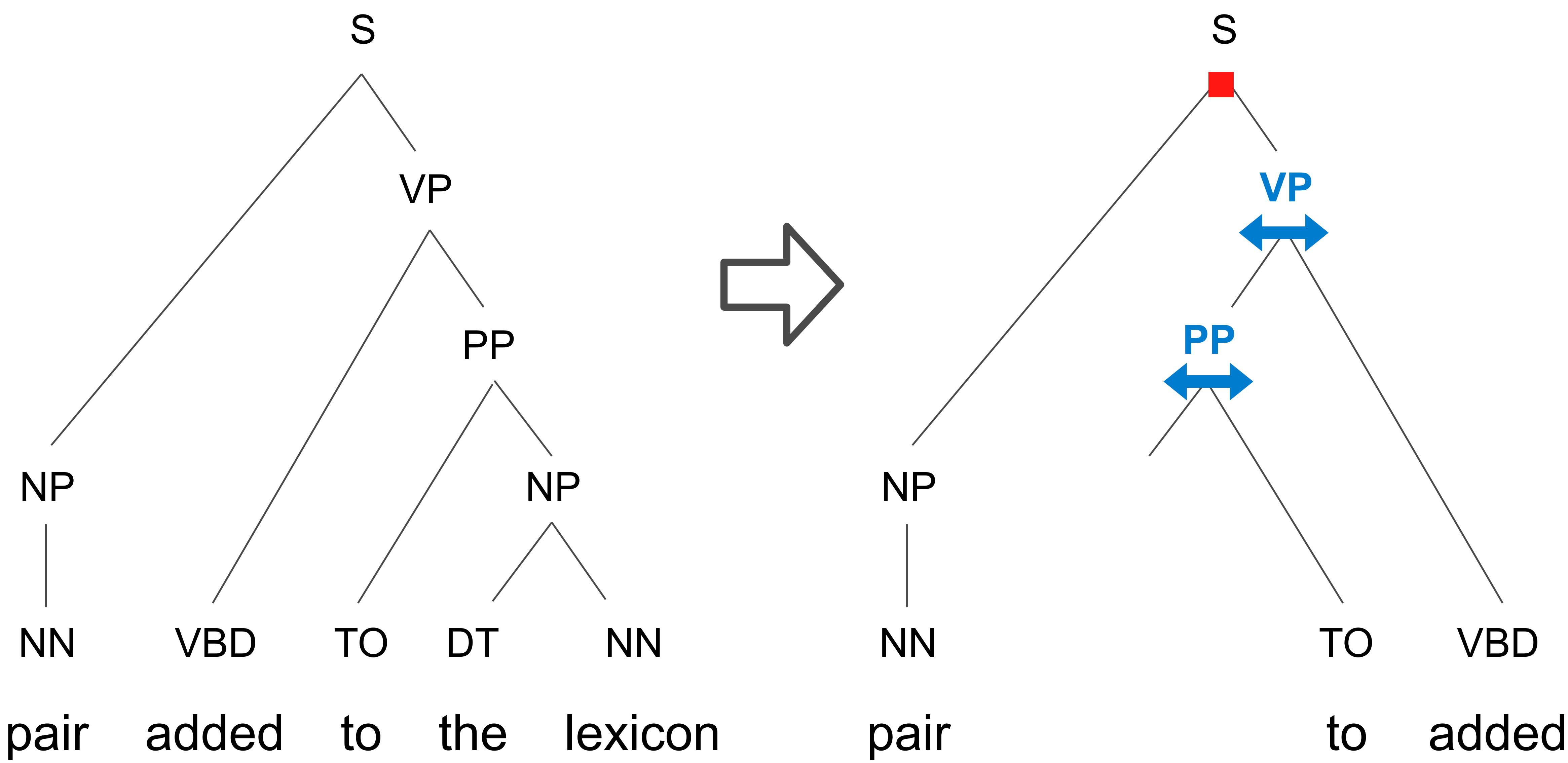
Syntactic Pre-Ordering Example



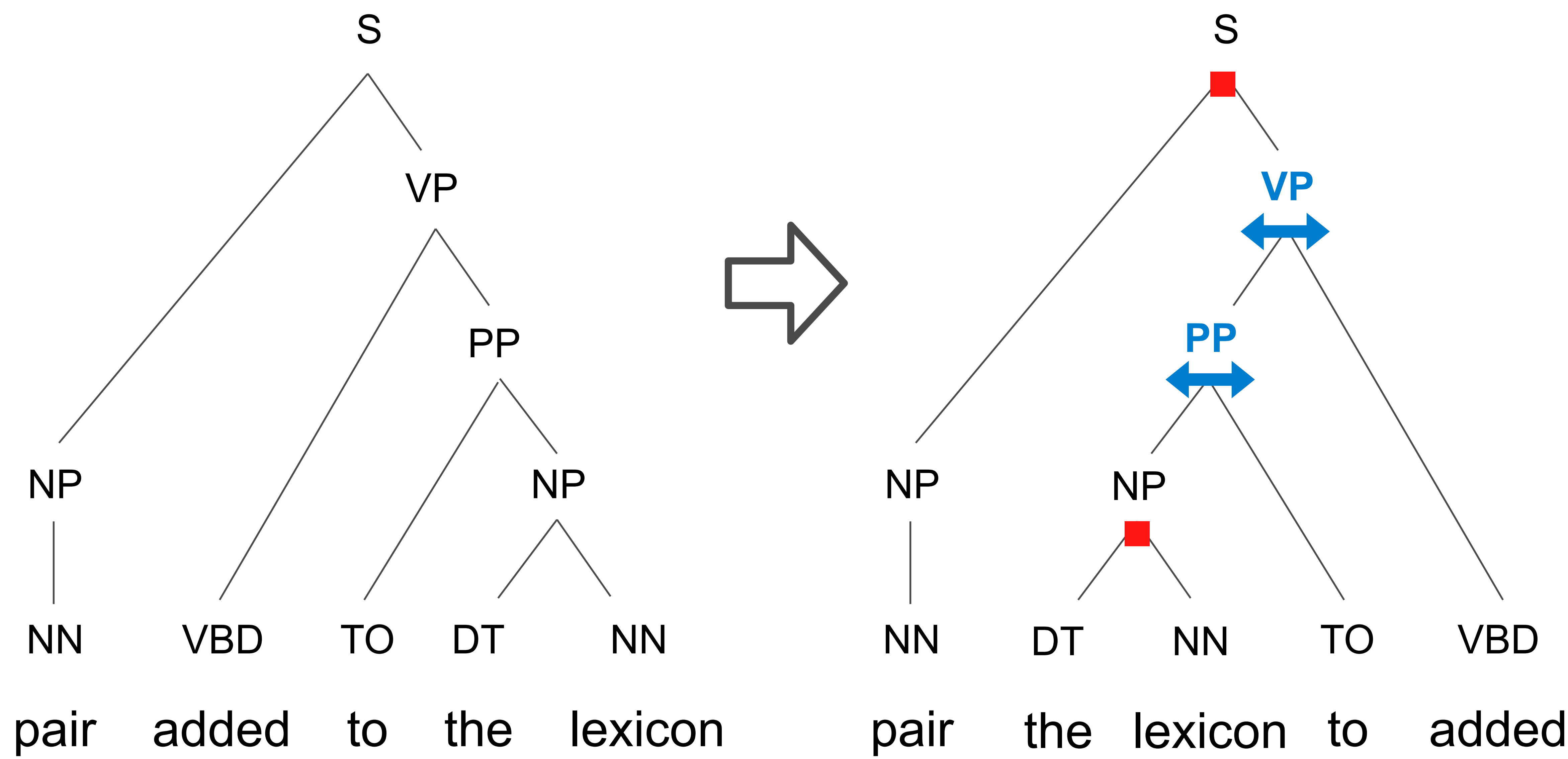
Syntactic Pre-Ordering Example



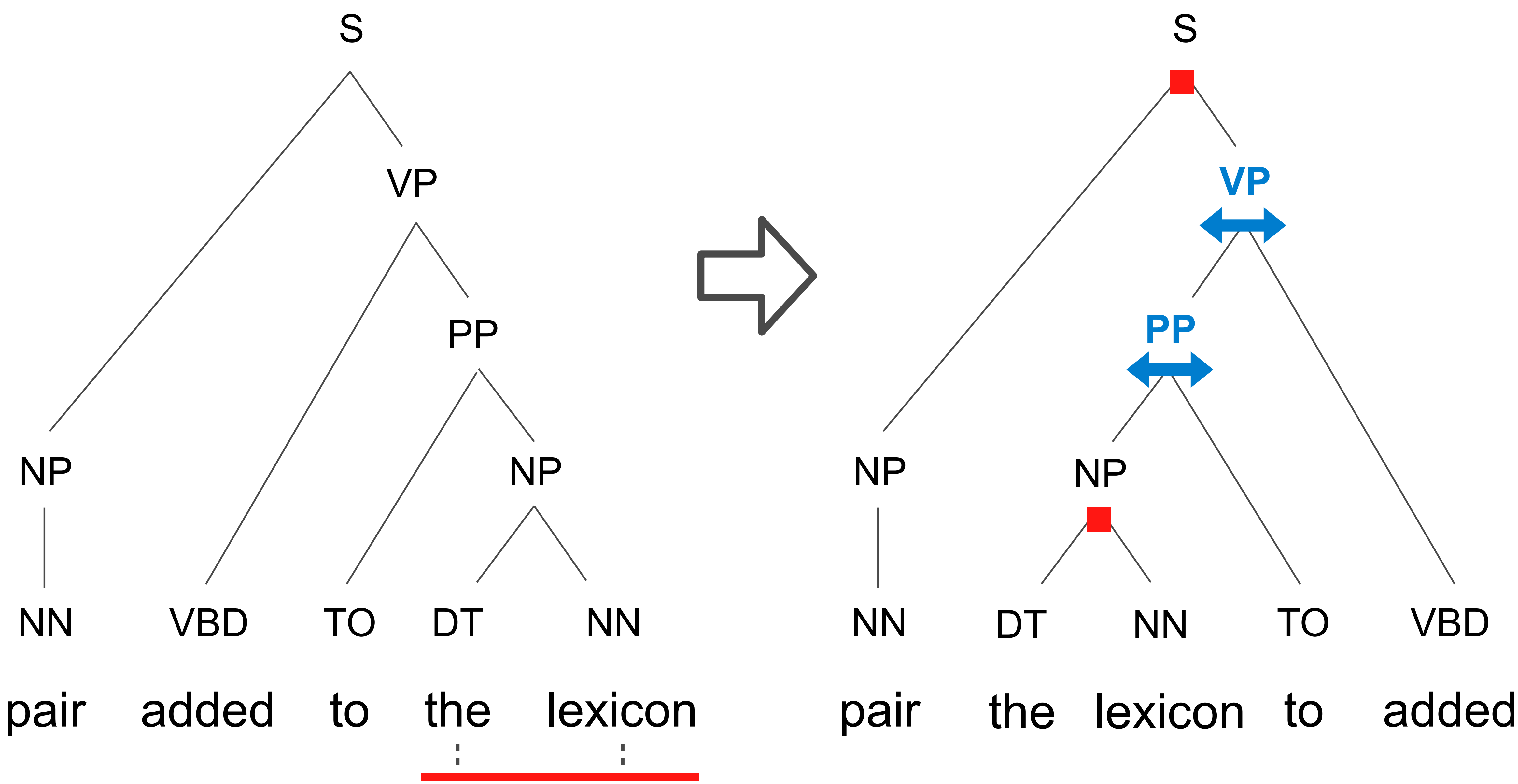
Syntactic Pre-Ordering Example



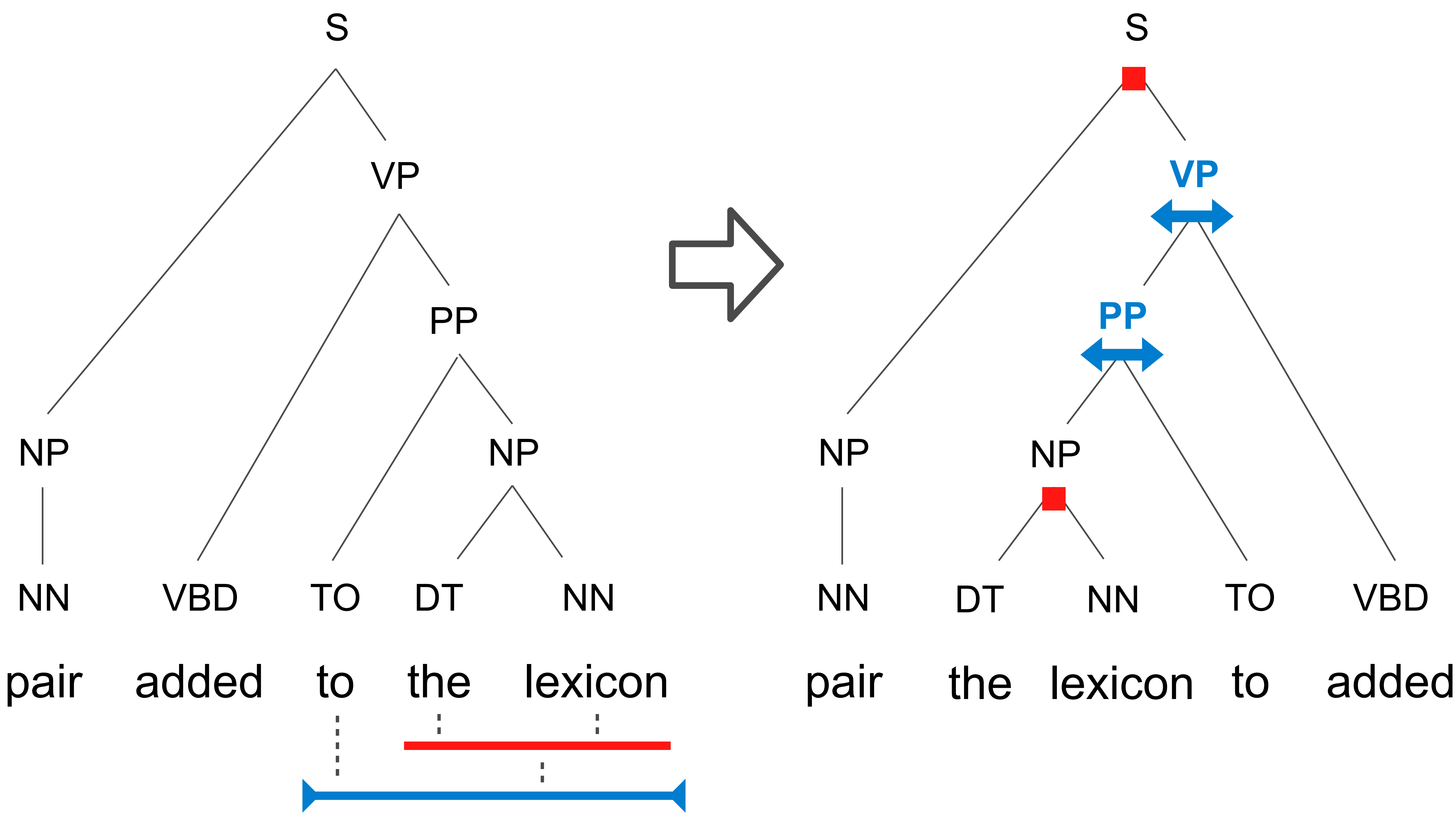
Syntactic Pre-Ordering Example



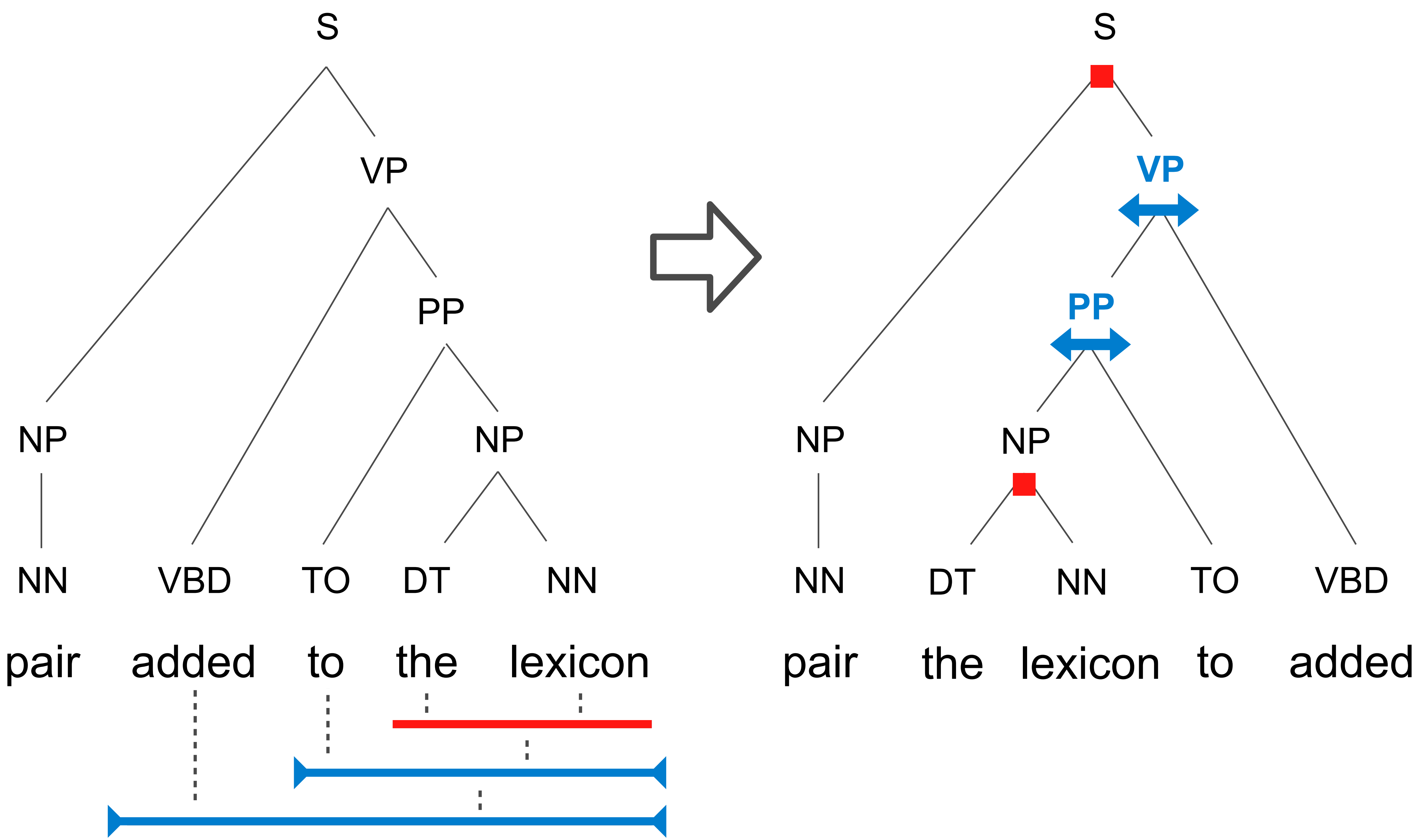
Syntactic Pre-Ordering Example



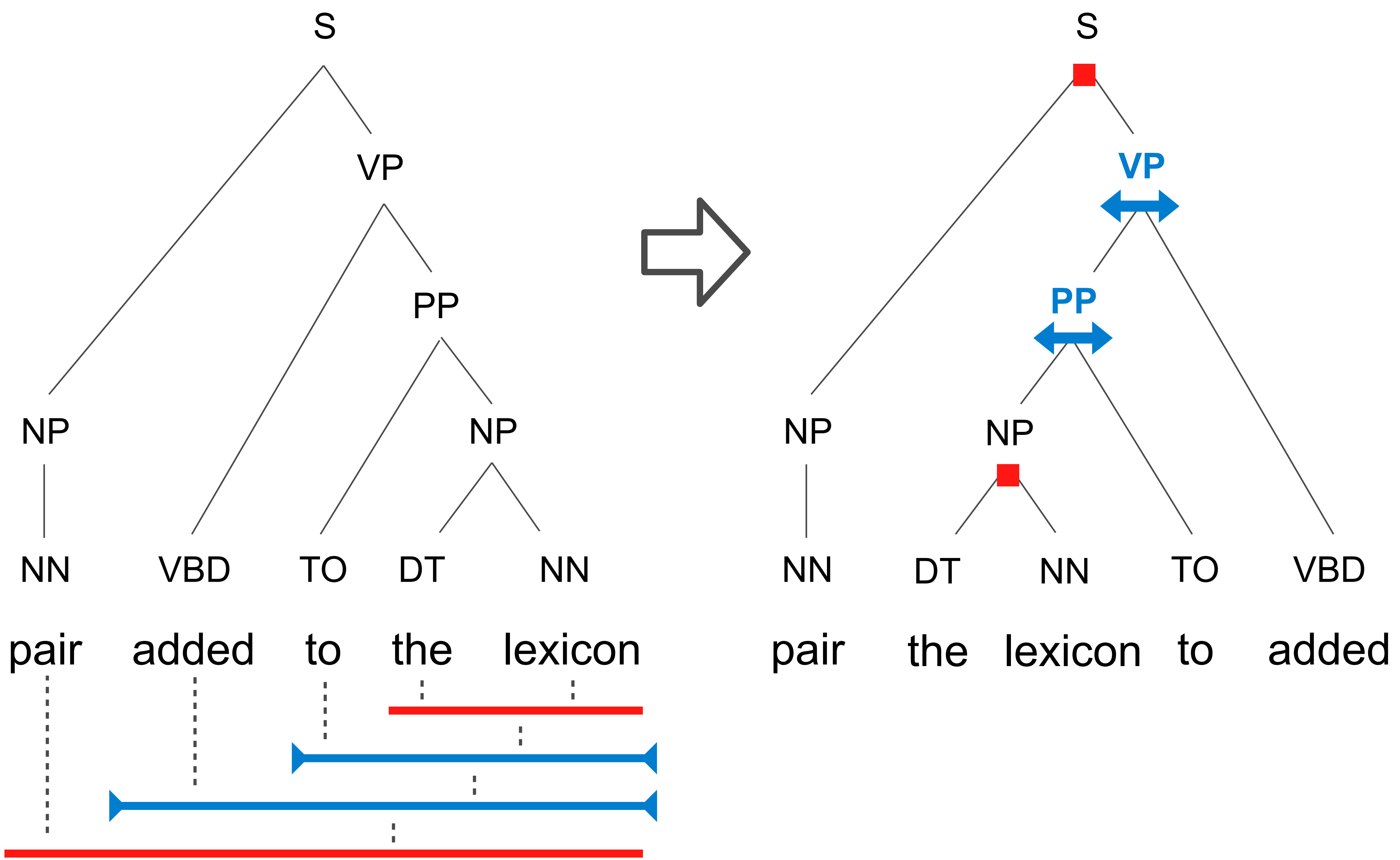
Syntactic Pre-Ordering Example



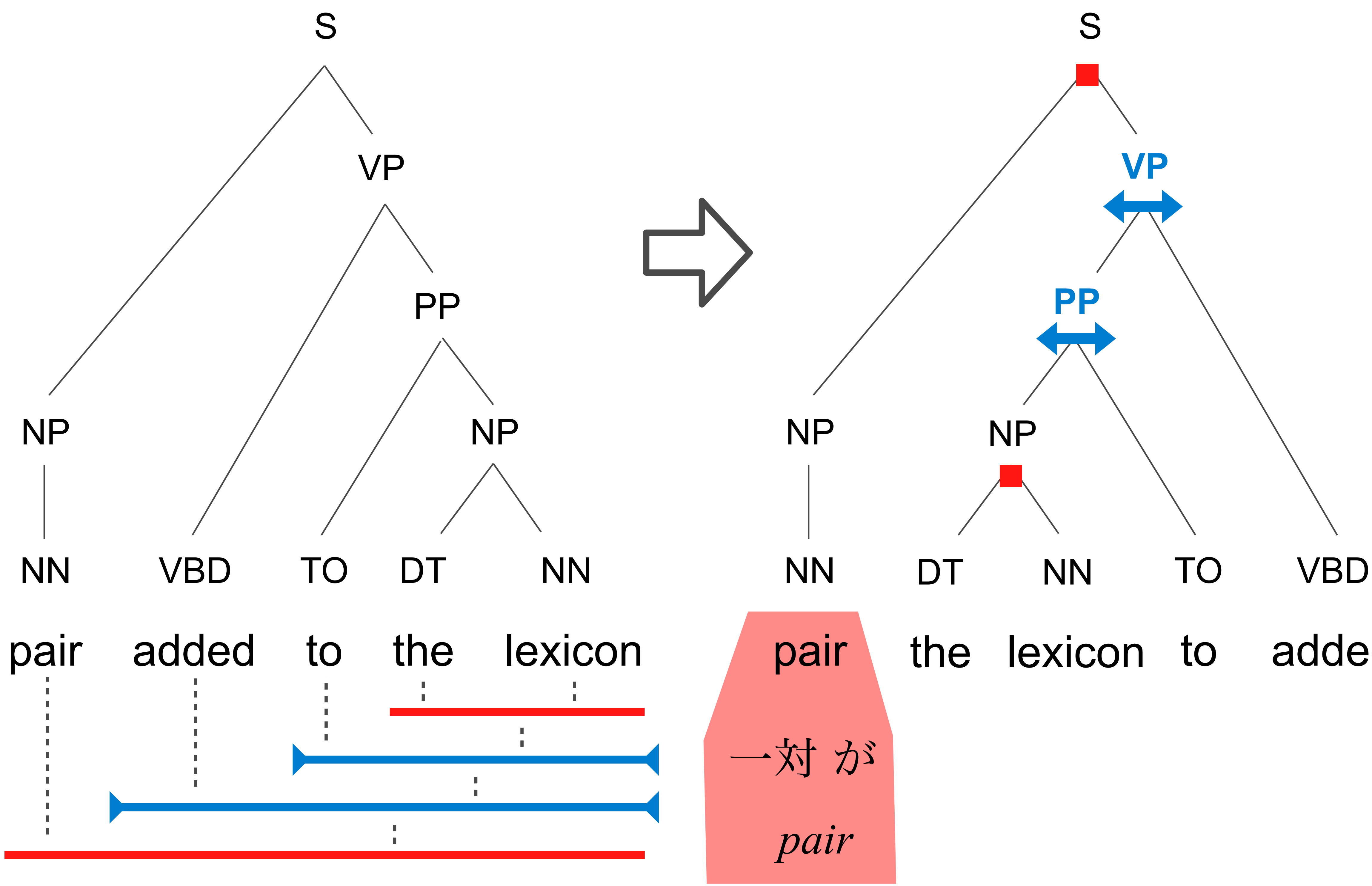
Syntactic Pre-Ordering Example



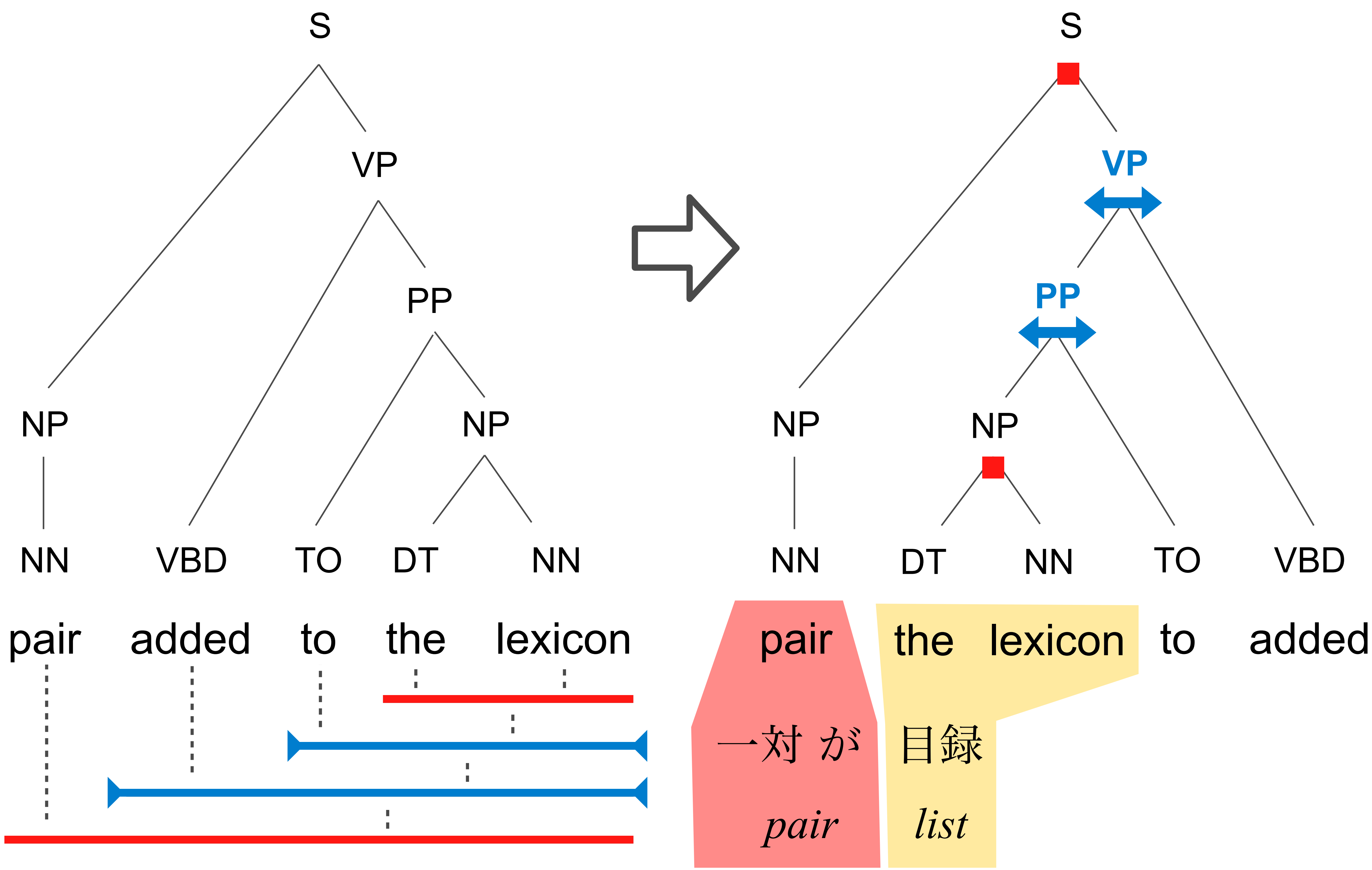
Syntactic Pre-Ordering Example



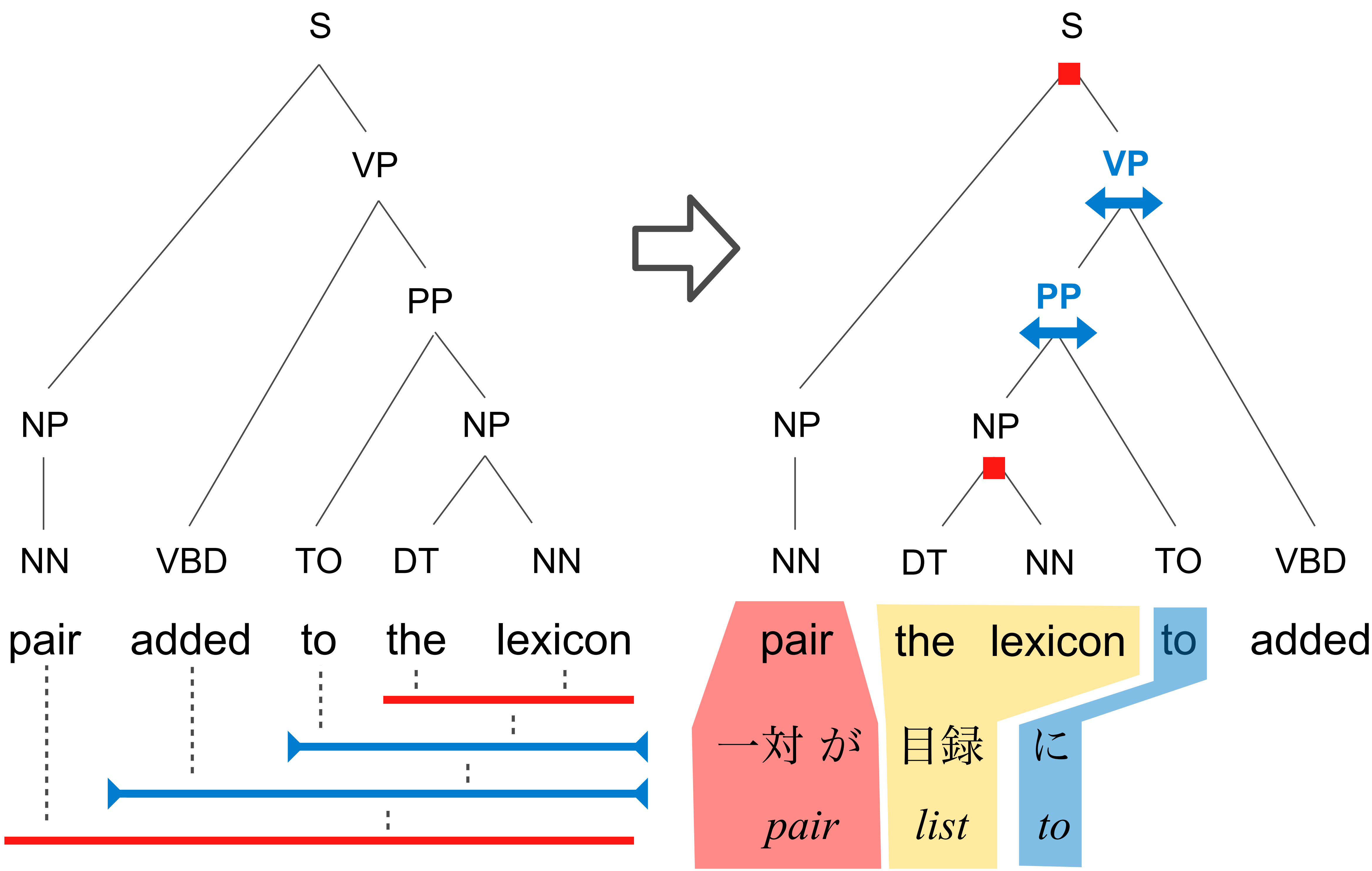
Syntactic Pre-Ordering Example



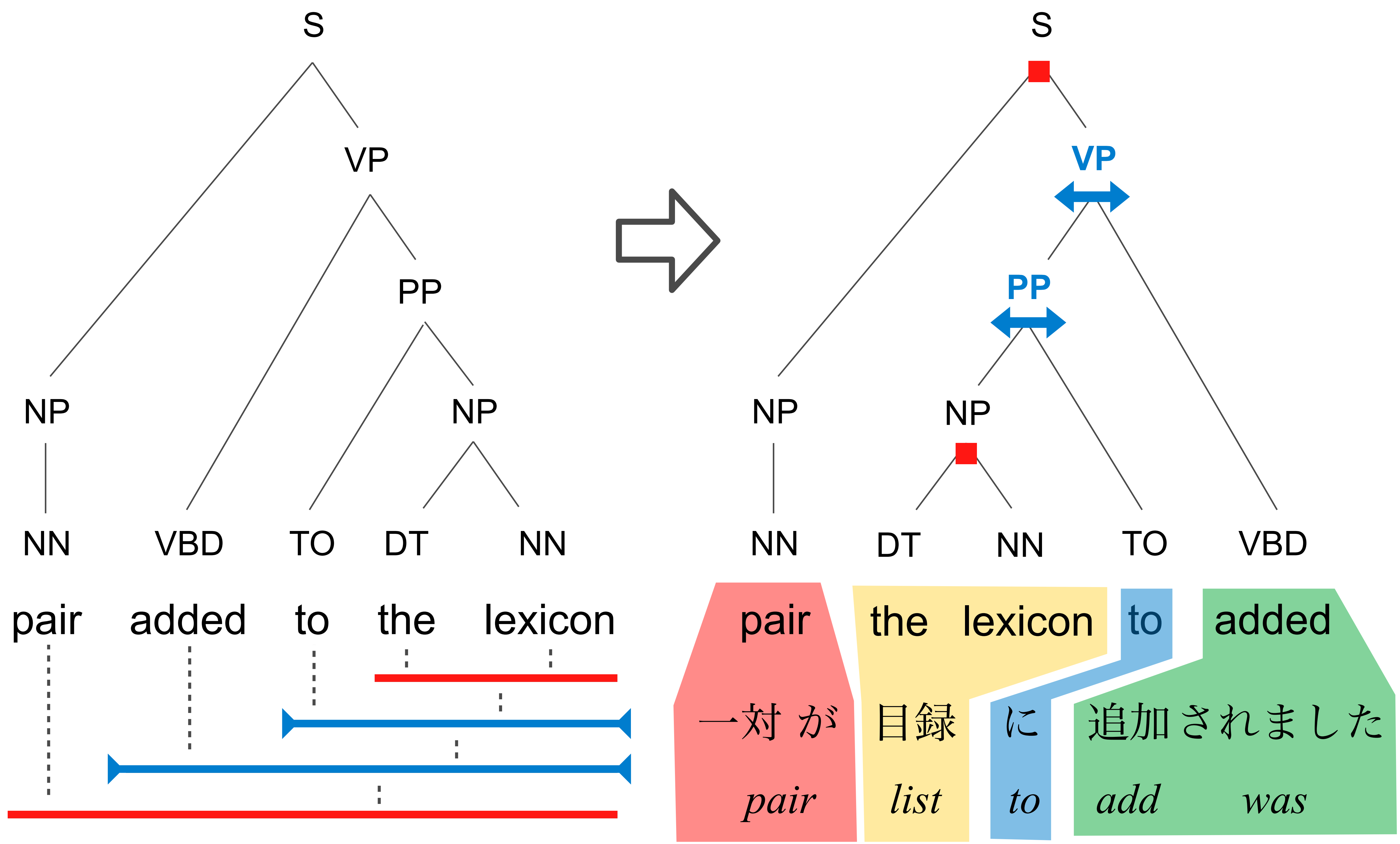
Syntactic Pre-Ordering Example



Syntactic Pre-Ordering Example



Syntactic Pre-Ordering Example



Bracketing and Word Classes are Often Sufficient

Decisions | **Methods**

pair added to the lexicon

Bracketing and Word Classes are Often Sufficient

Decisions	Methods

N	V	A	D	N
pair	added	to	the	lexicon

Bracketing and Word Classes are Often Sufficient

Decisions

Methods

Universal Parts-of-speech
(Petrov et al., 2011)

N	V	A	D	N
pair	added	to	the	lexicon

Bracketing and Word Classes are Often Sufficient

Decisions

Universal Parts-of-speech
(Petrov et al., 2011)

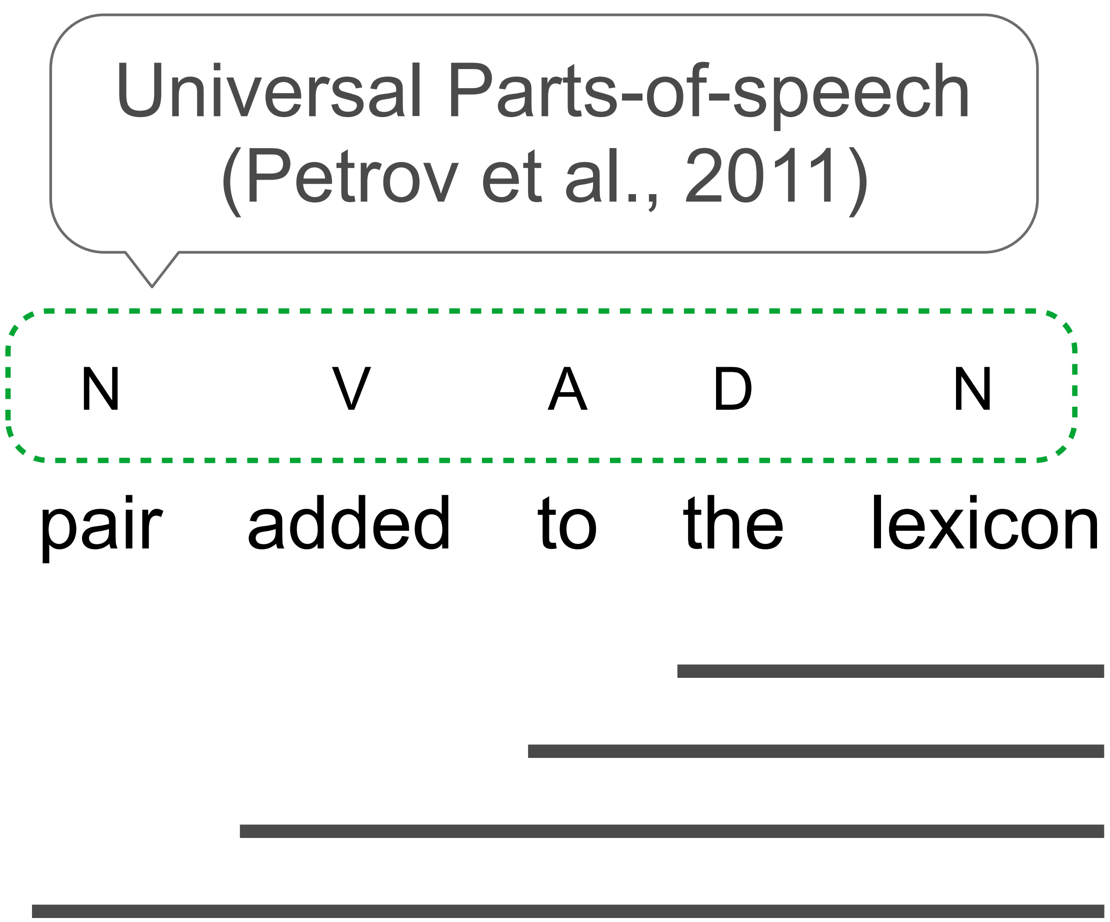
N	V	A	D	N
pair	added	to	the	lexicon

Methods

- Supervised tagging model
- Project models via alignments
- Unsupervised POS induction

Bracketing and Word Classes are Often Sufficient

Decisions

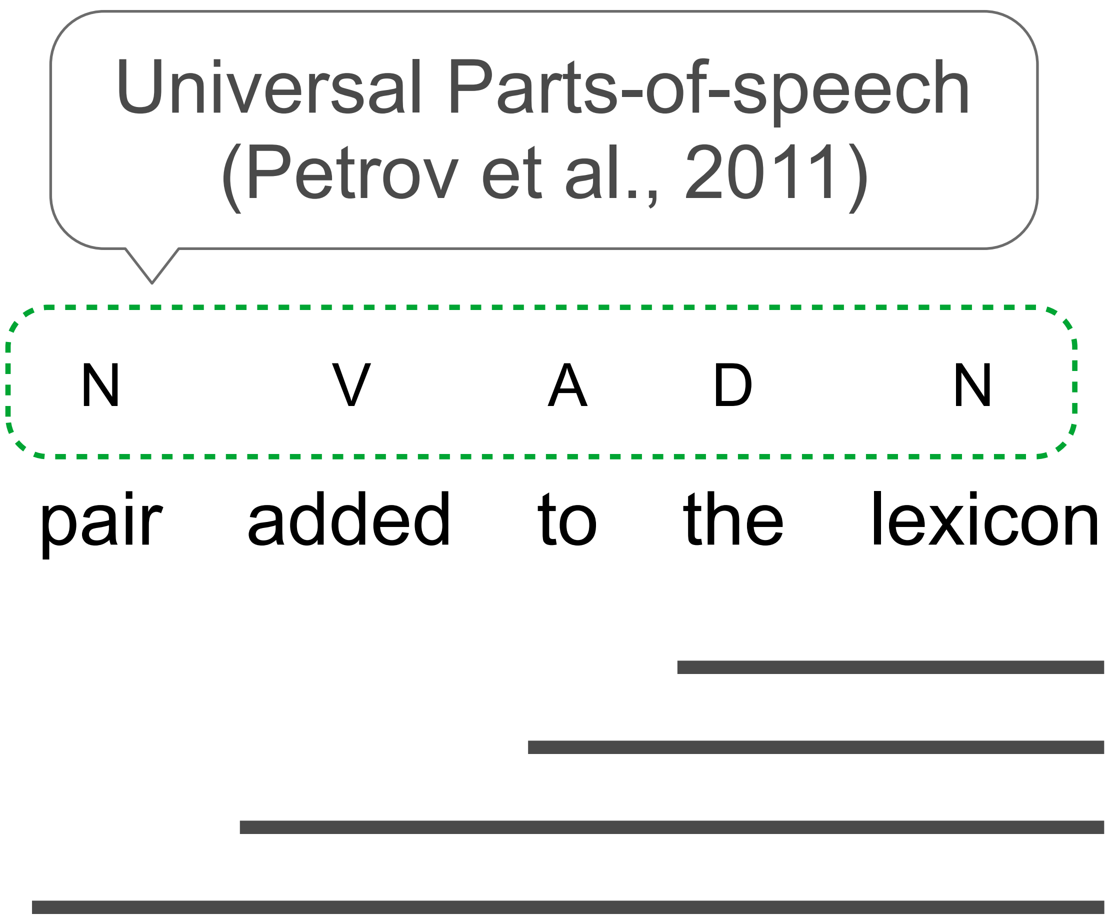


Methods

- Supervised tagging model
- Project models via alignments
- Unsupervised POS induction

Bracketing and Word Classes are Often Sufficient

Decisions



Methods

- Supervised tagging model
- Project models via alignments
- Unsupervised POS induction

- Alignments ≈ bracketing
- Discriminative bracketing model

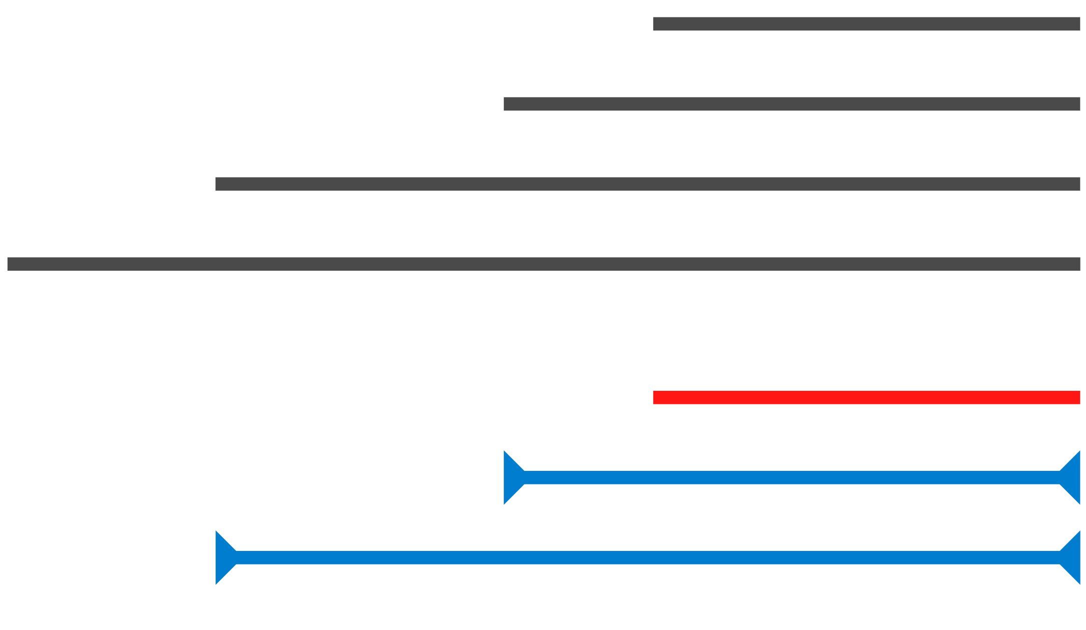
Bracketing and Word Classes are Often Sufficient

Decisions

Universal Parts-of-speech
(Petrov et al., 2011)

N V A D N

pair added to the lexicon



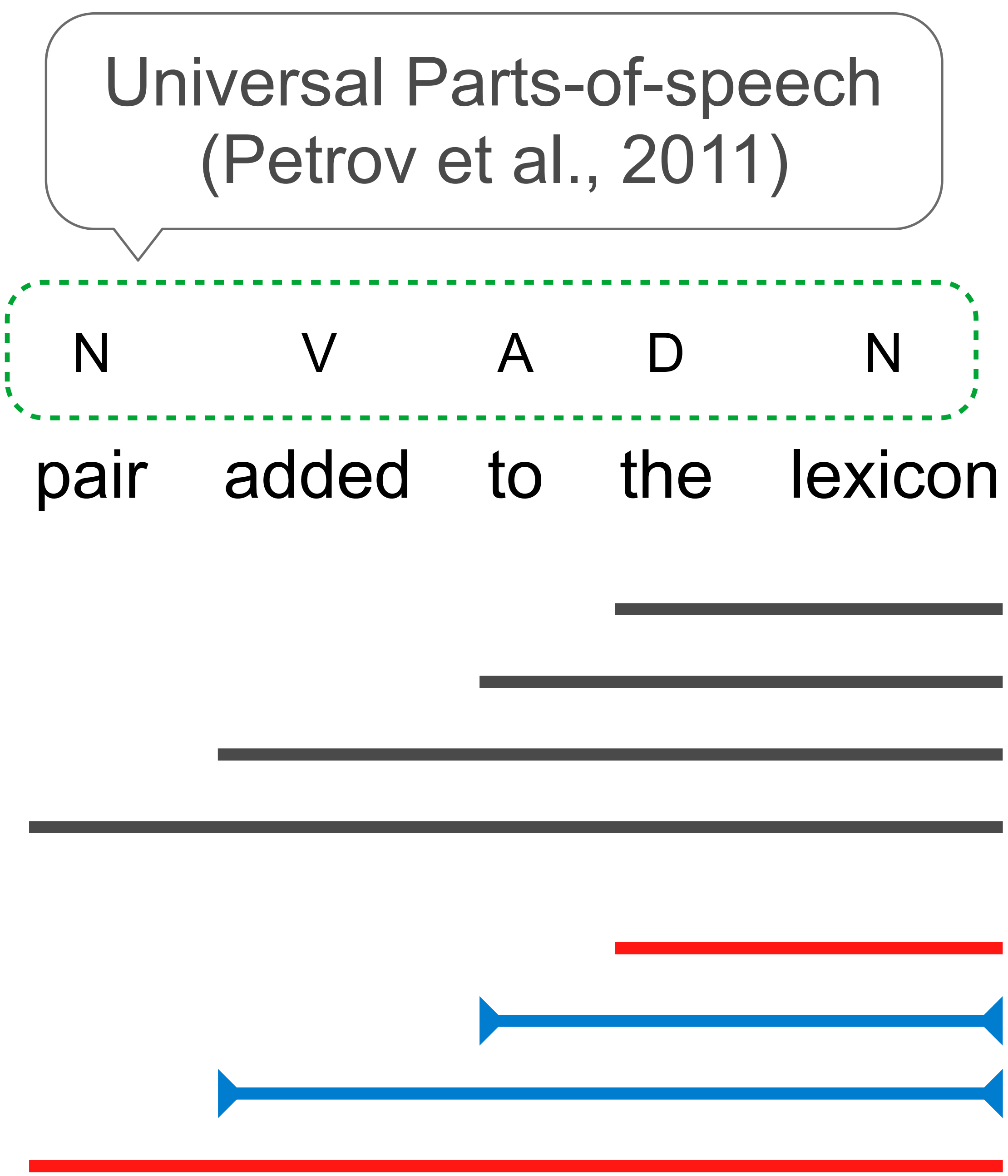
Methods

- Supervised tagging model
- Project models via alignments
- Unsupervised POS induction

- Alignments ≈ bracketing
- Discriminative bracketing model

Bracketing and Word Classes are Often Sufficient

Decisions



Methods

- Supervised tagging model
- Project models via alignments
- Unsupervised POS induction

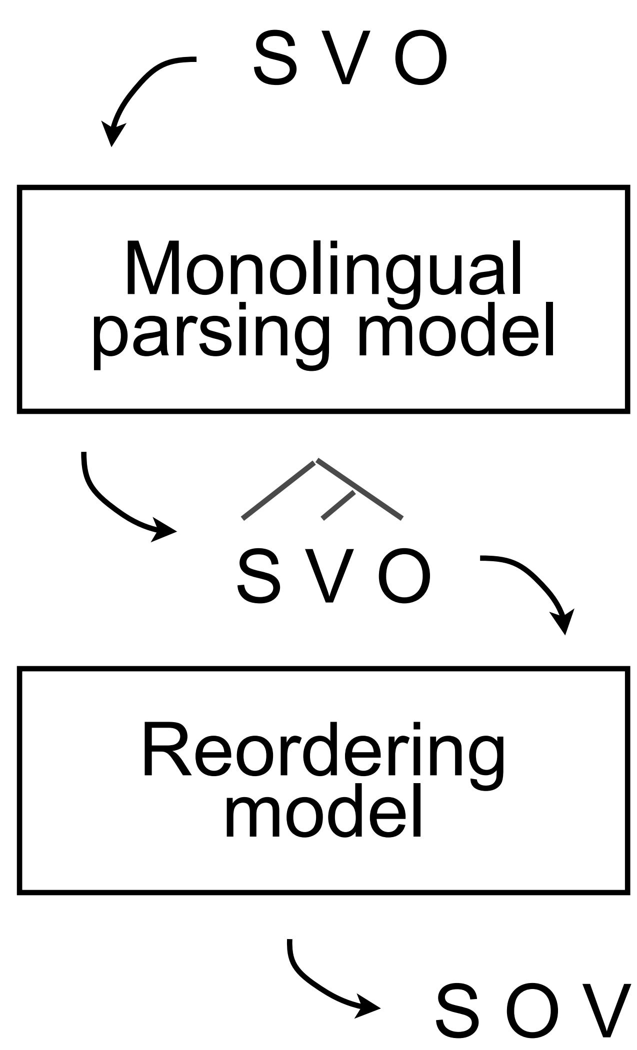
- Alignments ≈ bracketing
- Discriminative bracketing model

- Alignments gives reordering
- Reordering classifier

Pre-Ordering from a Parallel Corpus

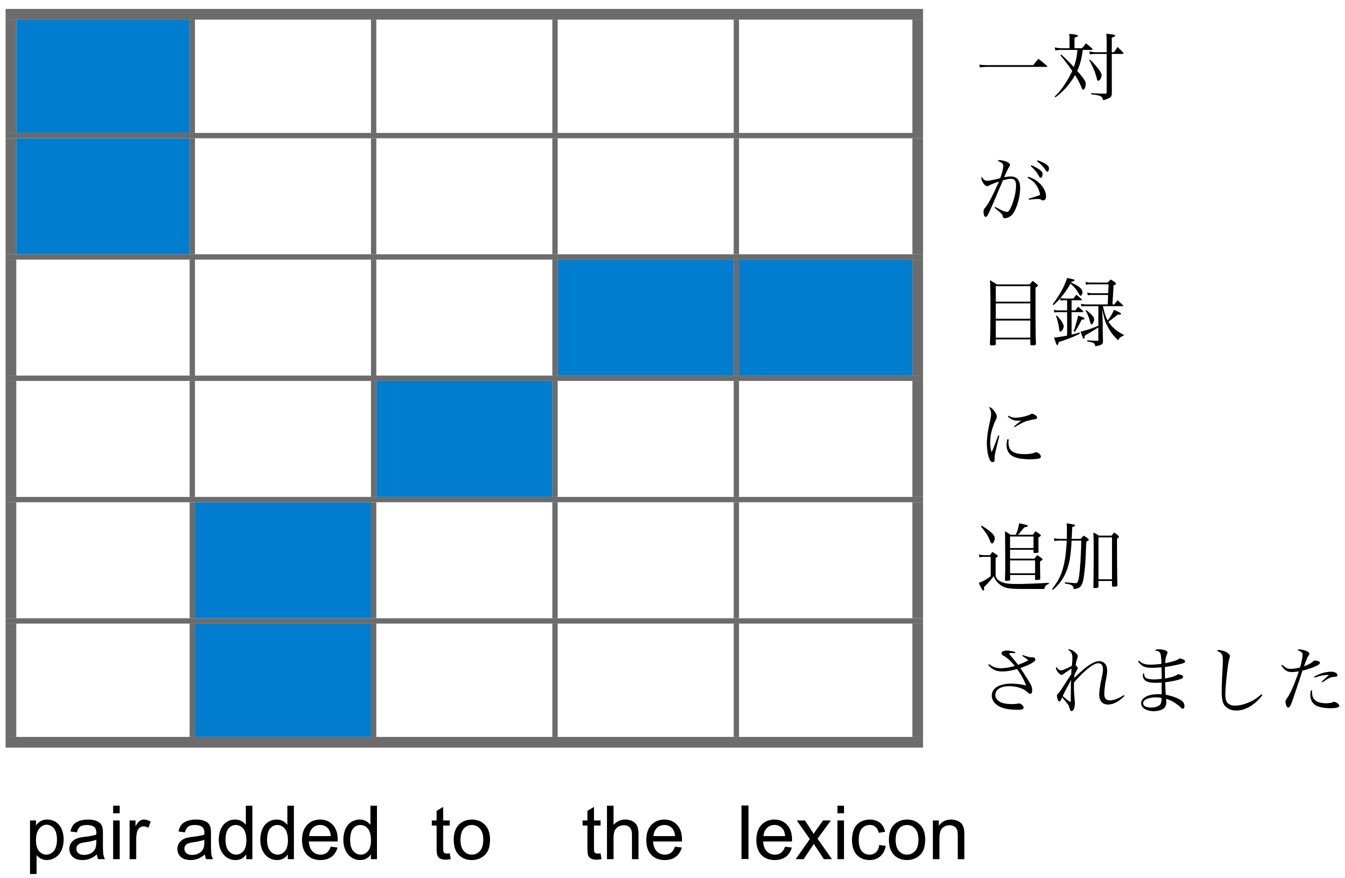
Analyze Aligned Parallel Corpus

Pre-Ordering Pipeline

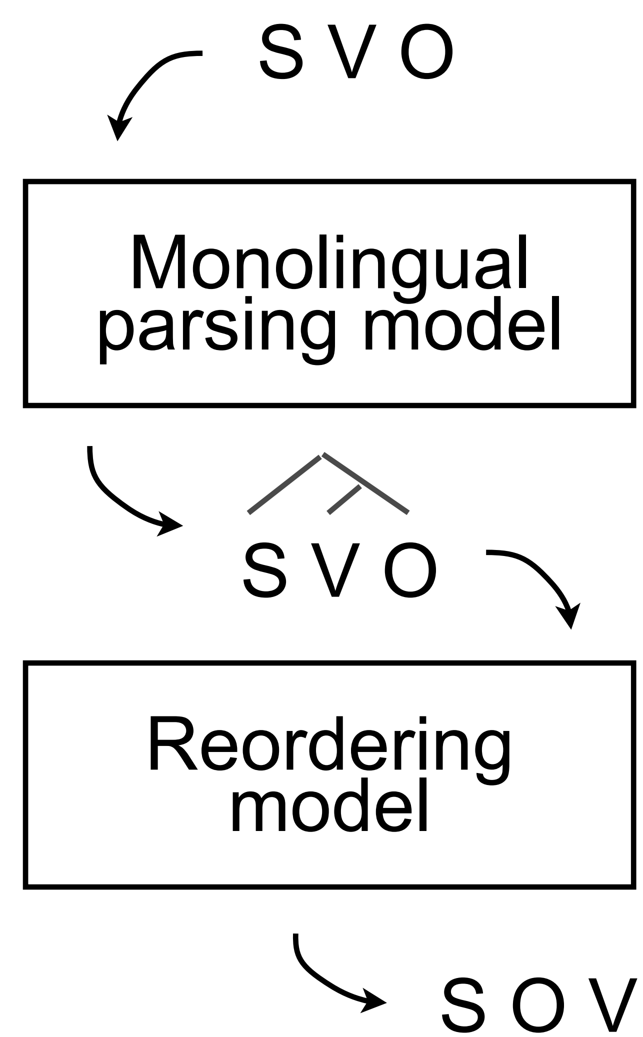


Pre-Ordering from a Parallel Corpus

Analyze Aligned Parallel Corpus

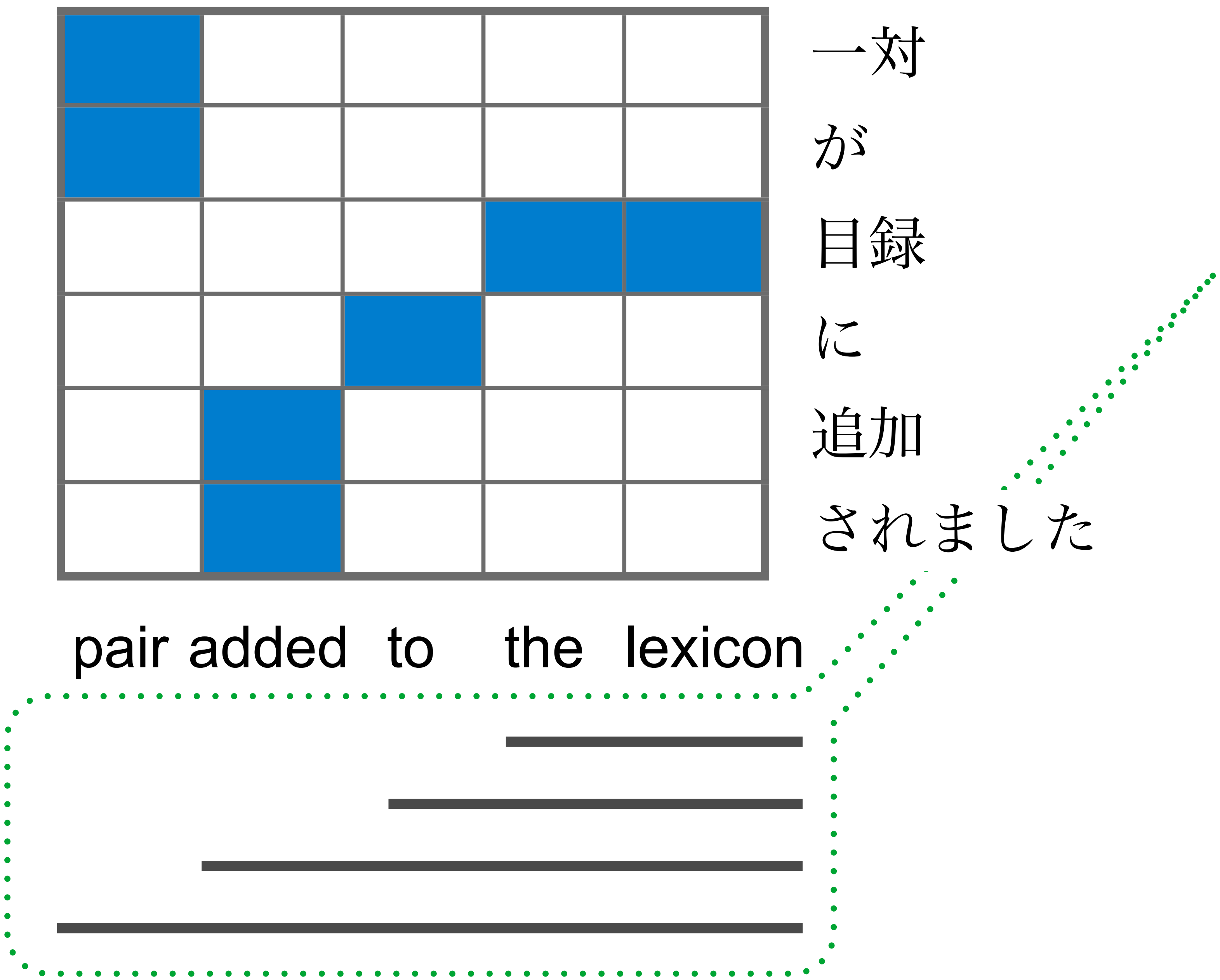


Pre-Ordering Pipeline

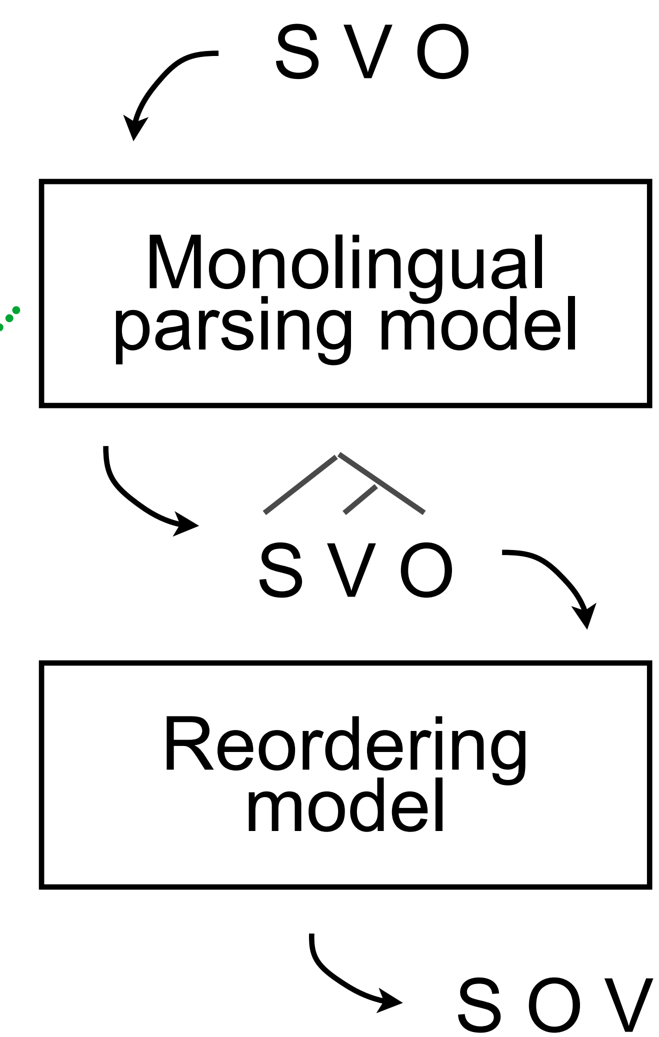


Pre-Ordering from a Parallel Corpus

Analyze Aligned Parallel Corpus

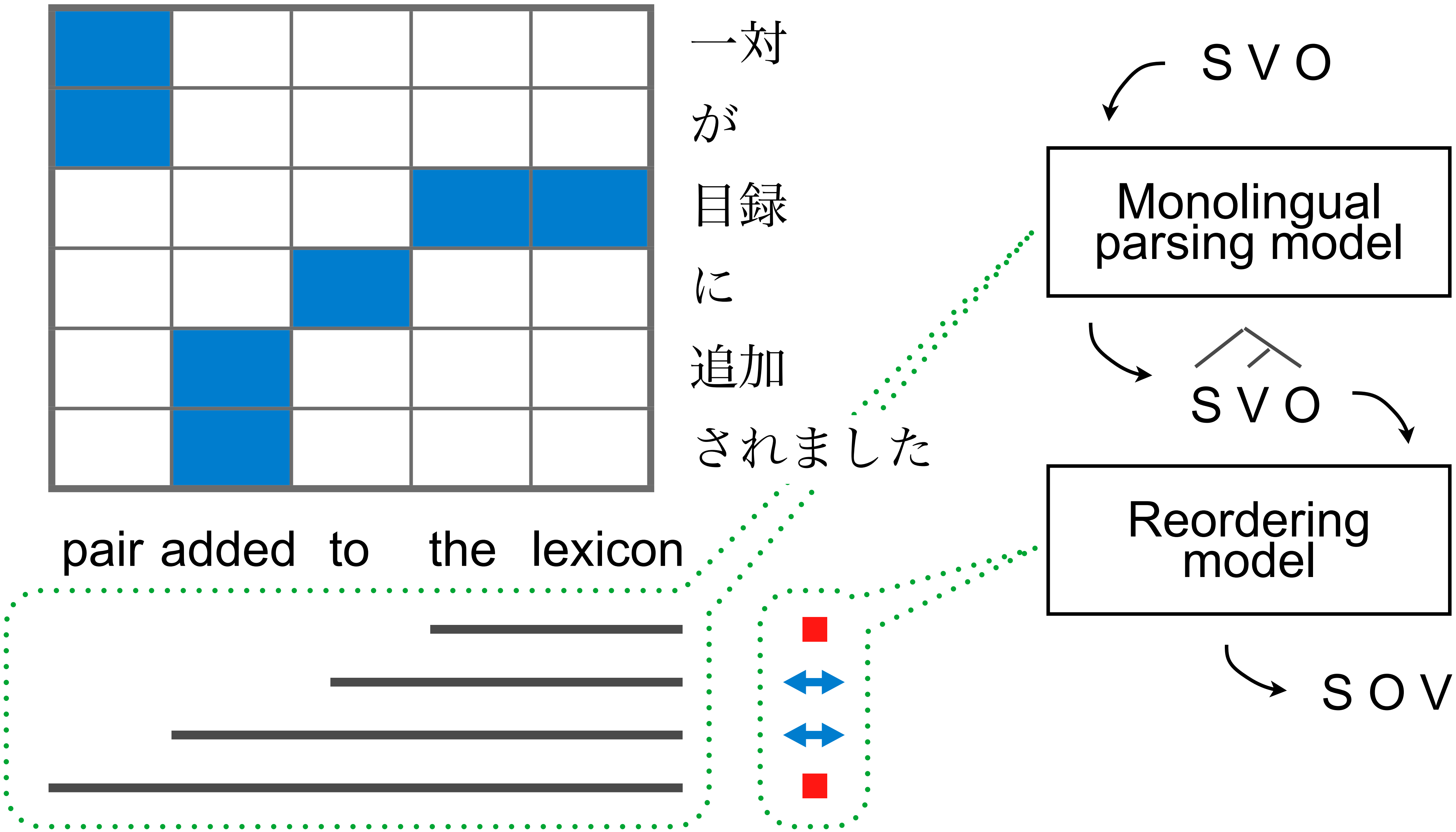


Pre-Ordering Pipeline

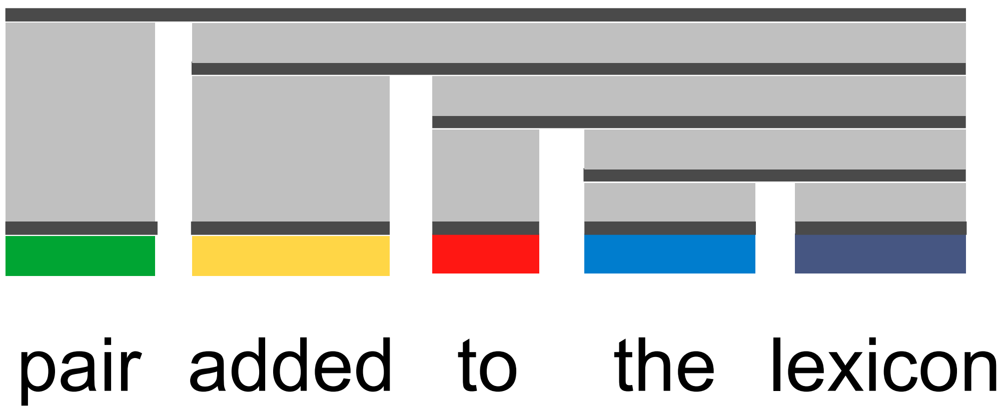


Pre-Ordering from a Parallel Corpus

Analyze Aligned Parallel Corpus | **Pre-Ordering Pipeline**

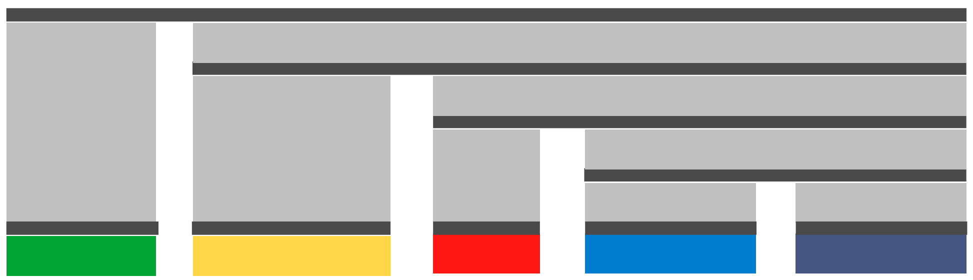


Binary Bracketing Trees with Phrasal Terminals



A tree is defined by the set of spans it contains

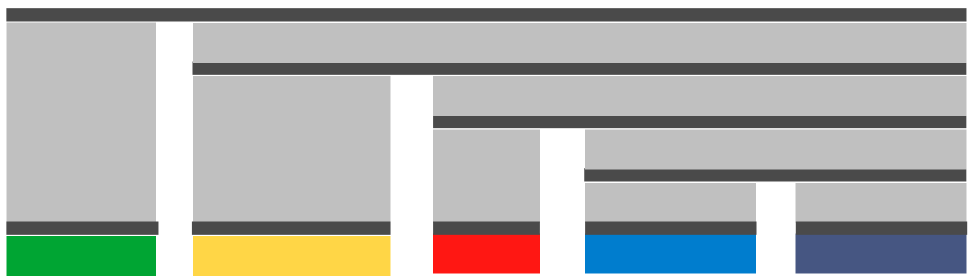
Binary Bracketing Trees with Phrasal Terminals



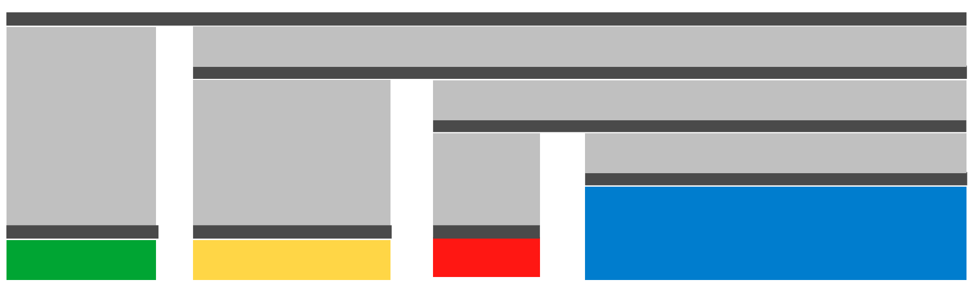
A tree is defined by the set of spans it contains

pair added to the lexicon

Binary Bracketing Trees with Phrasal Terminals



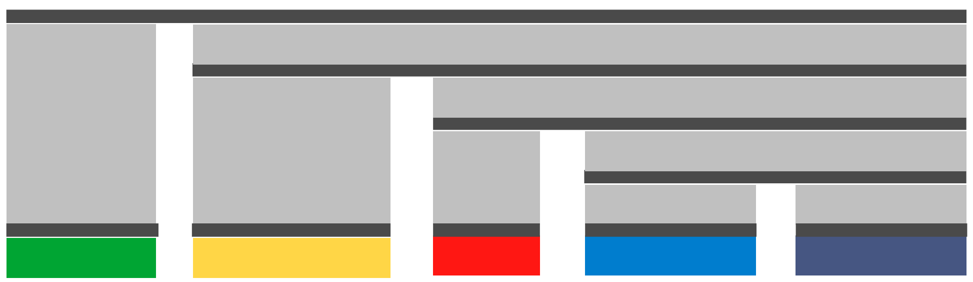
A tree is defined by the set of spans it contains



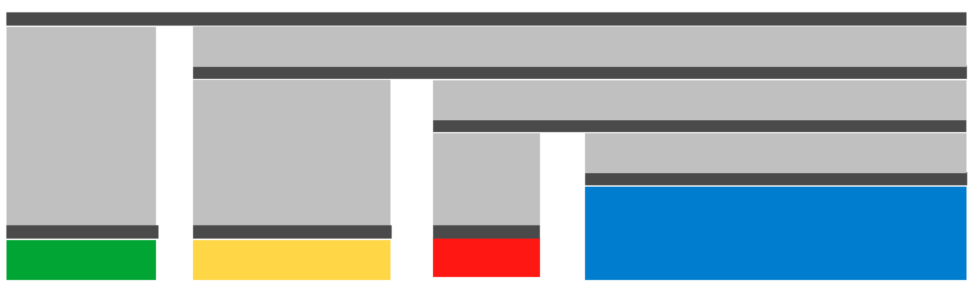
Not every word will correspond to a span

pair added to the lexicon

Binary Bracketing Trees with Phrasal Terminals



A tree is defined by the set of spans it contains

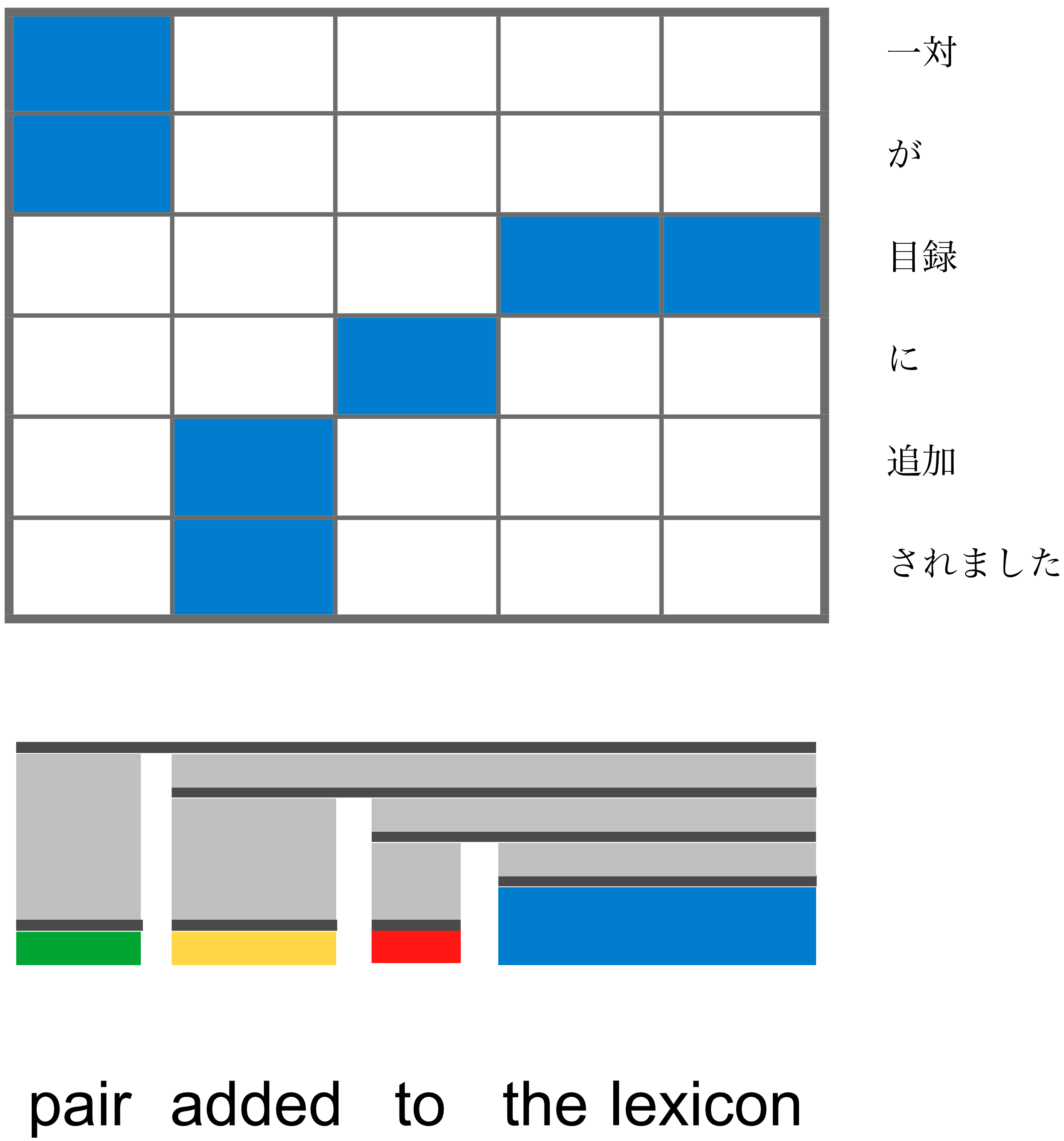


Not every word will correspond to a span

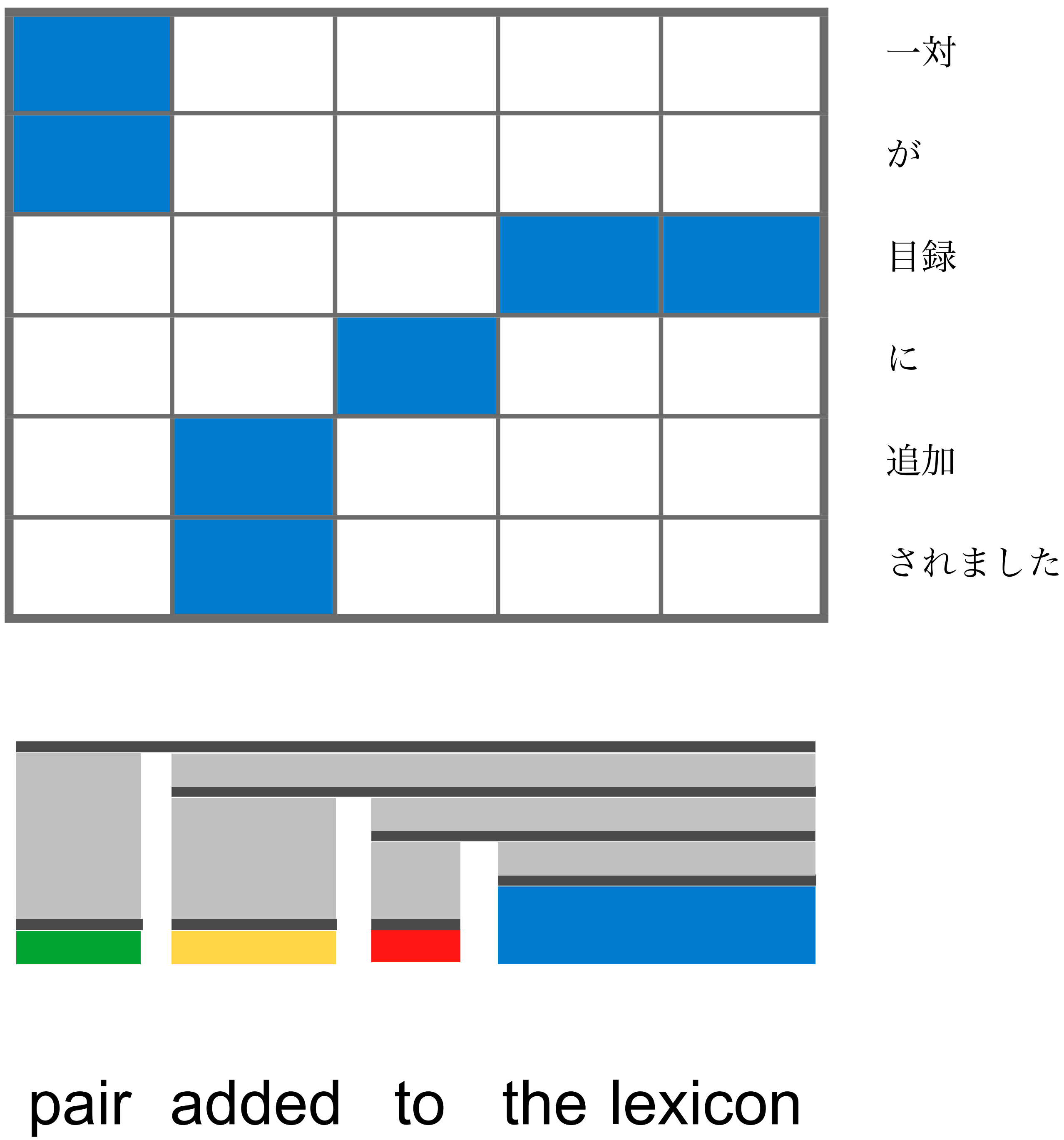
A tree can be composed of only a single span

pair added to the lexicon

Parallel Parsing Model Constraint

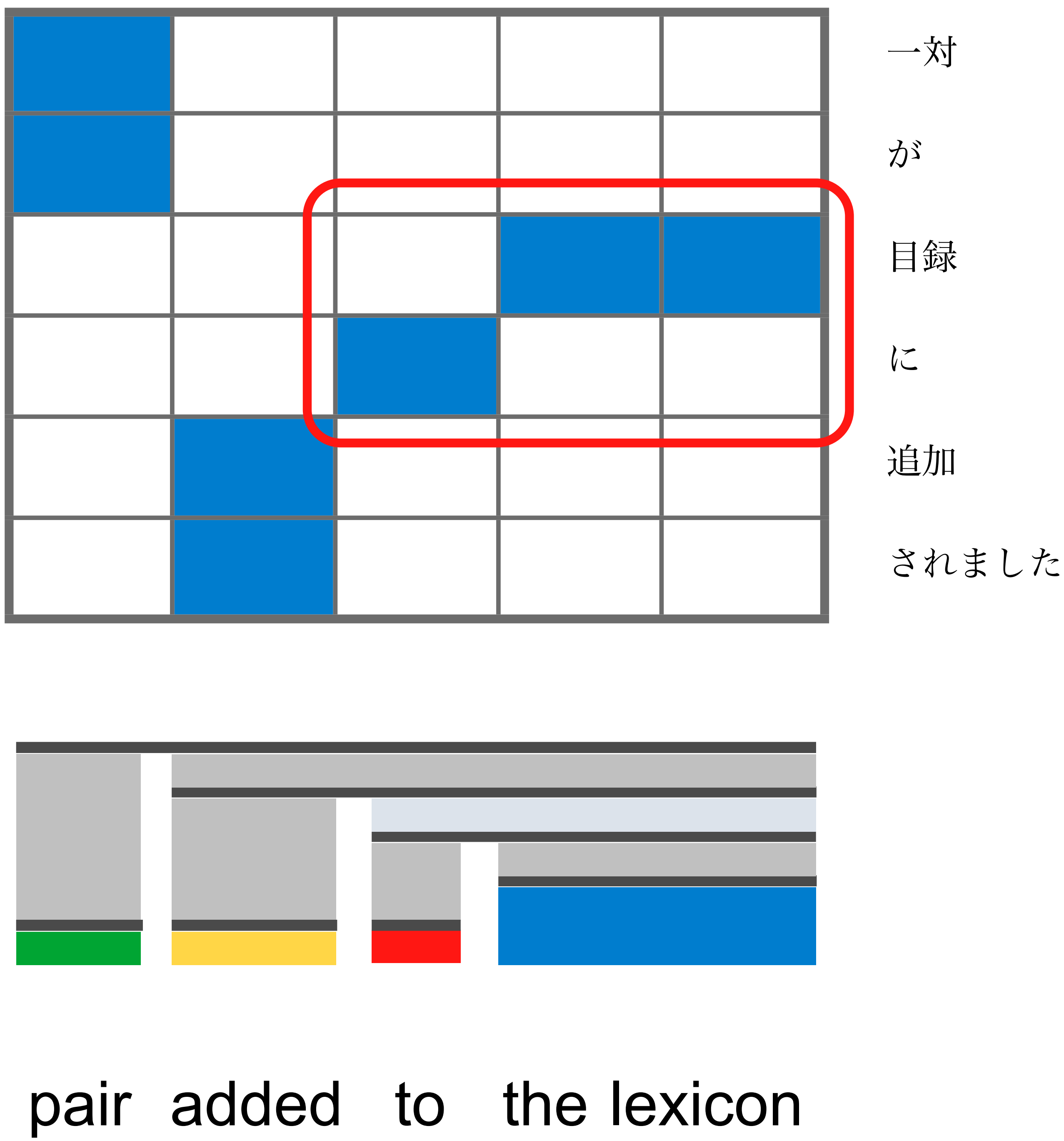


Parallel Parsing Model Constraint



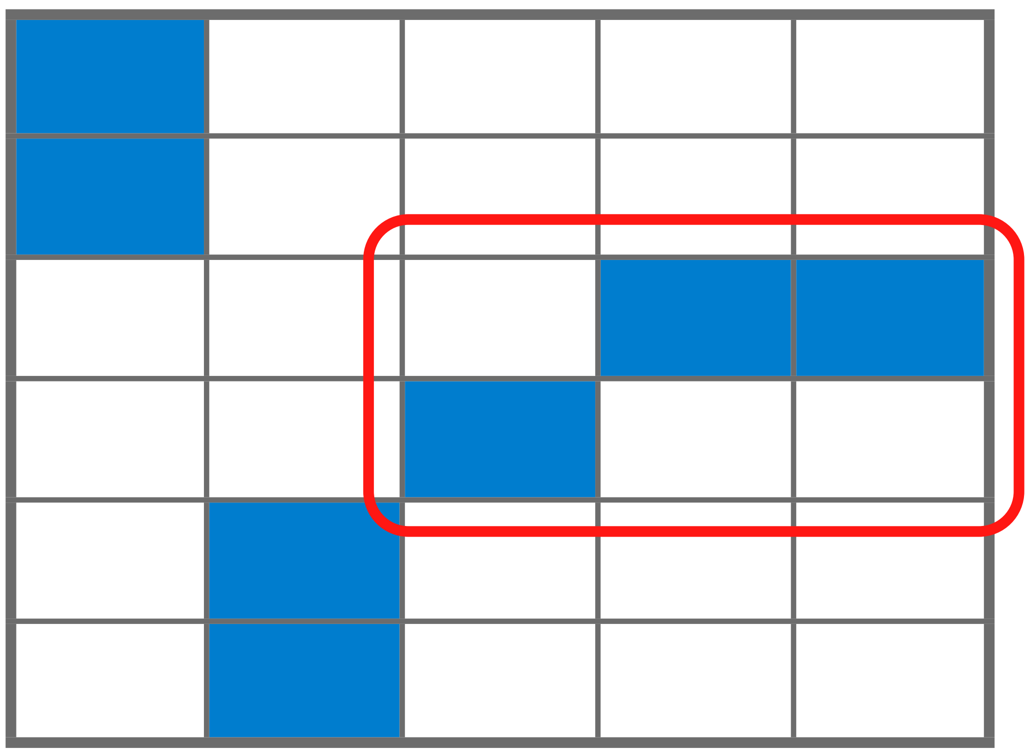
An alignment licenses a tree if every tree span is *aligned contiguously*

Parallel Parsing Model Constraint



An alignment licenses a tree if every tree span is *aligned contiguously*

Parallel Parsing Model Constraint



一対

が

目録

に

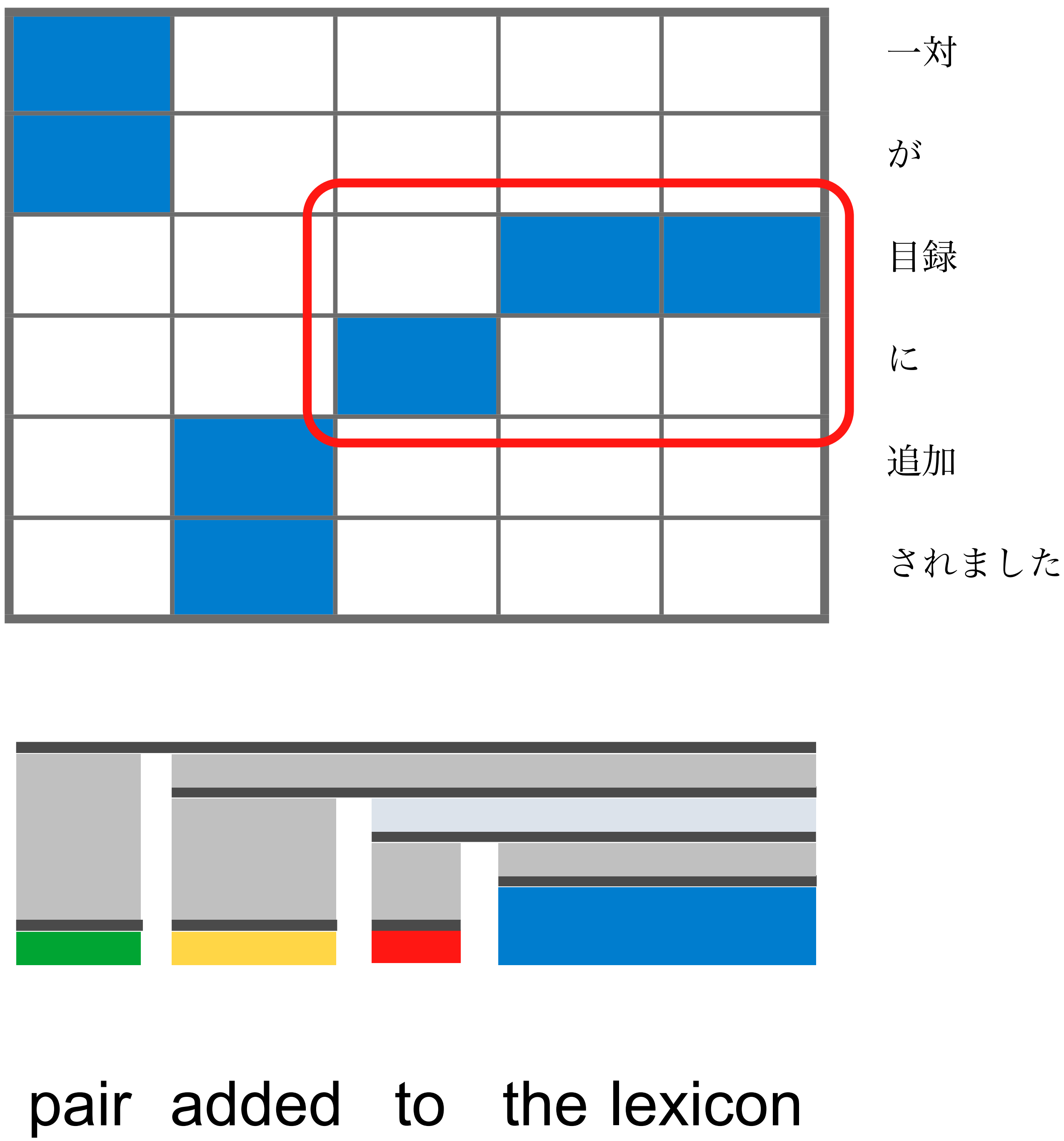
追加

されました

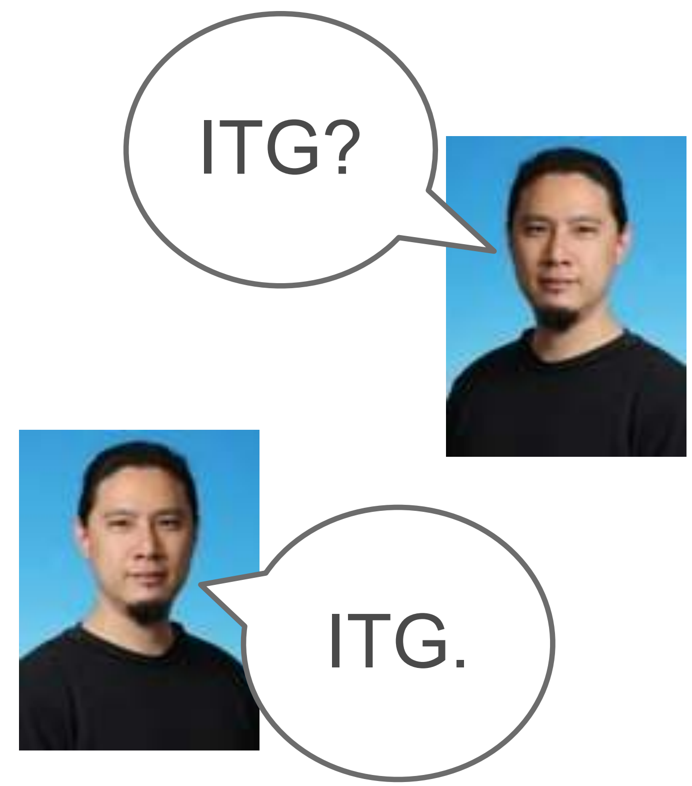
pair added to the lexicon

An alignment licenses a tree if every tree span is *aligned contiguously*

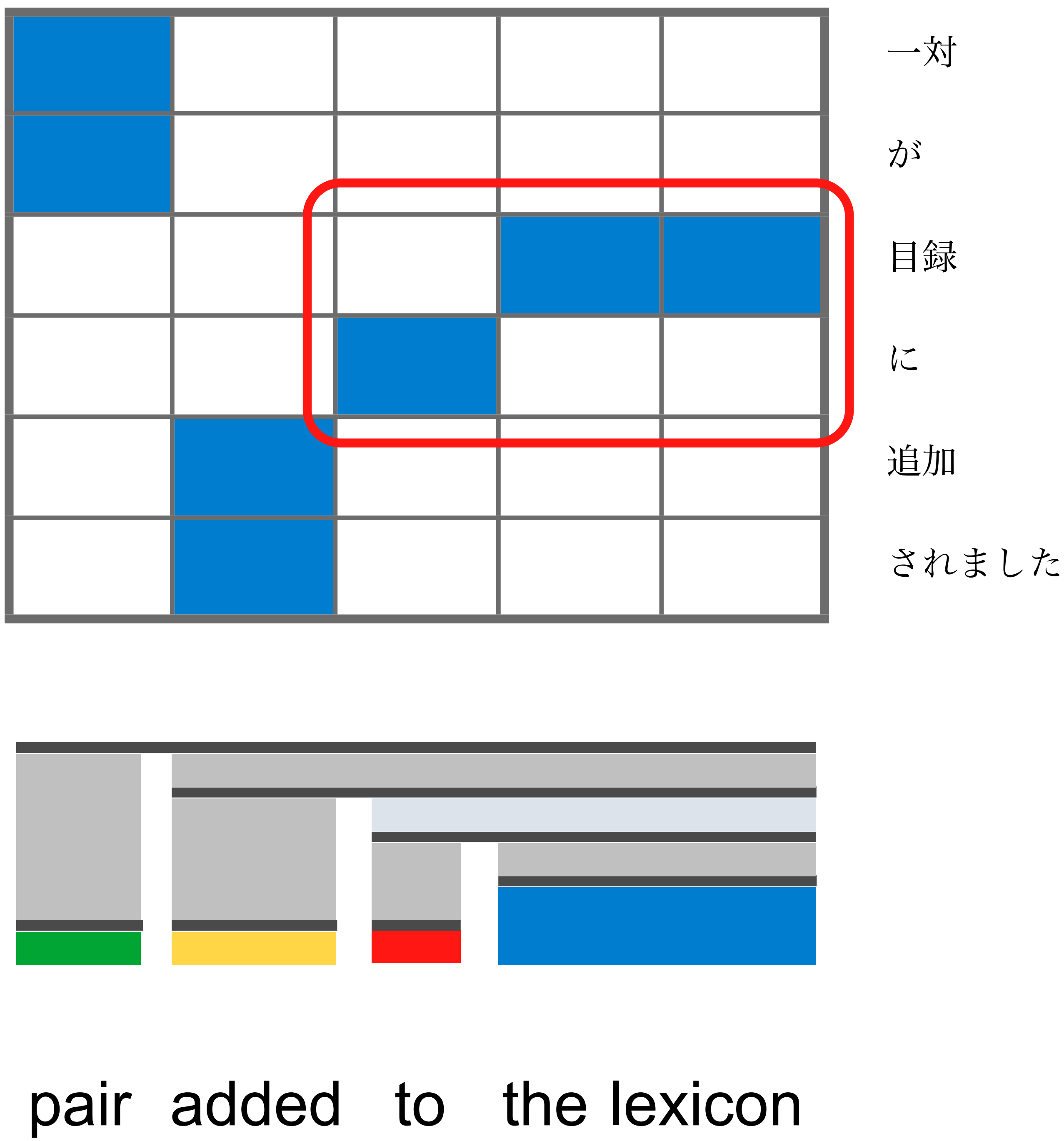
Parallel Parsing Model Constraint



An alignment licenses a tree if every tree span is *aligned contiguously*



Parallel Parsing Model Constraint

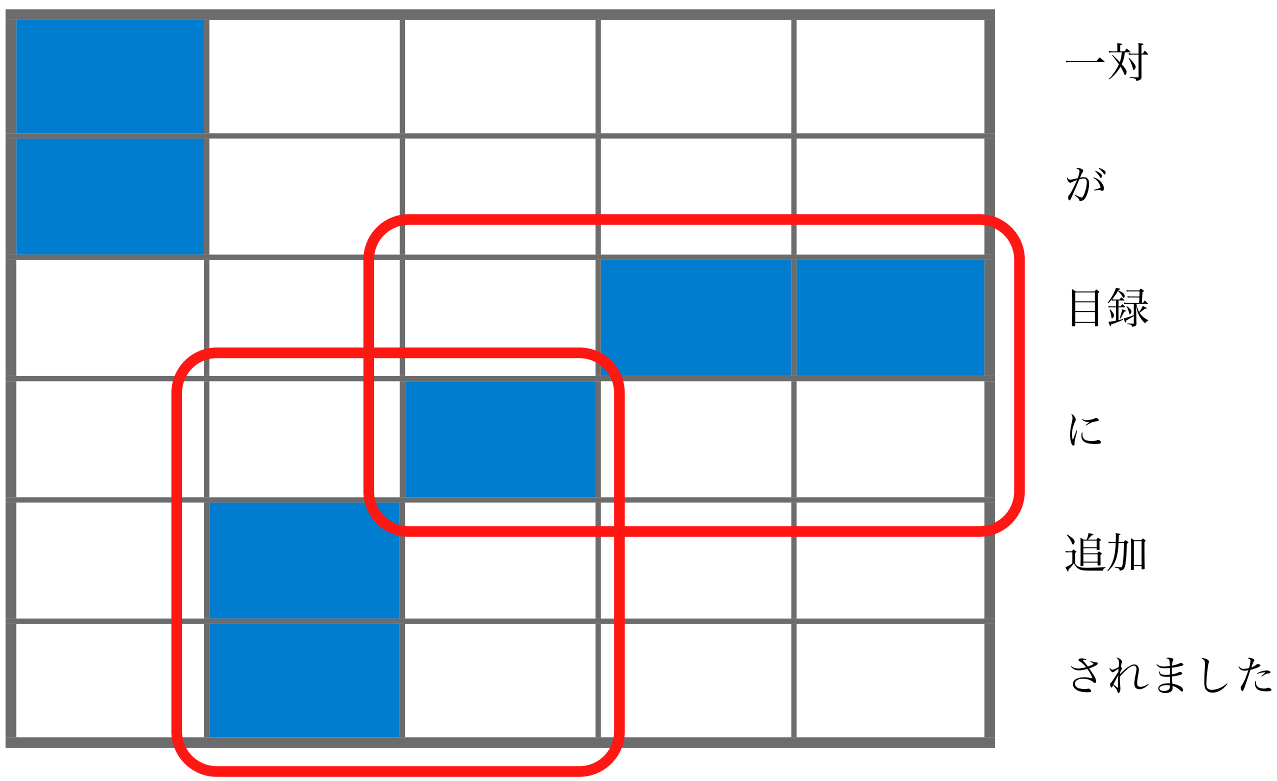


An alignment licenses a tree if every tree span is *aligned contiguously*

Remaining ambiguity:

- Ordering alternatives

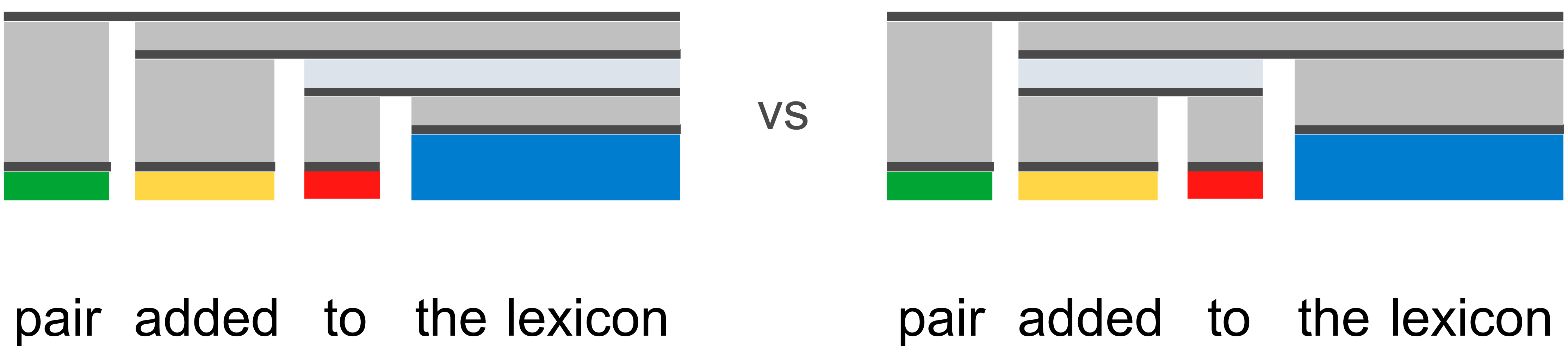
Parallel Parsing Model Constraint



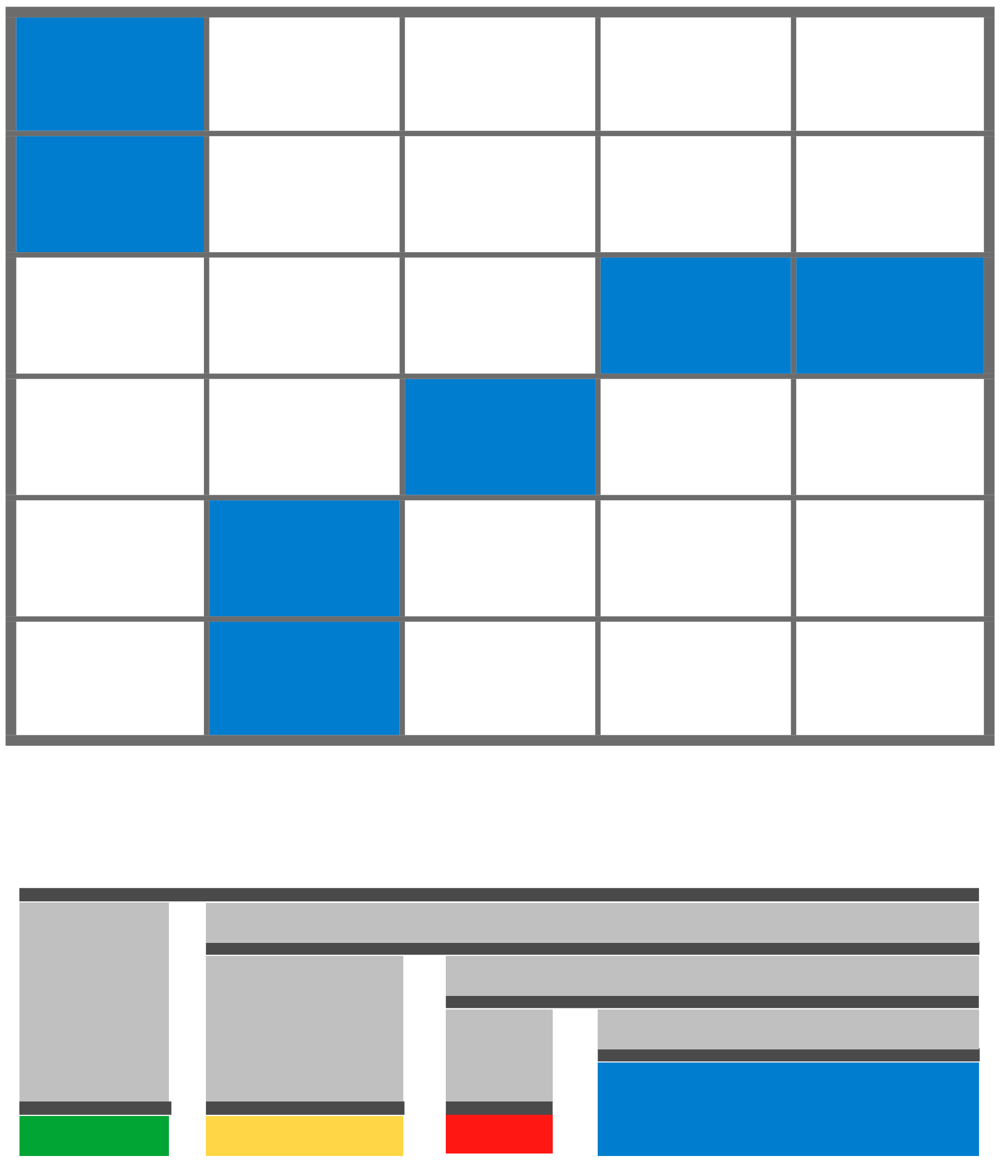
An alignment licenses a tree if every tree span is *aligned contiguously*

Remaining ambiguity:

- Ordering alternatives



Parallel Parsing Model Constraint

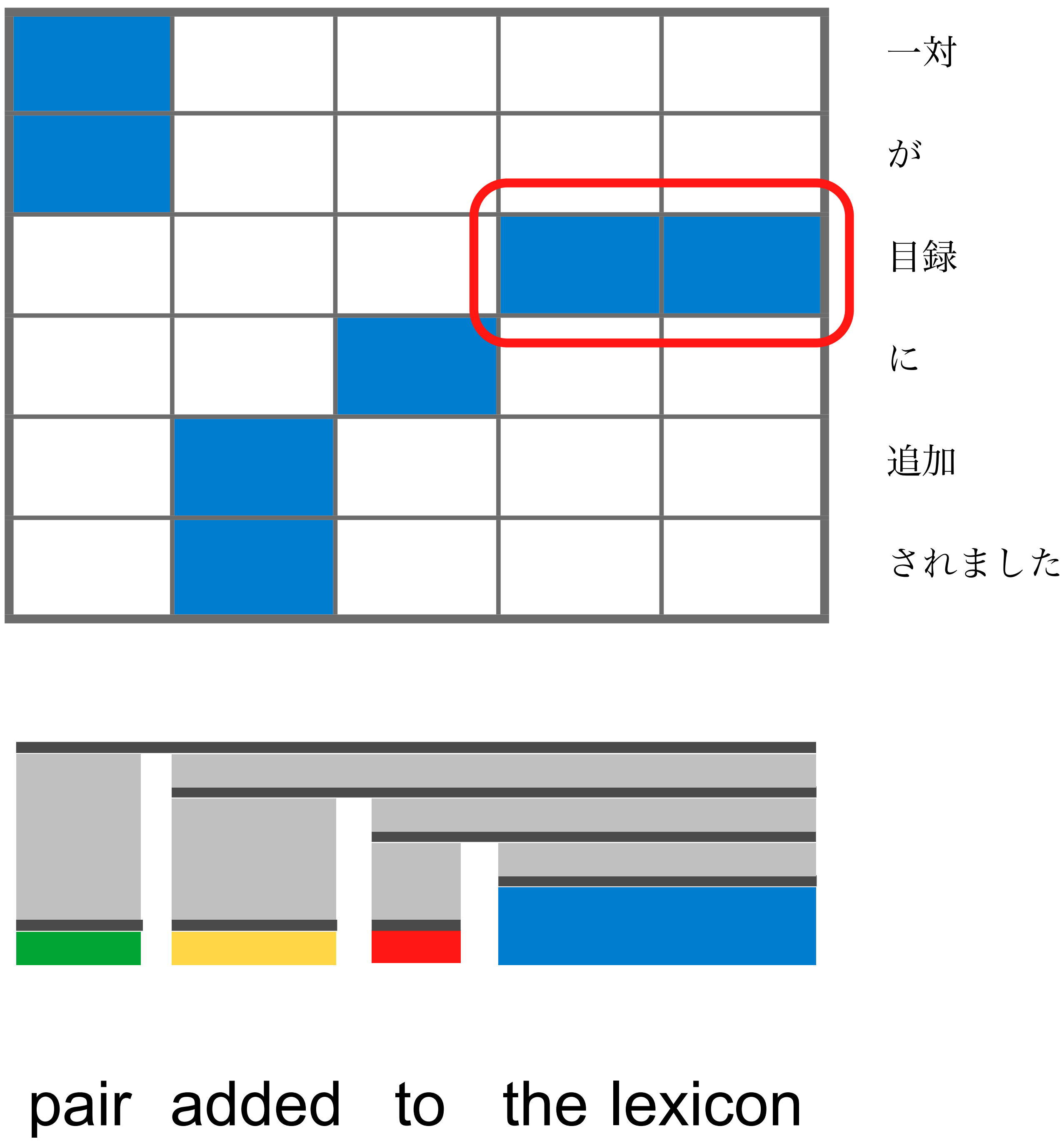


An alignment licenses a tree if every tree span is *aligned contiguously*

Remaining ambiguity:

- Ordering alternatives
- Phrase granularity

Parallel Parsing Model Constraint

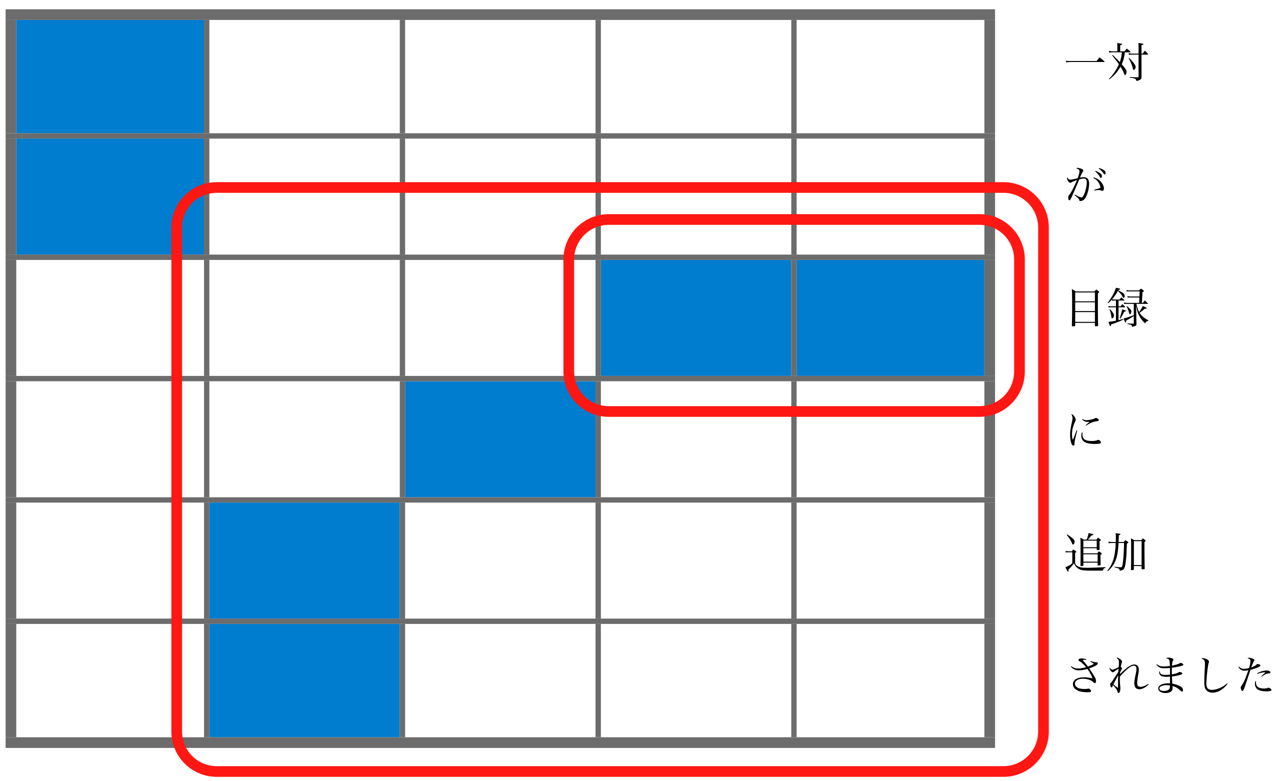


An alignment licenses a tree if every tree span is *aligned contiguously*

Remaining ambiguity:

- Ordering alternatives
- Phrase granularity

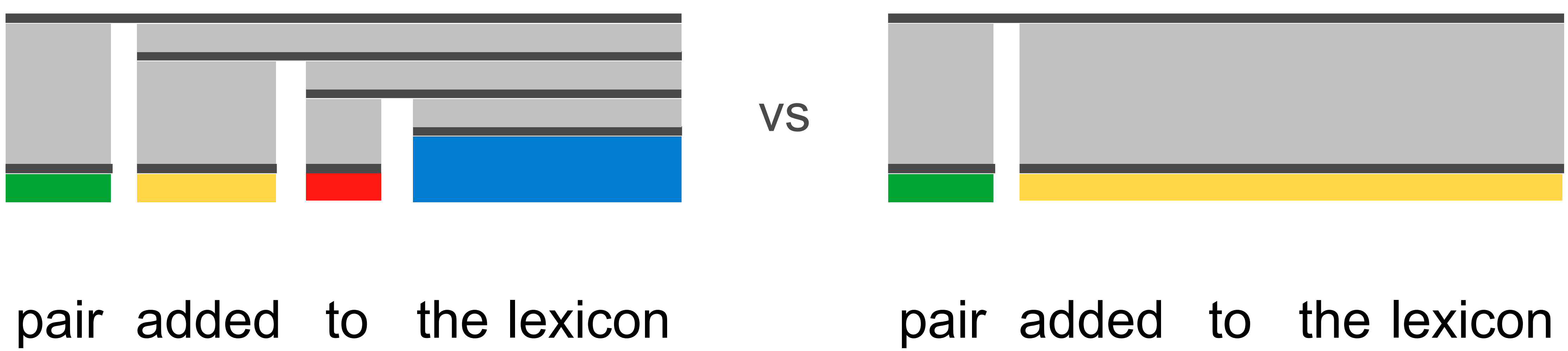
Parallel Parsing Model Constraint



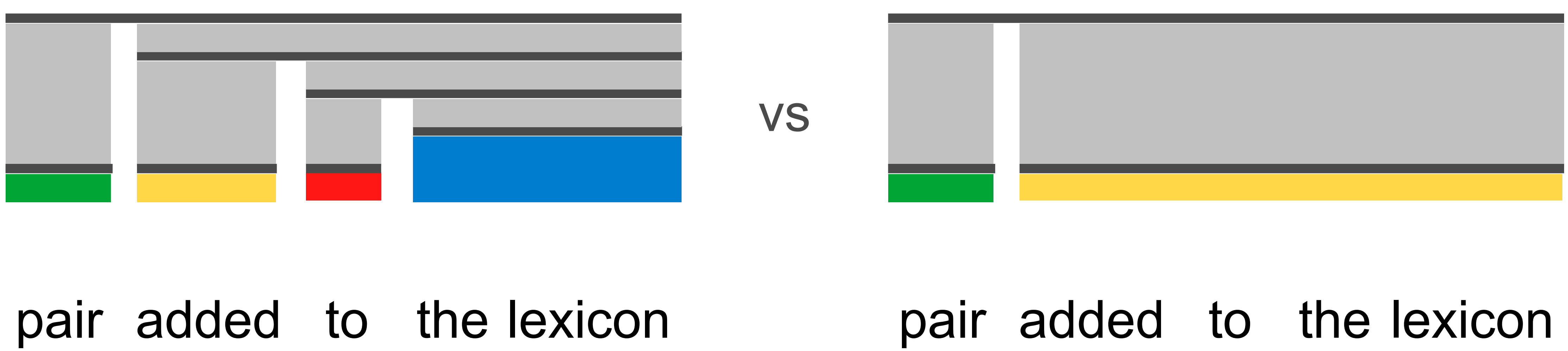
An alignment licenses a tree if every tree span is *aligned contiguously*

Remaining ambiguity:

- Ordering alternatives
- Phrase granularity

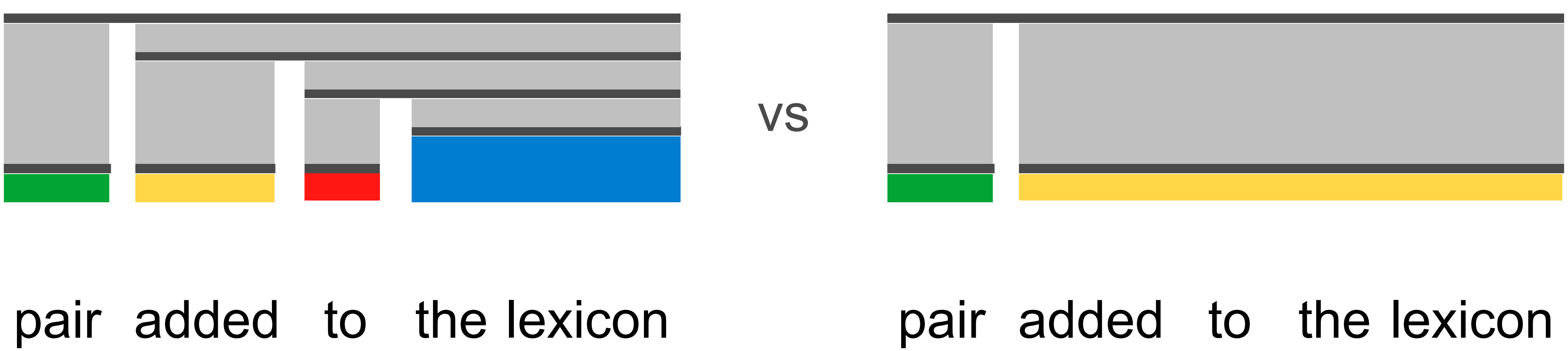


Parallel Parsing Model



Parallel Parsing Model

$$\phi(\text{pair}) \cdot \phi(\text{added})$$
$$\cdot\ \phi(\text{to}) \cdot \phi(\text{the lexicon})$$



Parallel Parsing Model

$$\phi(\text{pair}) \cdot \phi(\text{added}) \cdot \phi(\text{to}) \cdot \phi(\text{the lexicon})$$

$$\phi(\text{pair}) \cdot \phi(\text{added to the lexicon})$$

vs

pair added to the lexicon

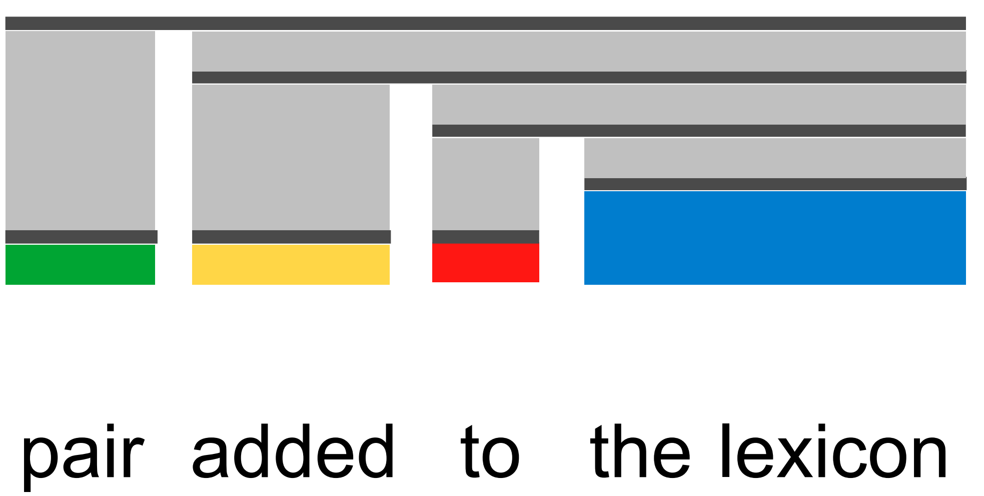
pair added to the lexicon

Parallel Parsing Model

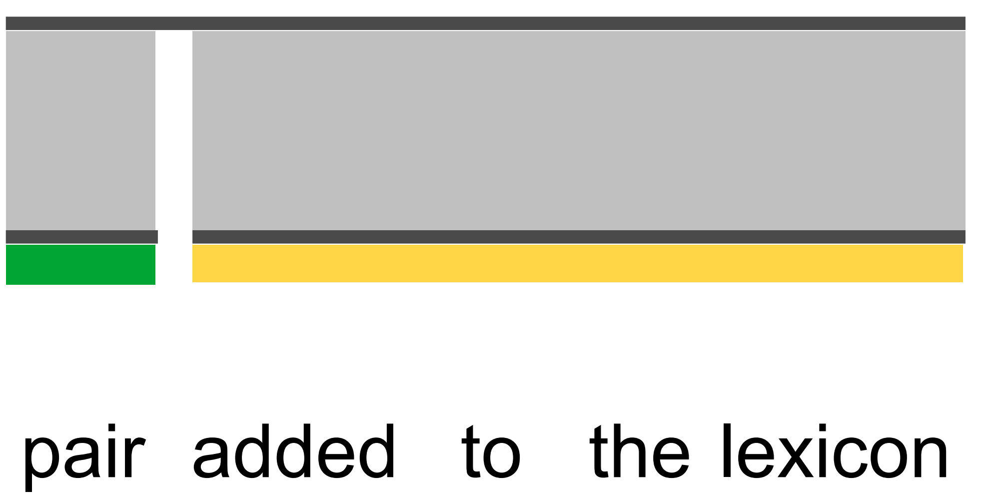
$$\phi(e) = \begin{cases} \kappa & \text{if } |t| = 1 \\ \dfrac{\text{contiguous}(e)}{\text{total}(e)} & \text{otherwise} \end{cases}$$

$$\phi(\text{pair}) \cdot \phi(\text{added}) \cdot \phi(\text{to}) \cdot \phi(\text{the lexicon})$$

$$\phi(\text{pair}) \cdot \phi(\text{added to the lexicon})$$



vs



Parallel Parsing Model

$$\phi(e) = \begin{cases} \kappa & \text{if } |t| = 1 \\ \dfrac{\text{contiguous}(e)}{\text{total}(e)} & \text{otherwise} \end{cases}$$

English-Japanese

$\kappa = 0.3$

$$\phi(\text{pair}) \cdot \phi(\text{added}) \cdot \phi(\text{to}) \cdot \phi(\text{the lexicon})$$

$$\phi(\text{pair}) \cdot \phi(\text{added to the lexicon})$$

pair added to the lexicon

vs

pair added to the lexicon

Monolingual Parsing Model

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features

Monolingual Parsing Model

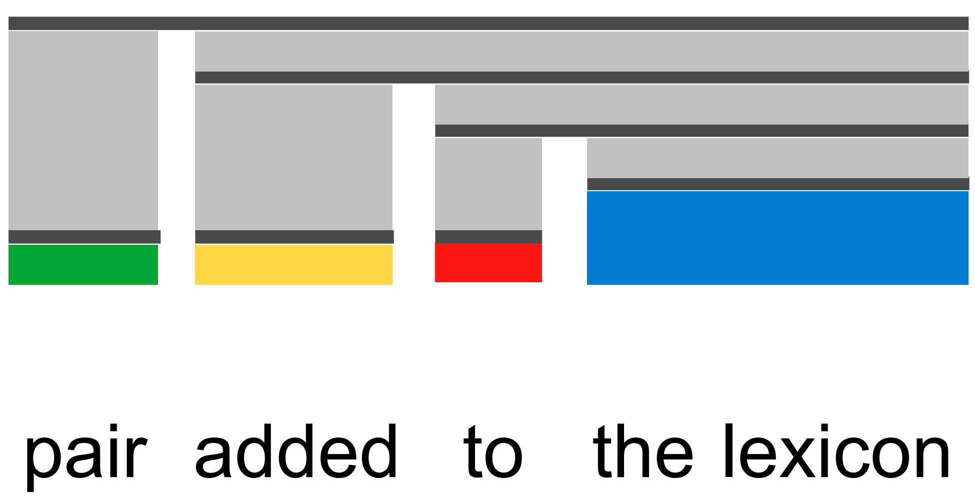
- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)

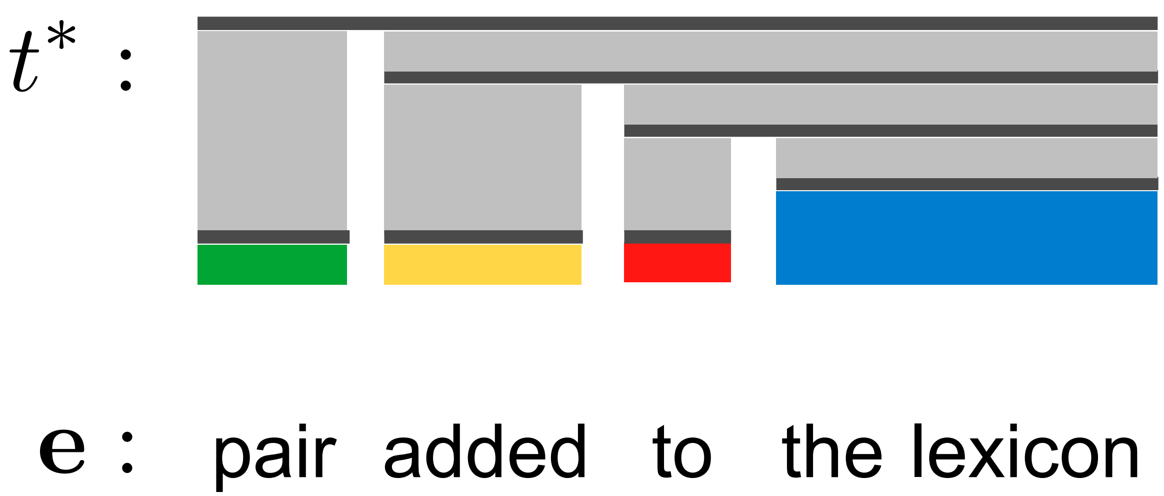
Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)



Monolingual Parsing Model

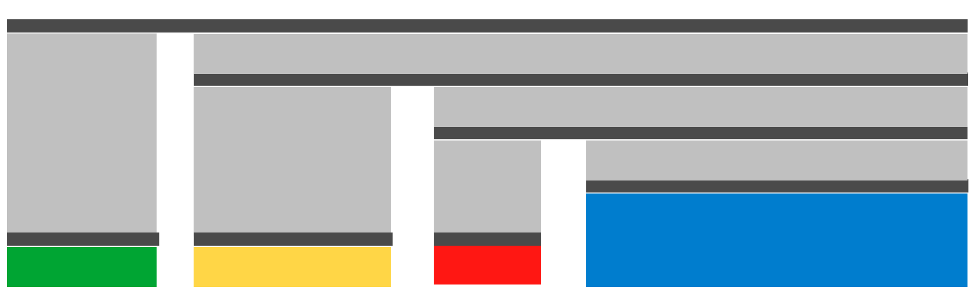
- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)



Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)

$t^* :$



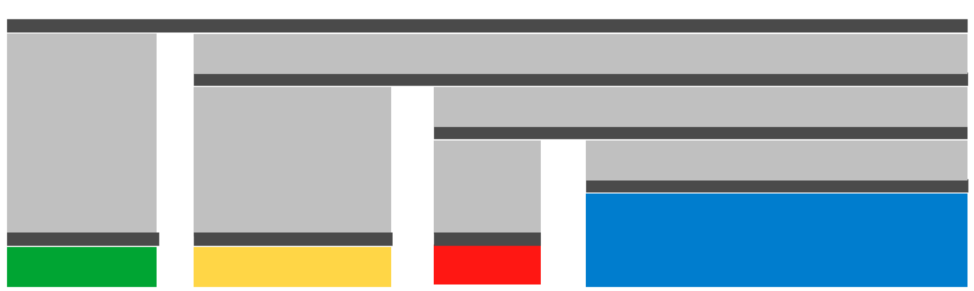
$\mathbf{e} :$ pair added to the lexicon

$$\mathrm{P}_w(t|\mathbf{e}) \propto \exp(w \cdot \theta(t, \mathbf{e}))$$

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)

t^* :



$\mathbf{e}$: pair added to the lexicon

$$\mathrm{P}_w(t|\mathbf{e}) \propto \exp(w \cdot \theta(t, \mathbf{e}))$$

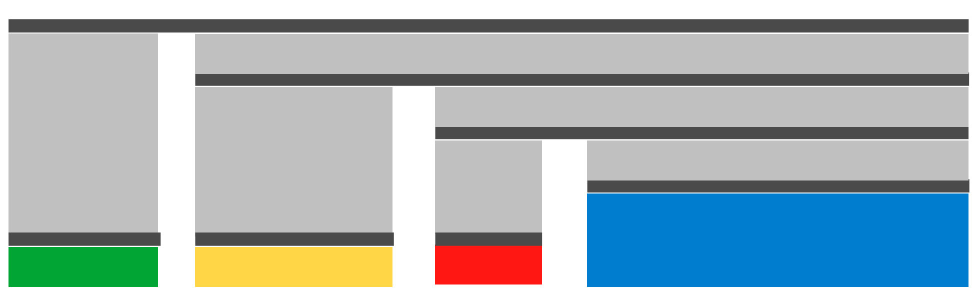
$$\arg\max_w \left[\prod_{(t^*, \mathbf{e})} \mathrm{P}_w(t^*|\mathbf{e}) \right]$$

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)

$$\prod_{e} \phi(e)$$

$t^* :$



$\mathbf{e} :$ pair added to the lexicon

$$\mathrm{P}_w(t|\mathbf{e}) \propto \exp(w \cdot \theta(t, \mathbf{e}))$$

$$\arg\max_{w} \left[\prod_{(t^*, \mathbf{e})} \mathrm{P}_w(t^*|\mathbf{e}) \right]$$

Monolingual Parsing Model

- Predict the same tree as the parallel parser, *without* the alignments
- Lexical, word class, corpus statistics, & length features
- Conditional likelihood objective
- Maximum terminal phrase length (2 for English-Japanese)

$\prod_e \phi(e)$, constituents, contexts, span length, common words, ...

t^* :

$\mathbf{e}$: pair added to the lexicon

$$P_w(t|\mathbf{e}) \propto \exp(w \cdot \theta(t, \mathbf{e}))$$

$$\arg\max_w \left[\prod_{(t^*, \mathbf{e})} P_w(t^*|\mathbf{e}) \right]$$

Reordering Models for Terminals & Non-Terminals

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals

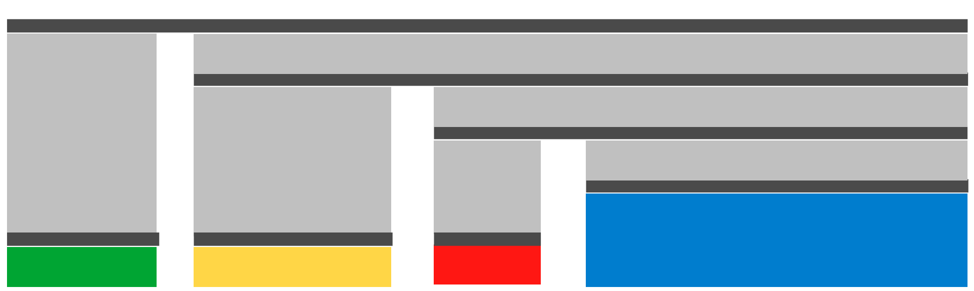
Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective

t^* :



$\mathbf{e}$: pair added to the lexicon

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective

Non-terminal model
trained on tree spans

$t^* :$

$\mathbf{e} :$ pair added to the lexicon

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective

Non-terminal model
trained on tree spans

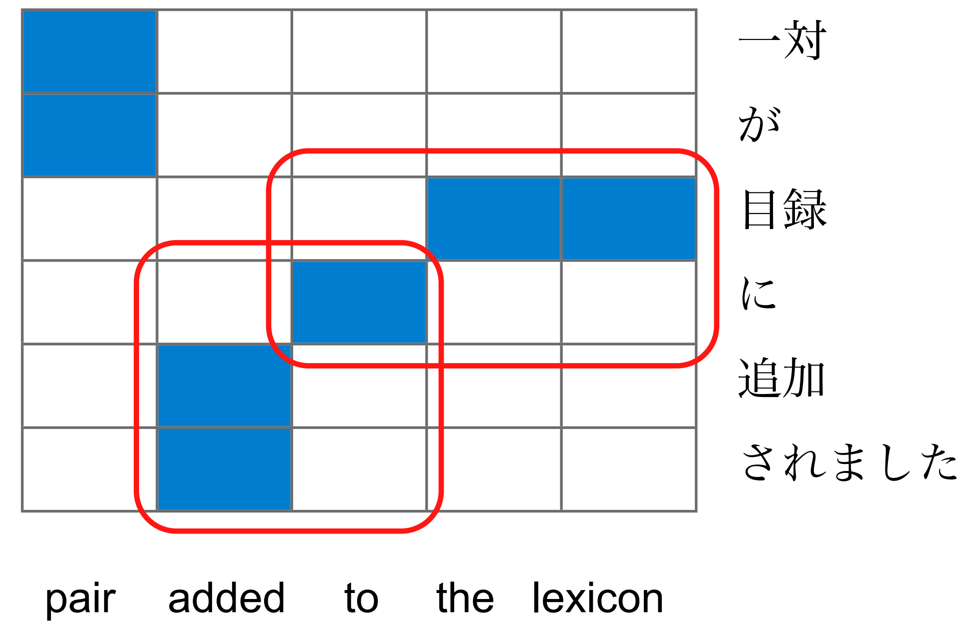
t^* :

$\mathbf{e}$: pair added to the lexicon

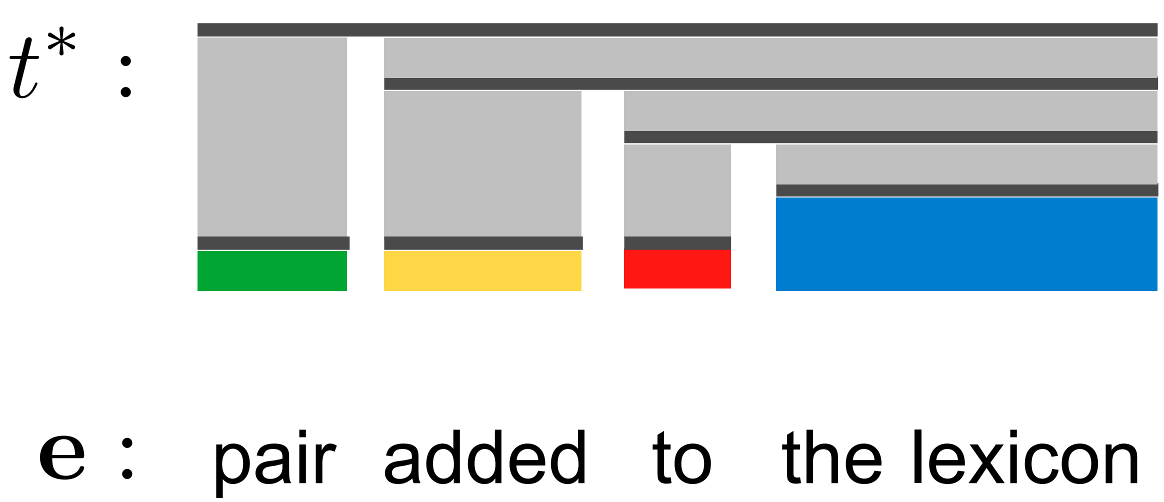
Terminal model trained on *all* contiguously aligned spans

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective



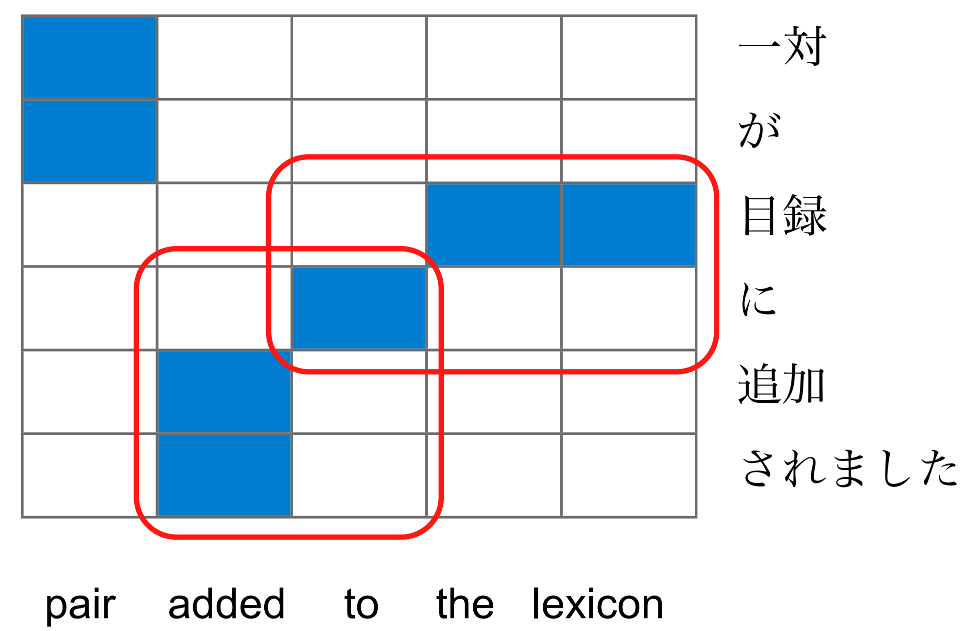
Non-terminal model trained on tree spans



Terminal model trained on *all* contiguously aligned spans

Reordering Models for Terminals & Non-Terminals

- Predict which spans to permute
- Features similar to monolingual parser
- Terminals vs non-terminals
- Maximum Entropy objective



Non-terminal model trained on tree spans

t^* :

$\mathbf{e}$: pair added to the lexicon

Terminal model trained on *all* contiguously aligned spans

Related Work (Sub-Sampled from Paper)

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)
- Synchronous grammar induction

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)
- Synchronous grammar induction
  - ITG models focus on lexical productions (Blunsom et al., ACL 09)

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)
- Synchronous grammar induction
  - ITG models focus on lexical productions (Blunsom et al., ACL 09)
- Monolingual grammar induction

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)
- Synchronous grammar induction
  - ITG models focus on lexical productions (Blunsom et al., ACL 09)
- Monolingual grammar induction
  - Likelihood objective over a monolingual corpus

Related Work (Sub-Sampled from Paper)

- Syntactic Pre-ordering
- Pre-ordering as permutation prediction
  - Learn the ordering for each pair of words (Tromble & Eisner, EMNLP 09)
  - Learn post-ordering collocations (Visweswariah, EMNLP 2011)
- Synchronous grammar induction
  - ITG models focus on lexical productions (Blunsom et al., ACL 09)
- Monolingual grammar induction
  - Likelihood objective over a monolingual corpus
  - Alignments (Kuhn, ACL 04) and bitexts (Snyder et al., ACL 09) are useful

An Evaluation Method for Pre-Ordering Pipelines

An Evaluation Method for Pre-Ordering Pipelines

1 Annotate a parallel corpus with word alignments

- Translators are responsible for annotation
- Solicit translations that align well
- Details in Talbot et al. (WMT 2011)

An Evaluation Method for Pre-Ordering Pipelines

1. Annotate a parallel corpus with word alignments
   - Translators are responsible for annotation
   - Solicit translations that align well
   - Details in Talbot et al. (WMT 2011)
2. Derive “gold ordering” σ^* from word alignments
   - Sort source words by the index of their aligned targets
   - Sensible handling of “null” alignments

An Evaluation Method for Pre-Ordering Pipelines

1. Annotate a parallel corpus with word alignments
   - Translators are responsible for annotation
   - Solicit translations that align well
   - Details in Talbot et al. (WMT 2011)
2. Derive "gold ordering" σ^* from word alignments
   - Sort source words by the index of their aligned targets
   - Sensible handling of "null" alignments
3. Compute "fuzzy reordering" metric for proposed ordering $\hat{\sigma}$

$$\frac{|\mathbf{e}| - \text{chunks}(\hat{\sigma}, \sigma^*)}{|\mathbf{e}| - 1}$$

An Evaluation Method for Pre-Ordering Pipelines

1. Annotate a parallel corpus with word alignments
   - Translators are responsible for annotation
   - Solicit translations that align well
   - Details in Talbot et al. (WMT 2011)
2. Derive "gold ordering" σ^* from word alignments
   - Sort source words by the index of their aligned targets
   - Sensible handling of "null" alignments
3. Compute "fuzzy reordering" metric for proposed ordering $\hat{\sigma}$

$$\frac{|\mathbf{e}| - \text{chunks}(\hat{\sigma}, \sigma^*)}{|\mathbf{e}| - 1}$$

Partition $\hat{\sigma}$ into pieces, s.t.

An Evaluation Method for Pre-Ordering Pipelines

1. Annotate a parallel corpus with word alignments
   - Translators are responsible for annotation
   - Solicit translations that align well
   - Details in Talbot et al. (WMT 2011)
2. Derive "gold ordering" σ^* from word alignments
   - Sort source words by the index of their aligned targets
   - Sensible handling of "null" alignments
3. Compute "fuzzy reordering" metric for proposed ordering $\hat{\sigma}$

$$\frac{|\mathbf{e}| - \text{chunks}(\hat{\sigma}, \sigma^*)}{|\mathbf{e}| - 1}$$

Partition $\hat{\sigma}$ into pieces, s.t.

All pieces contiguous in σ^*

An Evaluation Method for Pre-Ordering Pipelines

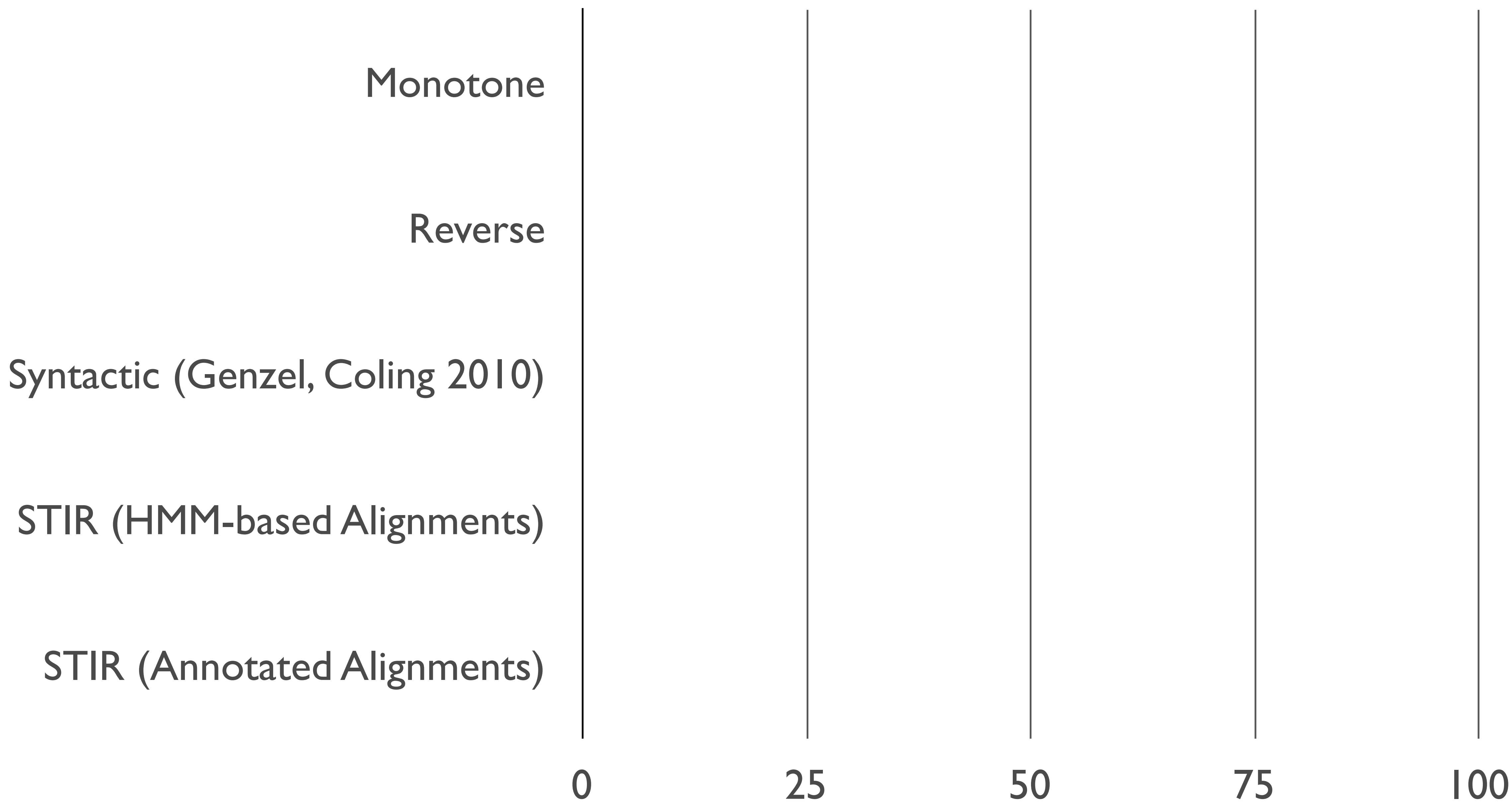
1. Annotate a parallel corpus with word alignments
   - Translators are responsible for annotation
   - Solicit translations that align well
   - Details in Talbot et al. (WMT 2011)
2. Derive "gold ordering" σ^* from word alignments
   - Sort source words by the index of their aligned targets
   - Sensible handling of "null" alignments
3. Compute "fuzzy reordering" metric for proposed ordering $\hat{\sigma}$

$$\frac{|\mathbf{e}| - \text{chunks}(\hat{\sigma}, \sigma^*)}{|\mathbf{e}| - 1}$$

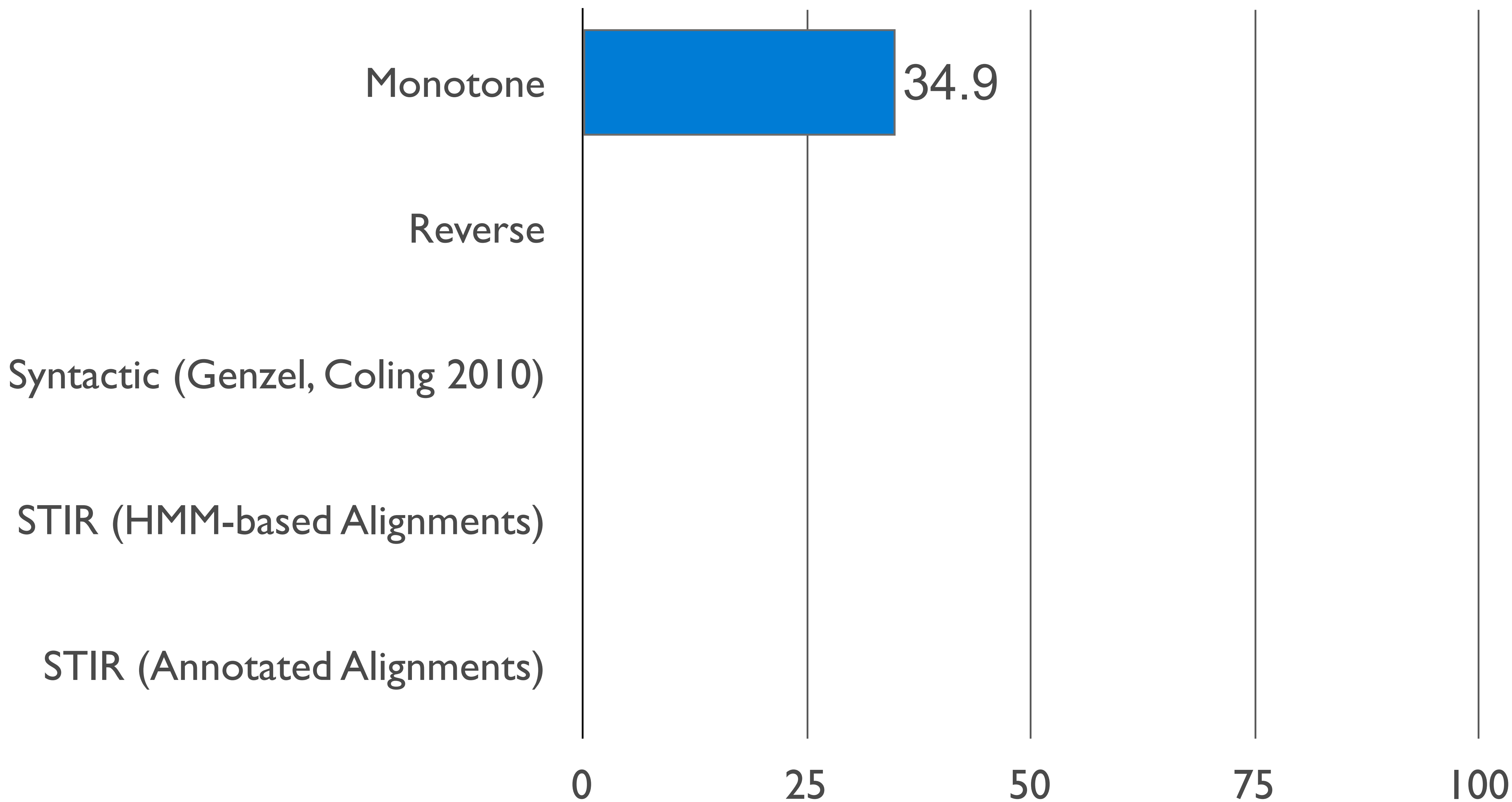
min size partition [Partition $\hat{\sigma}$ into pieces, s.t. All pieces contiguous in σ^*]

Results: Fuzzy Reordering

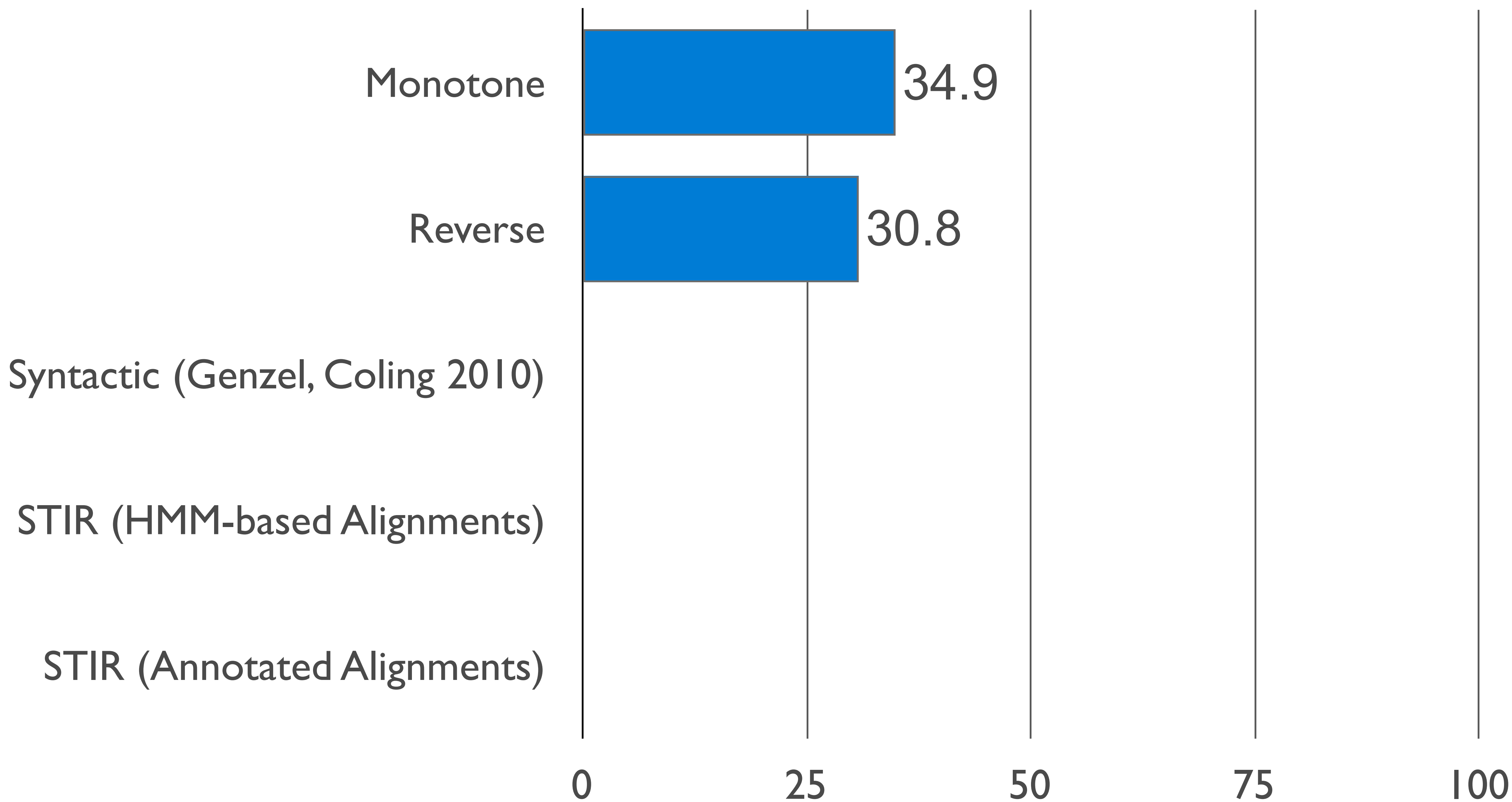
Results: Fuzzy Reordering



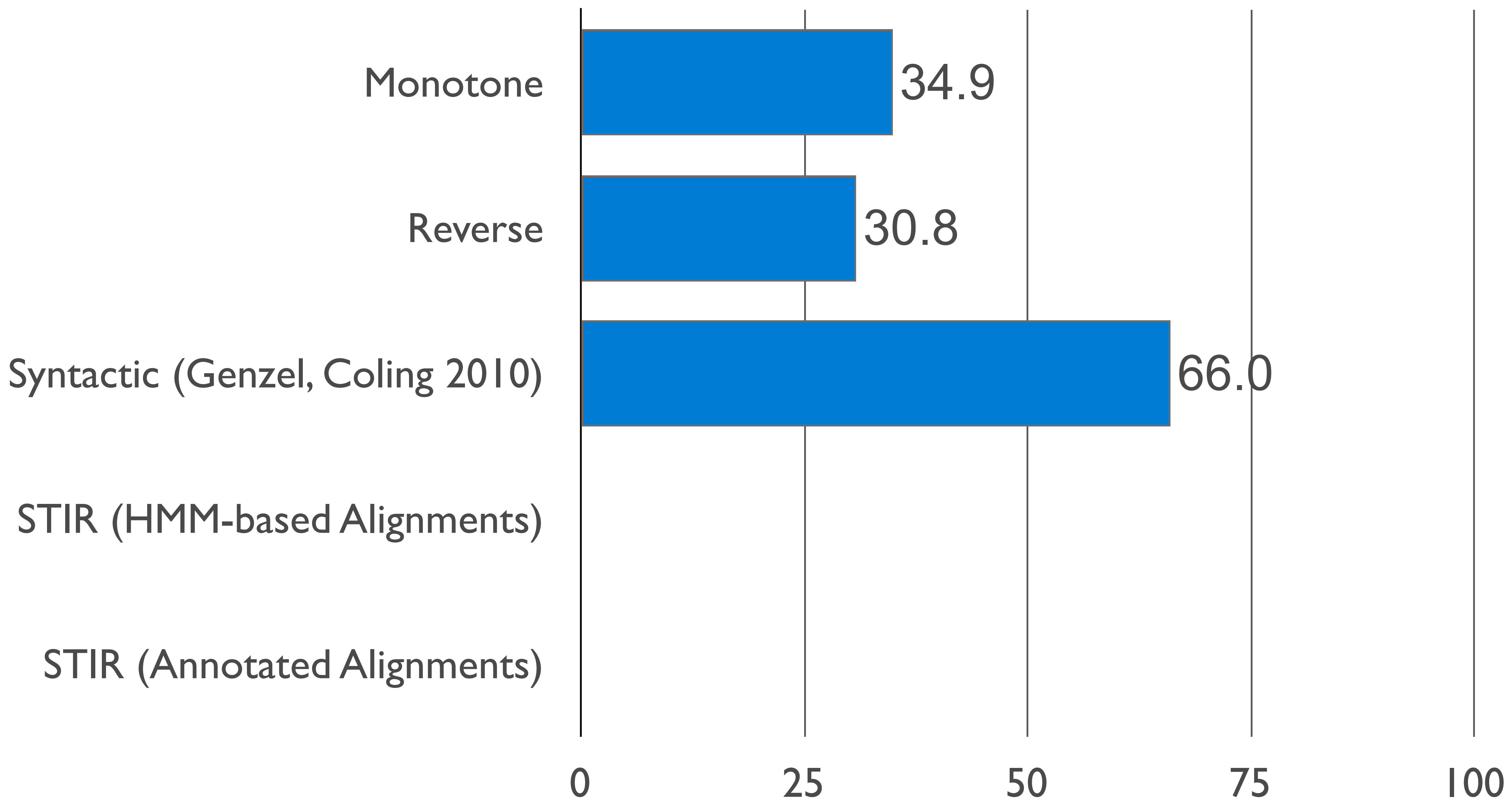
Results: Fuzzy Reordering



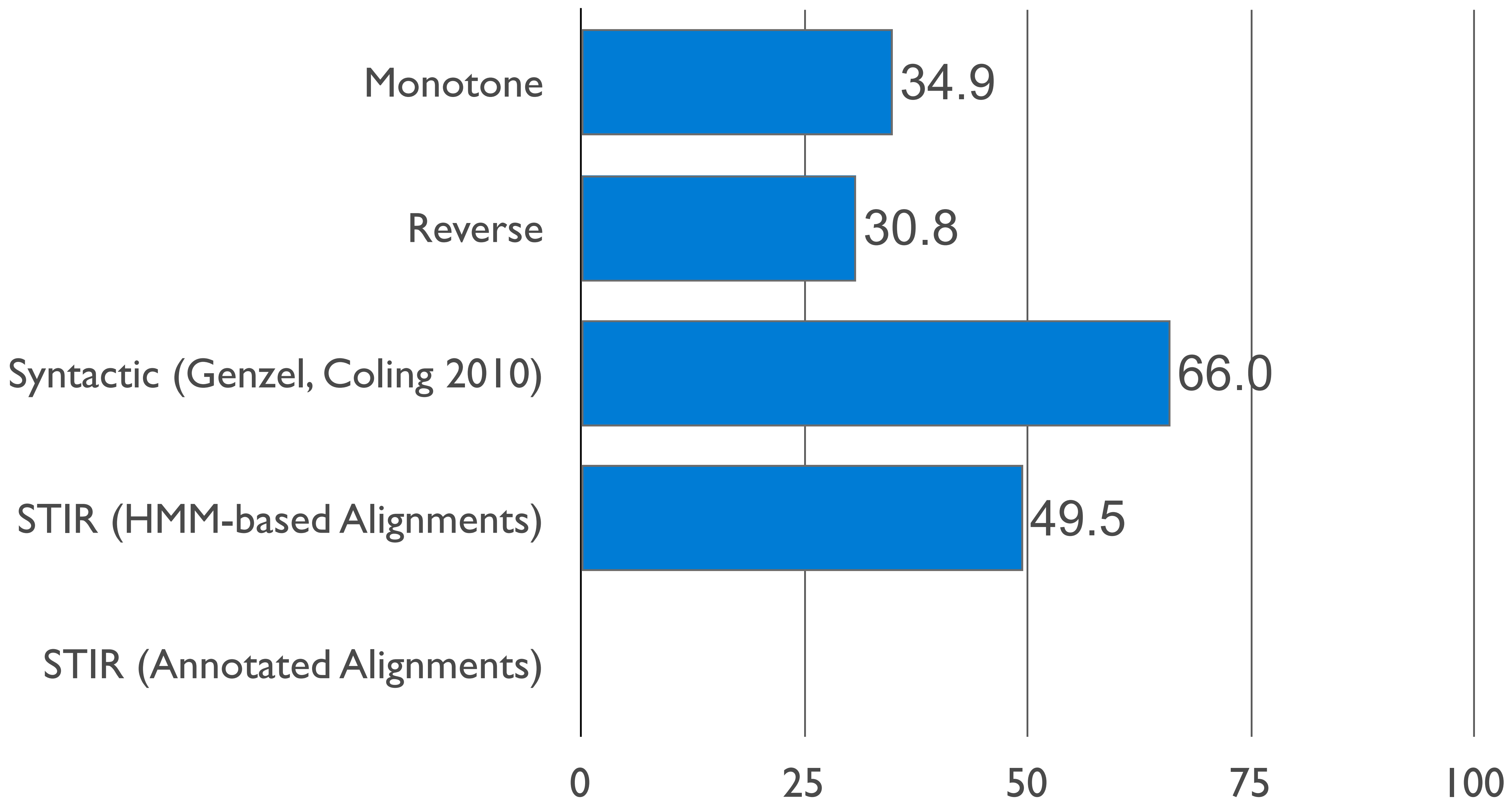
Results: Fuzzy Reordering



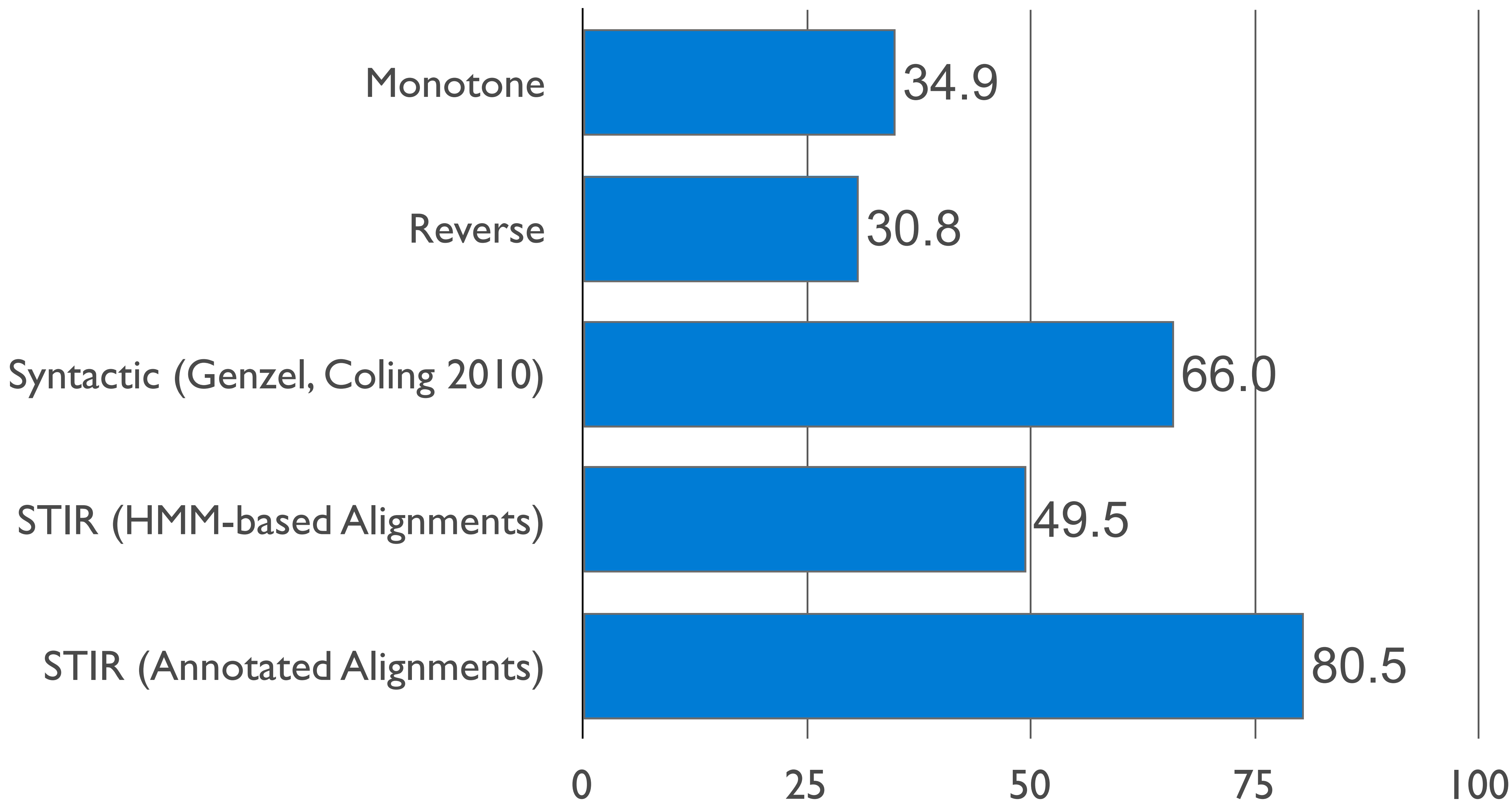
Results: Fuzzy Reordering



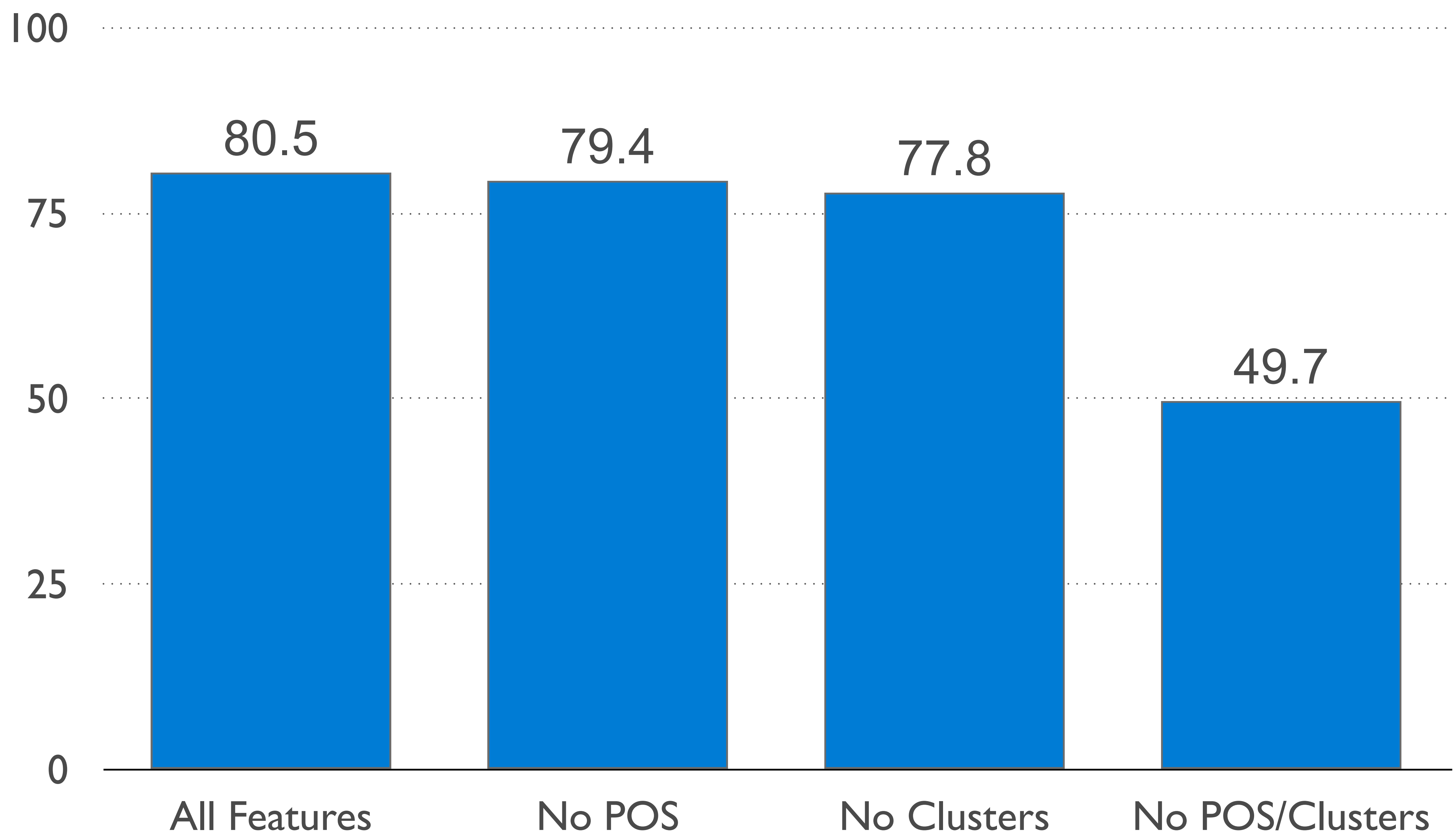
Results: Fuzzy Reordering



Results: Fuzzy Reordering



Results: Supervised & Induced Parts-of-Speech

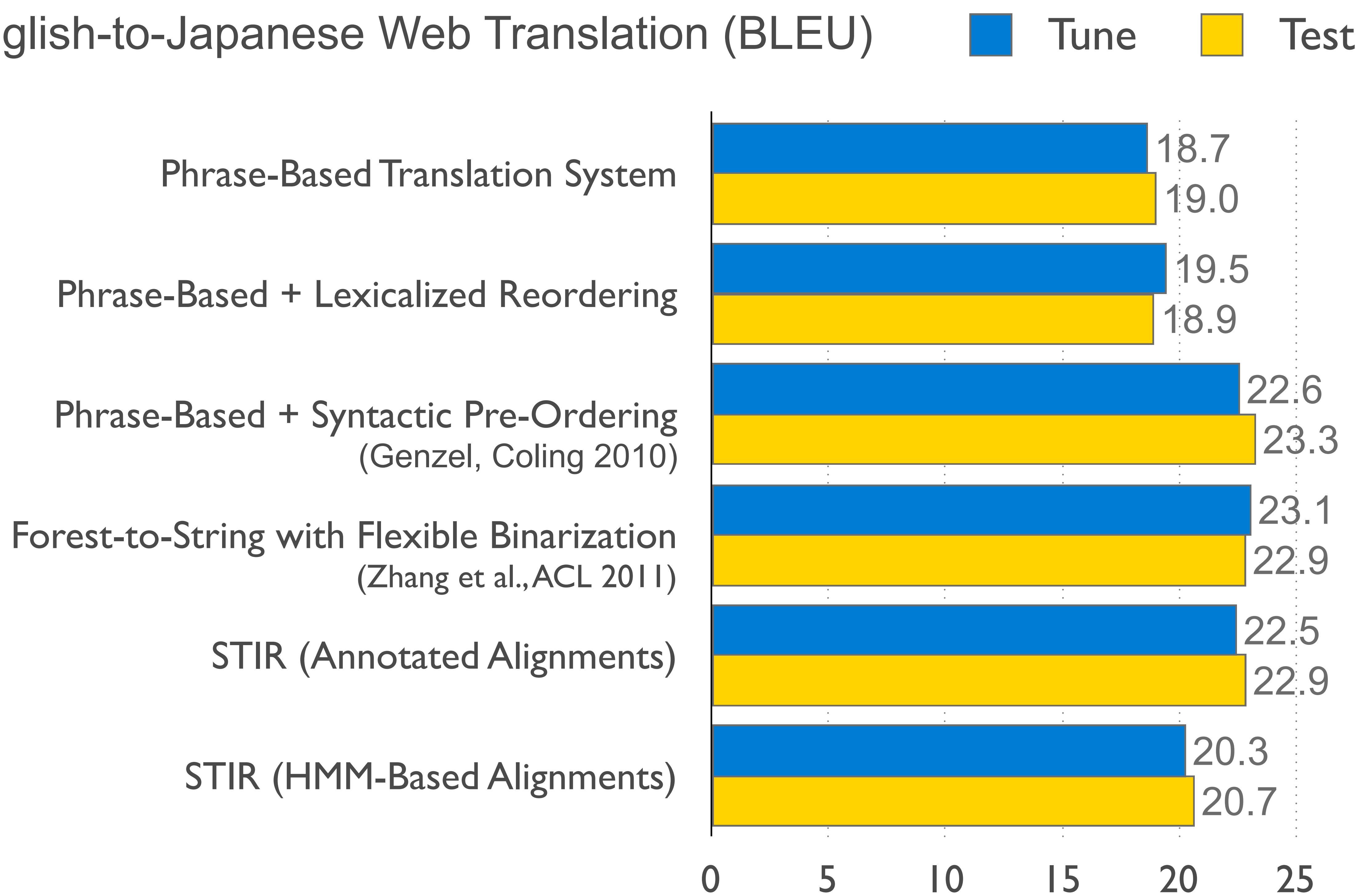


End-to-End Translation Experiments

- Translation model trained on ~700 million tokens of parallel text
- Primarily extracted from the web (Uszkoreit et al., Coling 2010)
- Alignments: 2 iterations IBM Model 1; 2 iterations HMM-based model
- Tune and test: 3100 and 1000 sentences sampled from the web

Results: End-to-End Translation

English-to-Japanese Web Translation (BLEU)



Conclusion

Conclusion

- Task-specific grammar induction matches supervised parsers

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages
  - Incorporate monolingual grammar induction models

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages
  - Incorporate monolingual grammar induction models
  - Improve performance using unsupervised alignments

Conclusion

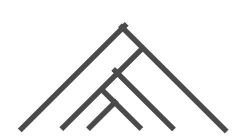
- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages
  - Incorporate monolingual grammar induction models
  - Improve performance using unsupervised alignments
  - Combine with supervised parsers

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages
  - Incorporate monolingual grammar induction models
  - Improve performance using unsupervised alignments
  - Combine with supervised parsers
- Open question: what grammatical phenomena does this learn?

Conclusion

- Task-specific grammar induction matches supervised parsers
- Reordering as an extrinsic evaluation for grammar induction
- Lots of future work:
  - Forest-based extensions
  - Induction from multiple languages
  - Incorporate monolingual grammar induction models
  - Improve performance using unsupervised alignments
  - Combine with supervised parsers
- Open question: what grammatical phenomena does this learn?



snahkT! snahkT! Thanks!