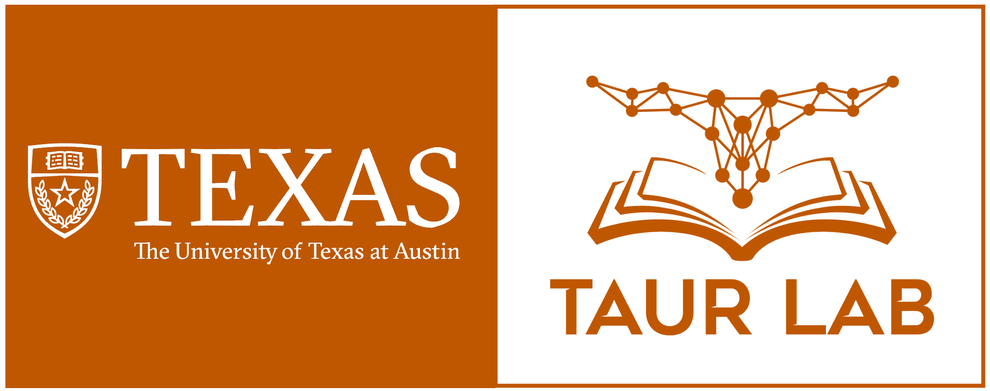
Neural Syntactic Preordering for Controlled Paraphrase Generation



Tanya Goyal and Greg Durrett

ACL 2020

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

seq2seq paraphrase model

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

seq2seq paraphrase model

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

seq2seq paraphrase model

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

seq2seq paraphrase model

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

If the Mumbai Indians win this match, they will reach the final.

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

seq2seq paraphrase model

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

If the Mumbai Indians win this match, they will reach the final.

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

→ seq2seq paraphrase model →

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

If the Mumbai Indians win this match, they will reach the final. ↗ *Clausal reordering if — they*

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

→ seq2seq paraphrase model →

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

If the Mumbai Indians win this match, they will reach the final. → *Clausal reordering if — they* → *Mumbai Indians will reach the finals if they win this match.*

Diversity in Paraphrase Generation

Input Sentence

If the Mumbai Indians win this match, they will reach the final.

→ seq2seq paraphrase model →

Generated Output

*If Mumbai Indians win, they will **proceed** to the final.*

*If Mumbai Indians win, they will **reach** the finals.*

*If Mumbai Indians **win this game**, they will reach finals.*

Small variations of the sentence!

Paraphrase models should be able to achieve **big structural reorderings!**

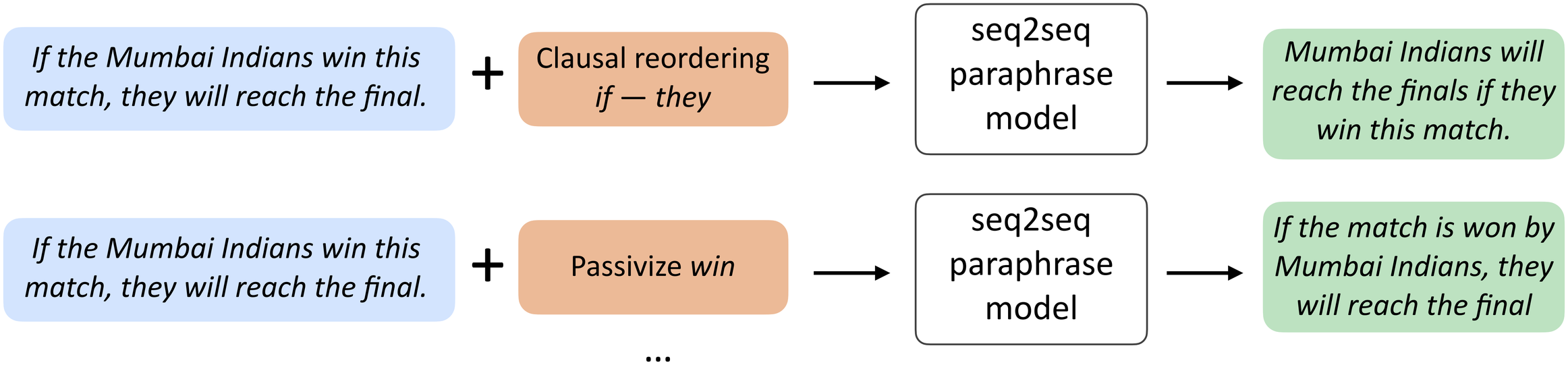
If the Mumbai Indians win this match, they will reach the final.

→ *Clausal reordering if — they* → *Mumbai Indians will reach the finals if they win this match.*

→ Passivize win → *If the match is won by Mumbai Indians, they will reach the final*

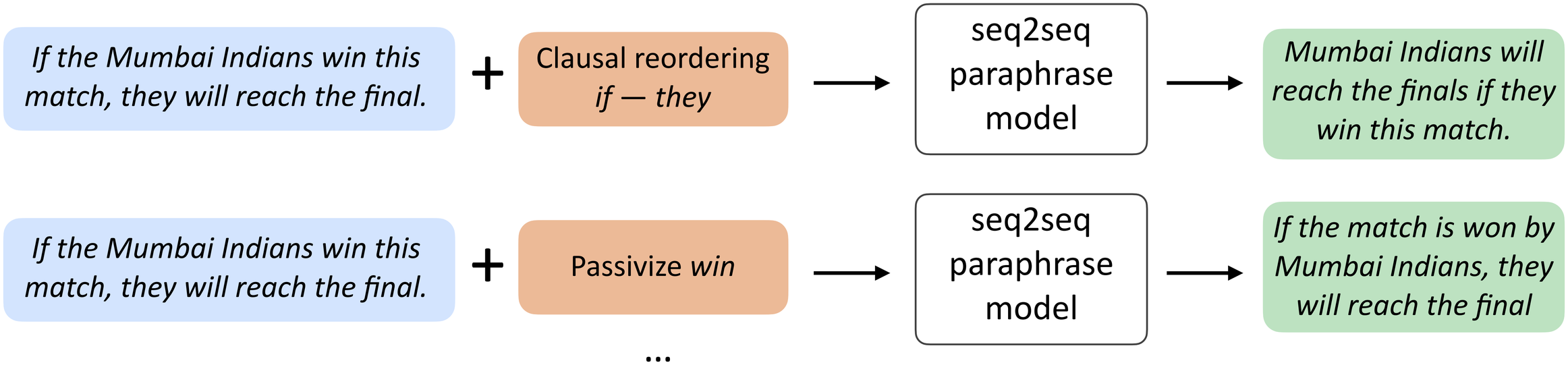
Controllable Paraphrase Generation

Can we **explicitly tell** our seq2seq model to follow these desired structures?
By controlling its behavior, we can achieve diversity!



Controllable Paraphrase Generation

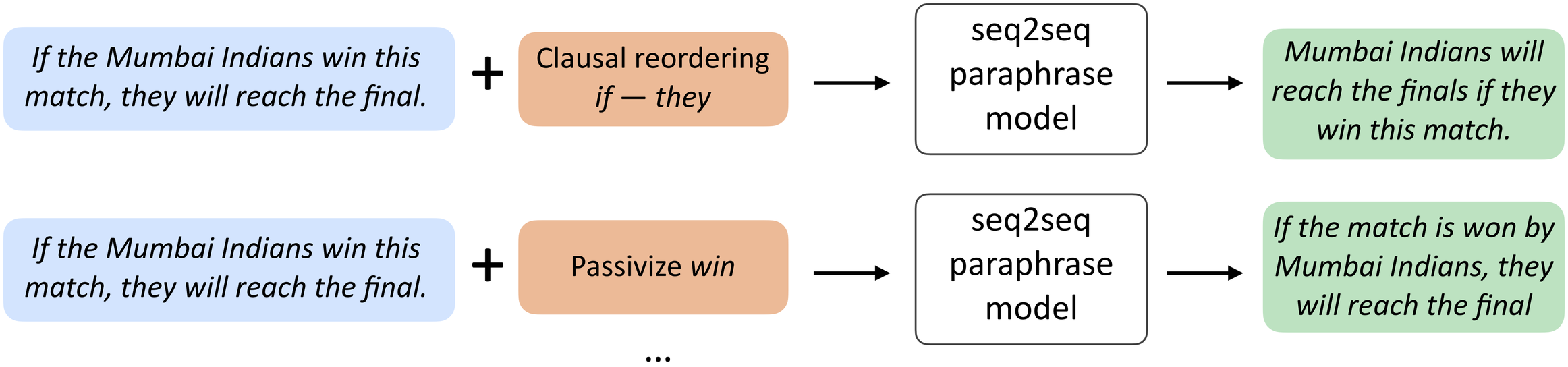
Can we **explicitly tell** our seq2seq model to follow these desired structures?
By controlling its behavior, we can achieve diversity!



How do we specify structure that the generation conditions on?

Controllable Paraphrase Generation

Can we **explicitly tell** our seq2seq model to follow these desired structures?
By controlling its behavior, we can achieve diversity!



How do we specify structure that the generation conditions on?

How do we enumerate the different paraphrasing possibilities?

Prior Work

Syntactic Template-based (Iyyer et al. 2018)

Exemplar-based (Chen et al. 2019)

Prior Work

Syntactic Template-based (Iyyer et al. 2018)

If the Mumbai Indians win this match, they will reach the final. + *S (NP VP)* → *Mumbai Indians will reach the finals if they win this match.*

Exemplar-based (Chen et al. 2019)

Prior Work

Syntactic Template-based (Iyyer et al. 2018)

If the Mumbai Indians win this match, they will reach the final. + *S (NP VP)* → *Mumbai Indians will reach the finals if they win this match.*

Limitation: Uses the same set of templates for ALL inputs, cannot identify input-appropriate templates.

Exemplar-based (Chen et al. 2019)

Prior Work

Syntactic Template-based (Iyyer et al. 2018)

If the Mumbai Indians win this match, they will reach the final. + *S (NP VP)* → *Mumbai Indians will reach the finals if they win this match.*

Limitation: Uses the same set of templates for ALL inputs, cannot identify input-appropriate templates.

Exemplar-based (Chen et al. 2019)

If the Mumbai Indians win this match, they will reach the final. + *I will eat if hungry* → *Mumbai Indians will reach the finals if they win this match.*

Prior Work

Syntactic Template-based (Iyyer et al. 2018)

If the Mumbai Indians win this match, they will reach the final. + *S (NP VP)* → *Mumbai Indians will reach the finals if they win this match.*

Limitation: Uses the same set of templates for ALL inputs, cannot identify input-appropriate templates.

Exemplar-based (Chen et al. 2019)

If the Mumbai Indians win this match, they will reach the final. + *I will eat if hungry* → *Mumbai Indians will reach the finals if they win this match.*

Limitation: Requires appropriate exemplar as input, cannot generate these automatically.

How do we specify desired target structure?

If Mumbai Indians win this match, they will reach the final.

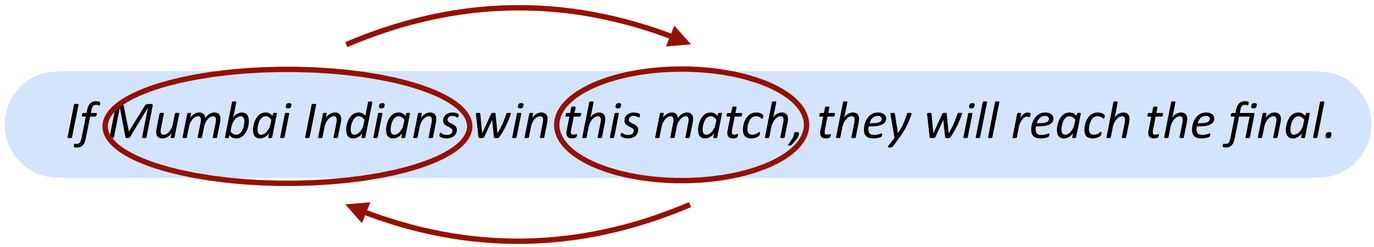
If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?

If Mumbai Indians win this match, they will reach the final.

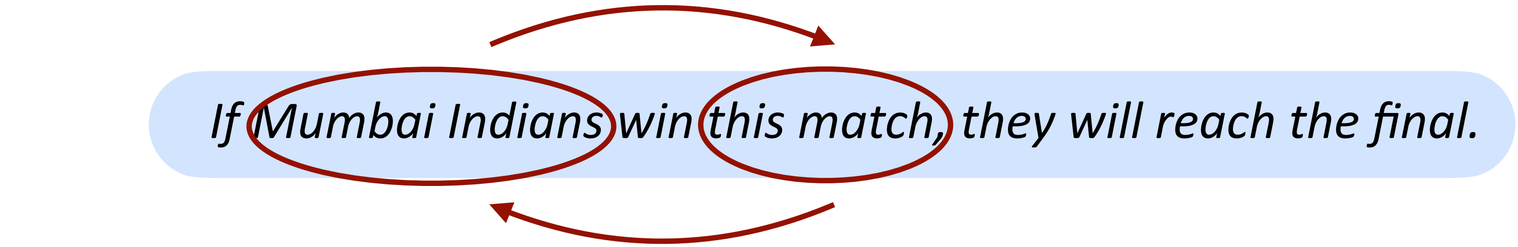
If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?



If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?

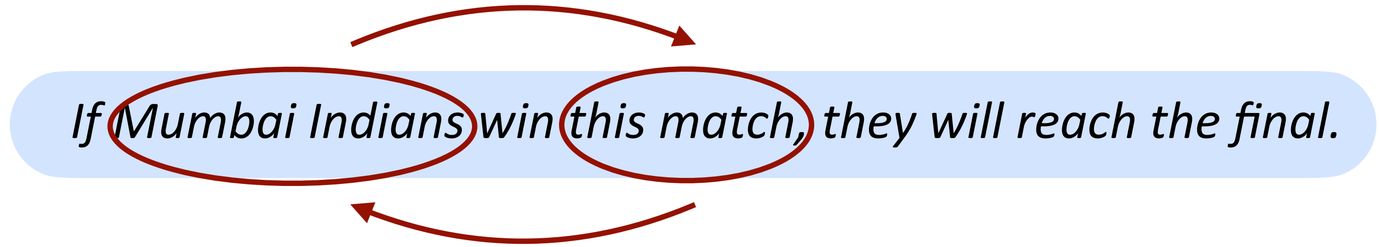


this match

Mumbai Indians

If the match is won by Mumbai Indians, they will reach the final.

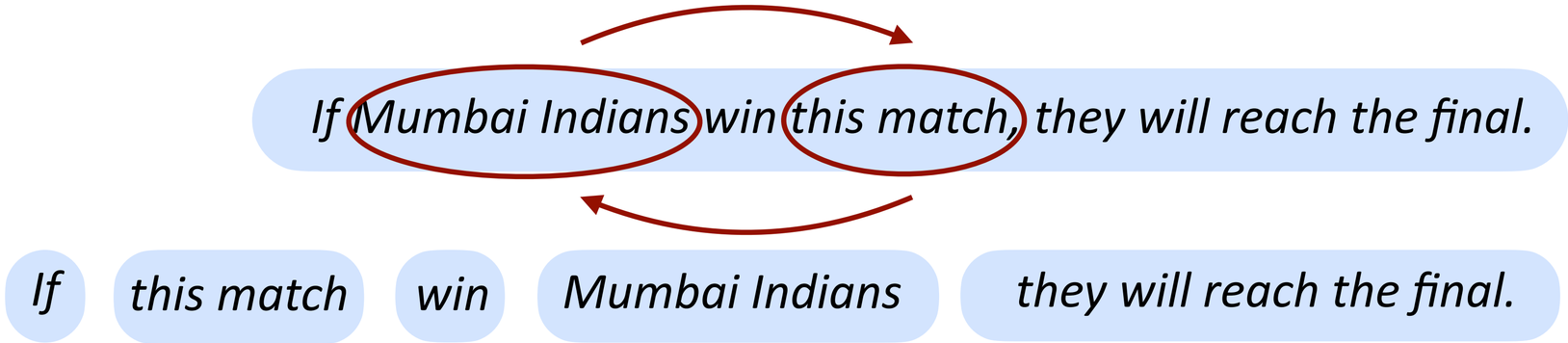
How do we specify desired target structure?



If | *this match* | *win* | *Mumbai Indians* | *they will reach the final.*

If the match is won by Mumbai Indians, they will reach the final.

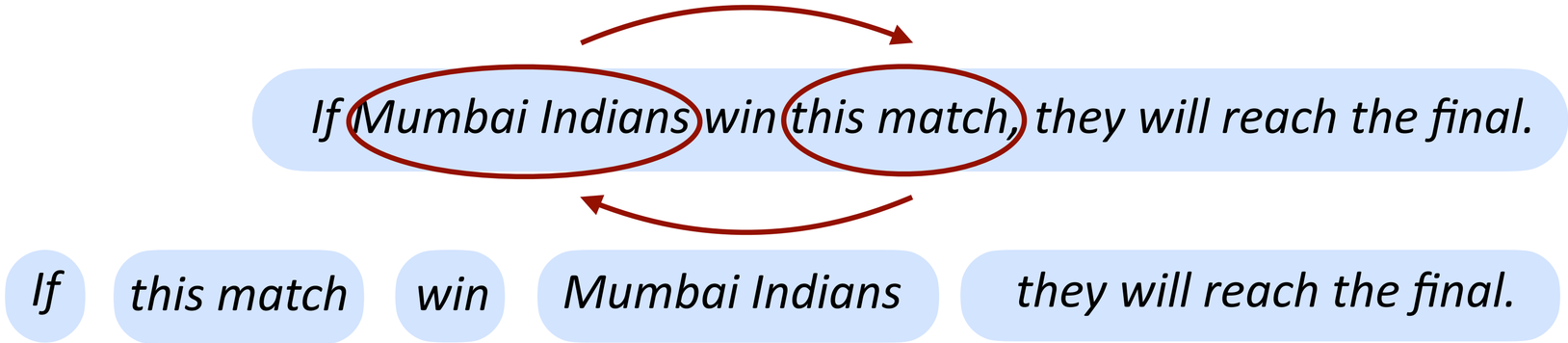
How do we specify desired target structure?



Similar to preordering in MT and denotes the right order of content

If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?

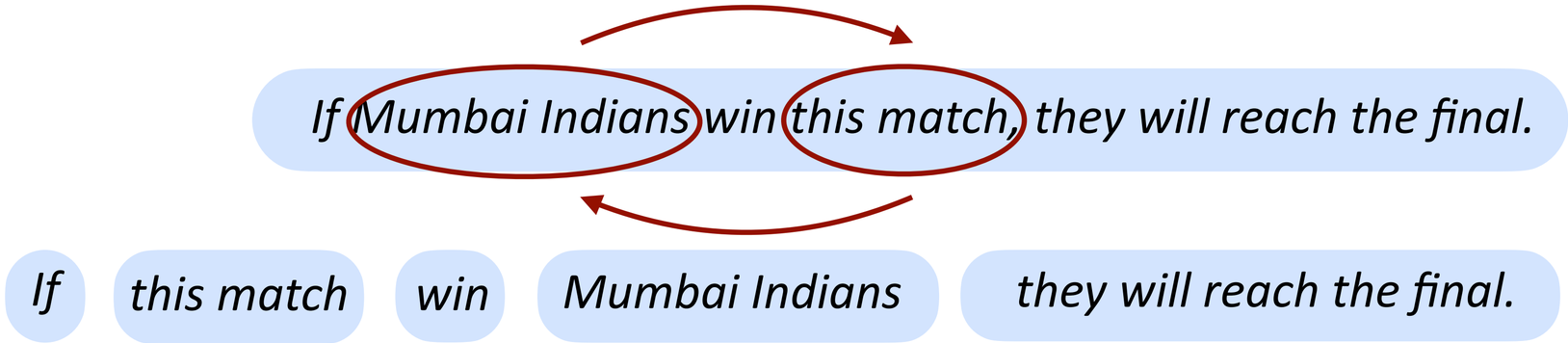


Similar to preordering in MT and denotes the right order of content

Does not retain semantics.

If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?

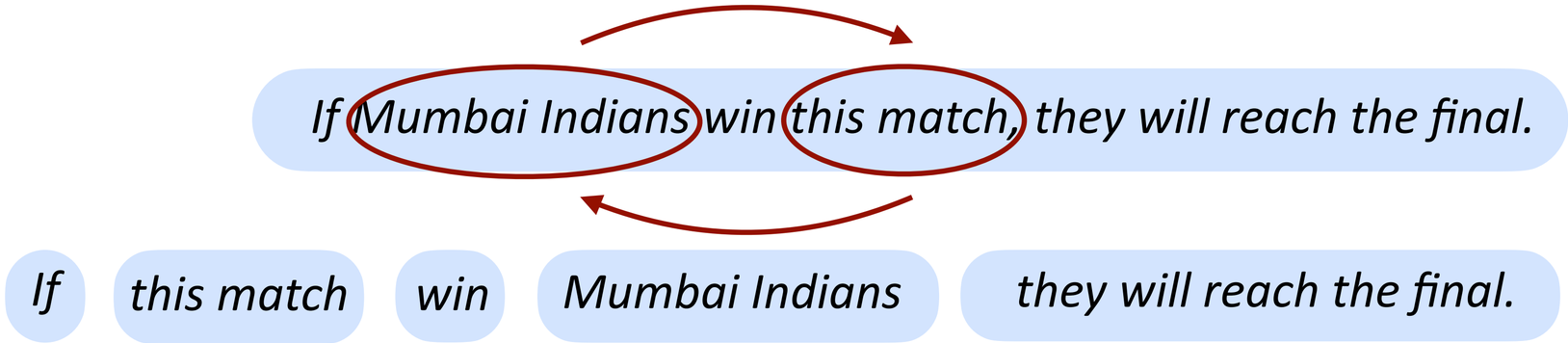


Similar to preordering in MT and denotes the right order of content

Does not retain semantics.

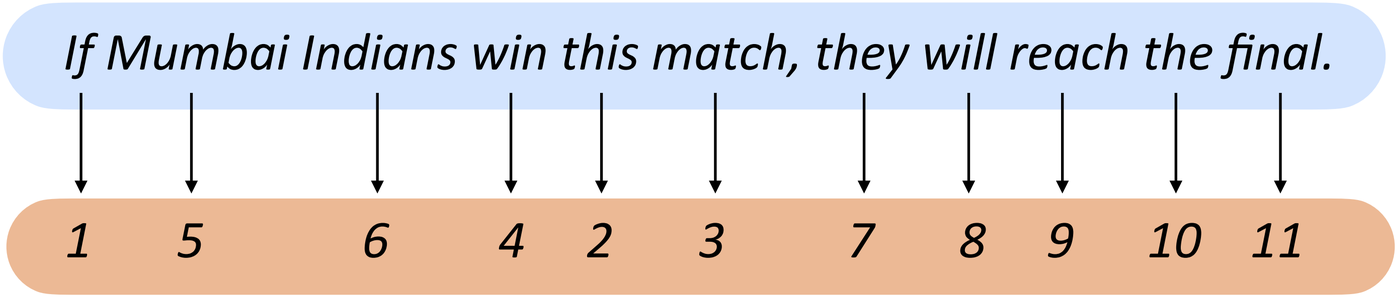
If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?



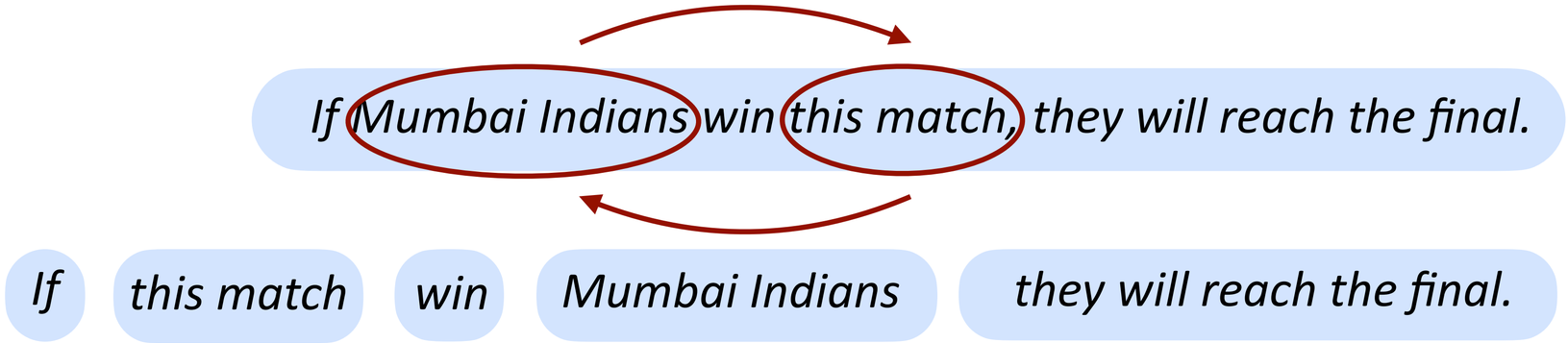
Similar to preordering in MT and denotes the right order of content

Does not retain semantics.



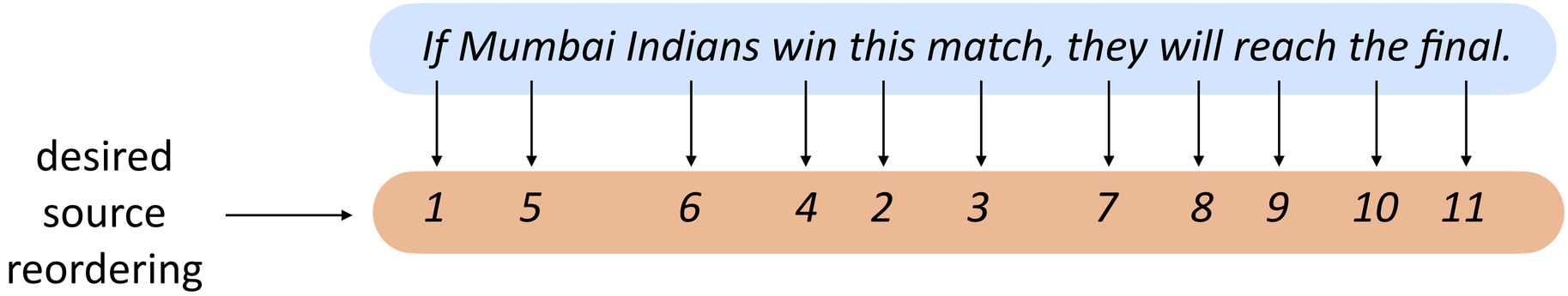
If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?



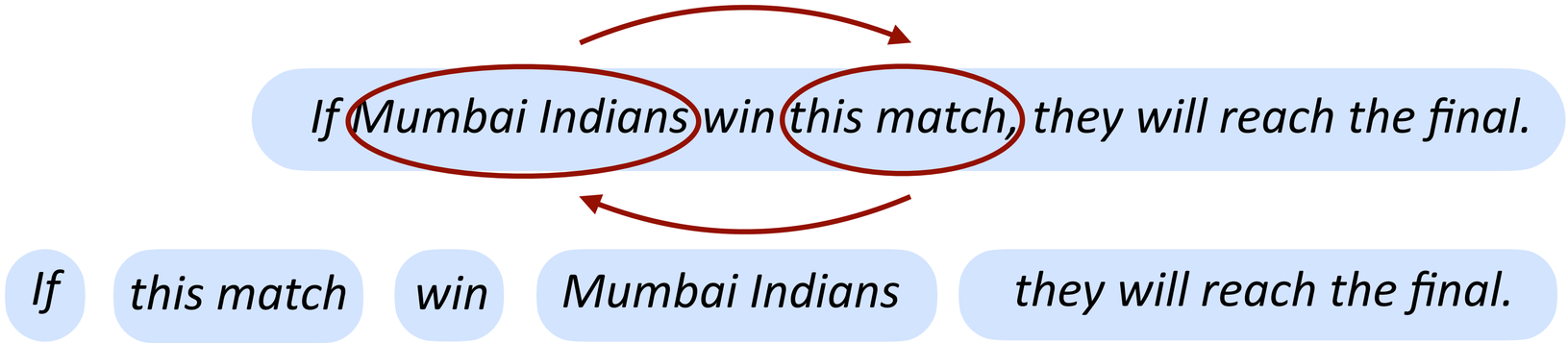
Similar to preordering in MT and denotes the right order of content

Does not retain semantics.



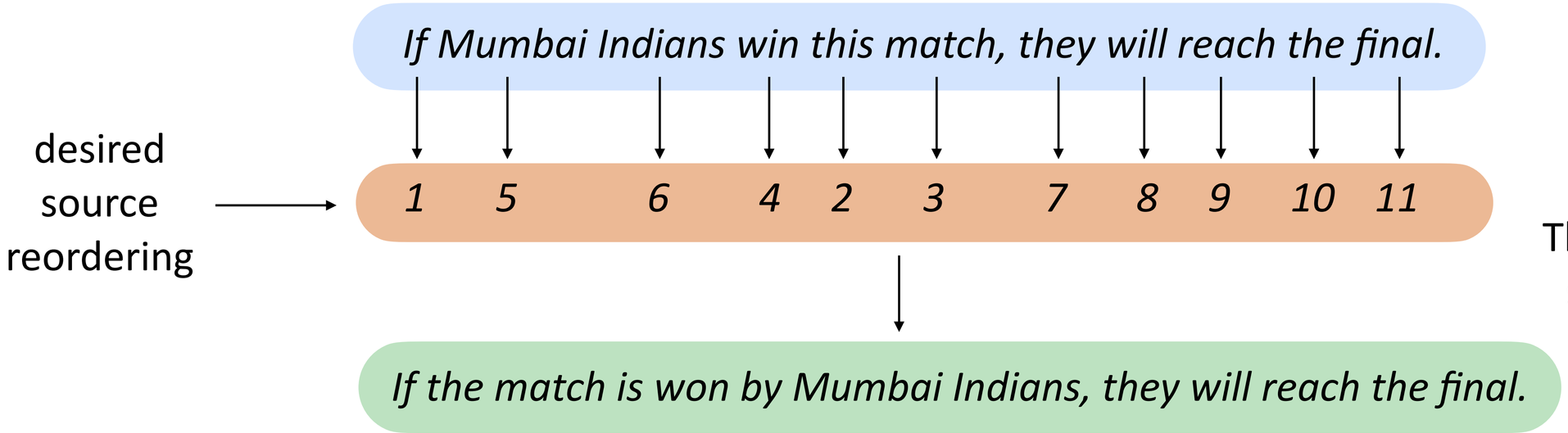
If the match is won by Mumbai Indians, they will reach the final.

How do we specify desired target structure?



Similar to preordering in MT and denotes the right order of content

Does not retain semantics.



The model needs to perform some rewriting to smooth over the transformations.

How do we generate the source reordering?

How do we generate the source reordering?

- Learn to reorder abstracted phrases instead of the whole sentence

How do we generate the source reordering?

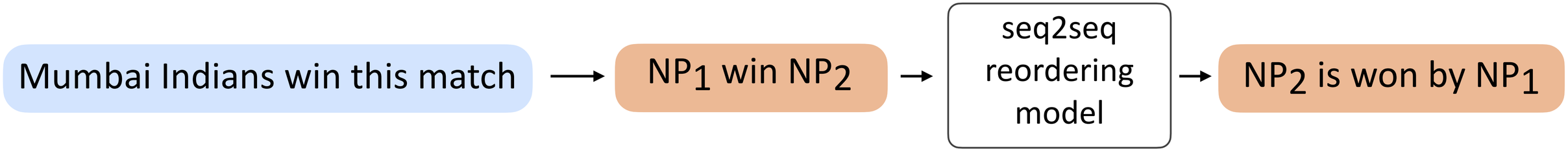
- Learn to reorder abstracted phrases instead of the whole sentence

How do we generate the source reordering?

- Learn to reorder abstracted phrases instead of the whole sentence
- By considering multiple phrase abstractions, we can enumerate many structures

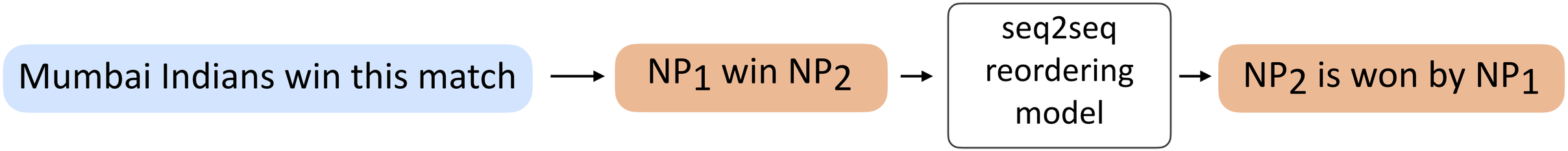
How do we generate the source reordering?

- Learn to reorder abstracted phrases instead of the whole sentence
- By considering multiple phrase abstractions, we can enumerate many structures



How do we generate the source reordering?

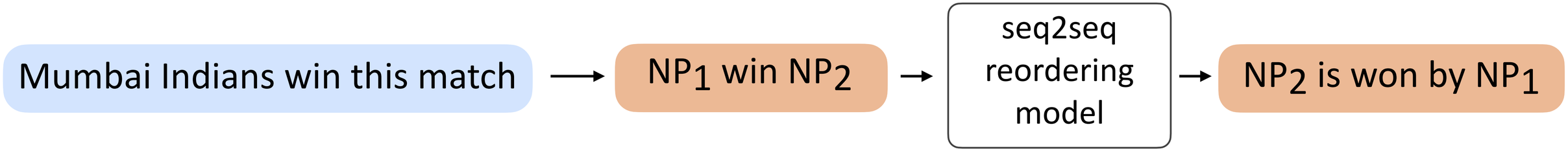
- Learn to reorder abstracted phrases instead of the whole sentence
- By considering multiple phrase abstractions, we can enumerate many structures



(captures reordering like the grammar rules in syntactic MT)

How do we generate the source reordering?

- Learn to reorder abstracted phrases instead of the whole sentence
- By considering multiple phrase abstractions, we can enumerate many structures

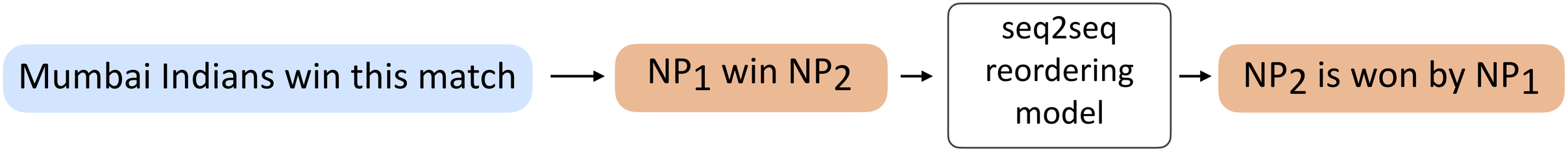


(captures reordering like the grammar rules in syntactic MT)

- Now use this to derive the input reordering:

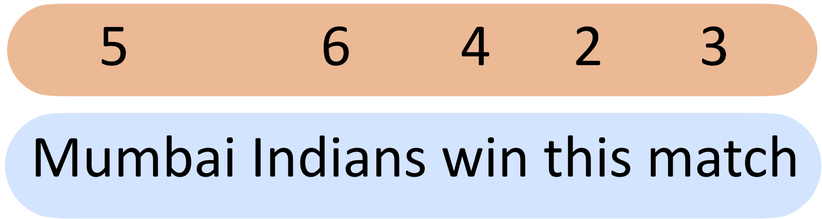
How do we generate the source reordering?

- Learn to reorder abstracted phrases instead of the whole sentence
- By considering multiple phrase abstractions, we can enumerate many structures



(captures reordering like the grammar rules in syntactic MT)

- Now use this to derive the input reordering:



Overview

Overview

Input Sequence

If Mumbai Indians win this match, they will reach the final.

Overview

Input Sequence

If Mumbai Indians win this match, they will reach the final.

Desired order of source words

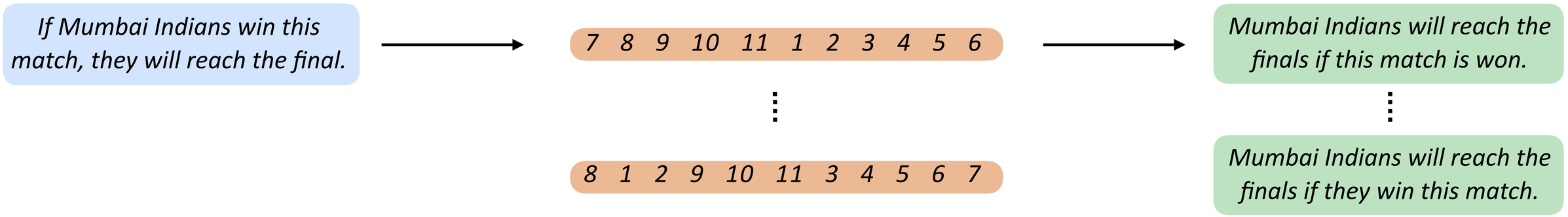
1 5 6 4 2 3 7 8 9 10 11

7 8 9 10 11 1 2 3 4 5 6

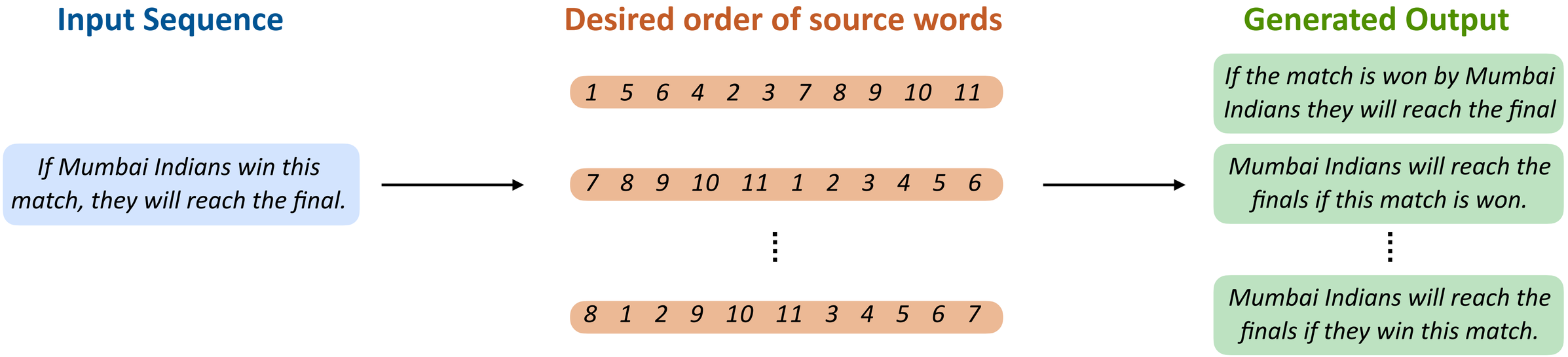
⋮

8 1 2 9 10 11 3 4 5 6 7

Overview

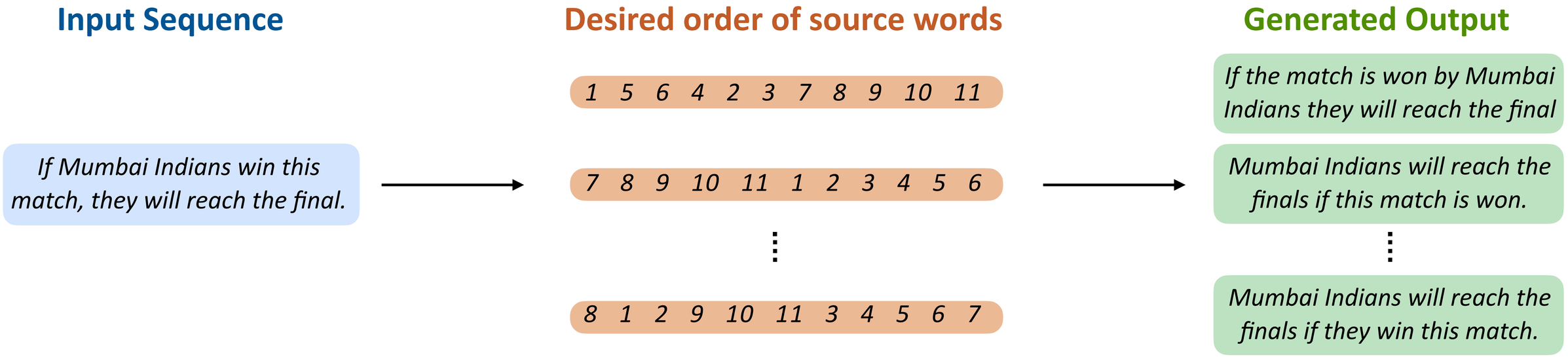

Input Sequence
Desired order of source words
Generated Output
If Mumbai Indians win this match, they will reach the final.
1 5 6 4 2 3 7 8 9 10 11
7 8 9 10 11 1 2 3 4 5 6
8 1 2 9 10 11 3 4 5 6 7
If the match is won by Mumbai Indians they will reach the final
Mumbai Indians will reach the finals if this match is won.
Mumbai Indians will reach the finals if they win this match.

Overview



SOW: Source Order reWriting

Overview

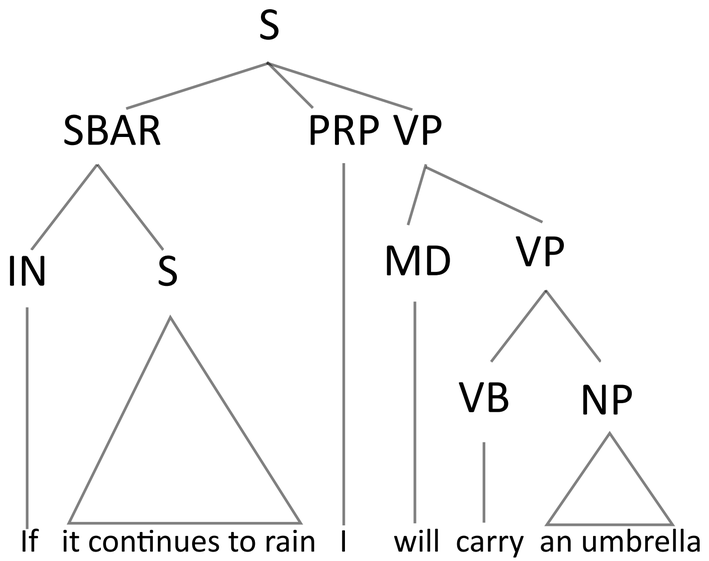


SOW: Source Order reWriting

REAP: Rearrangement Aware Paraphrasing

SOW model: Source Order Rewriting

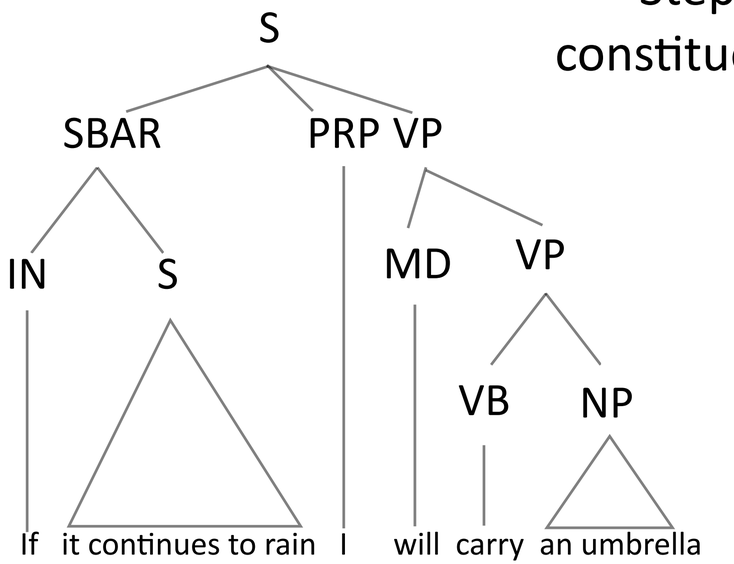
Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

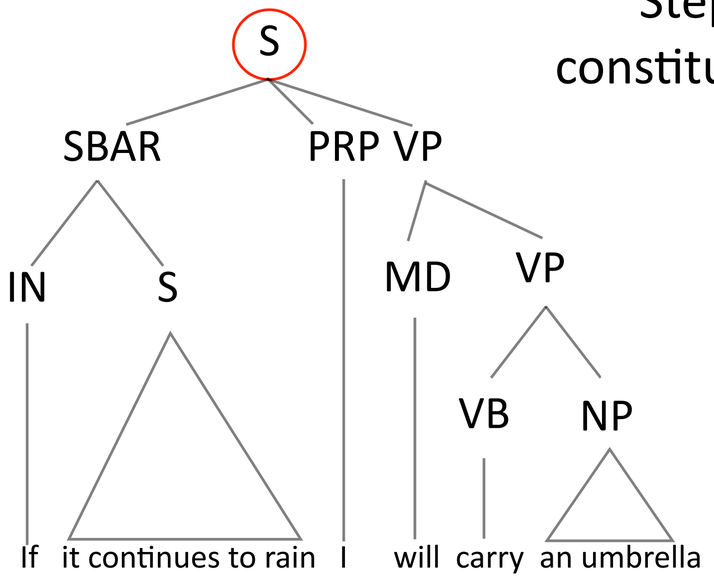
Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

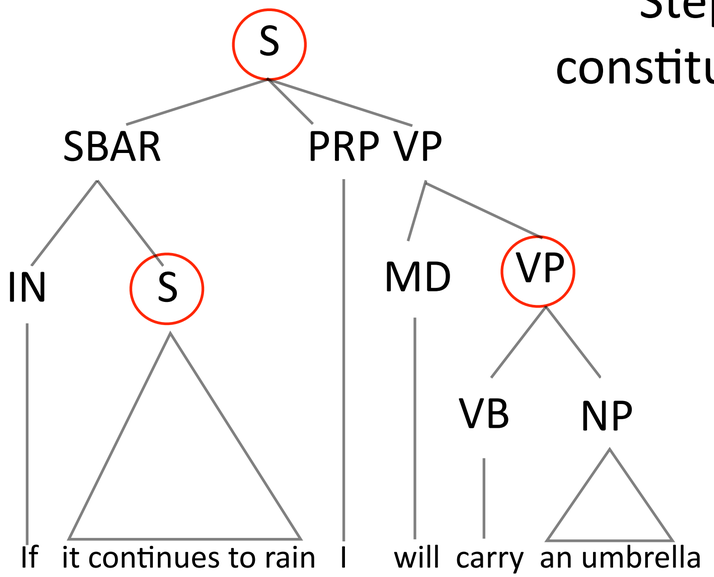
Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

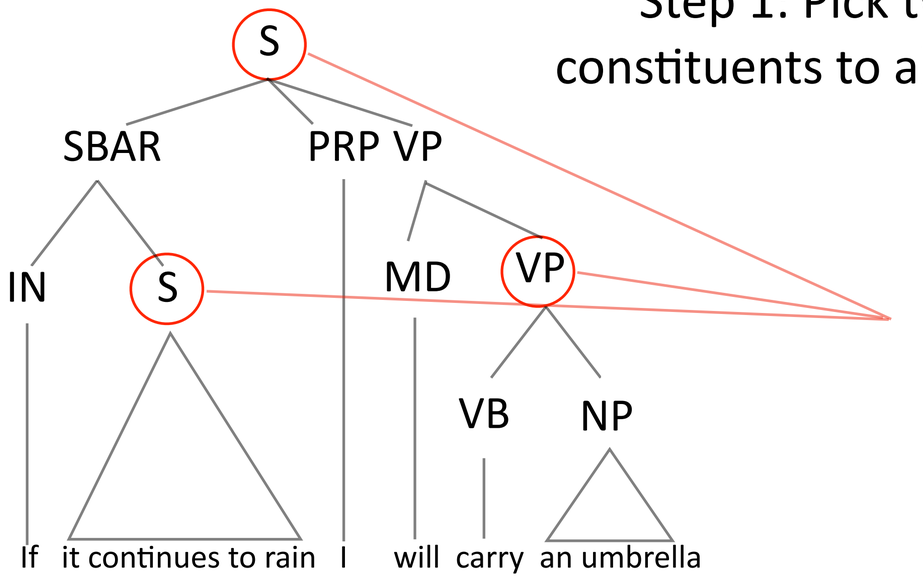
Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

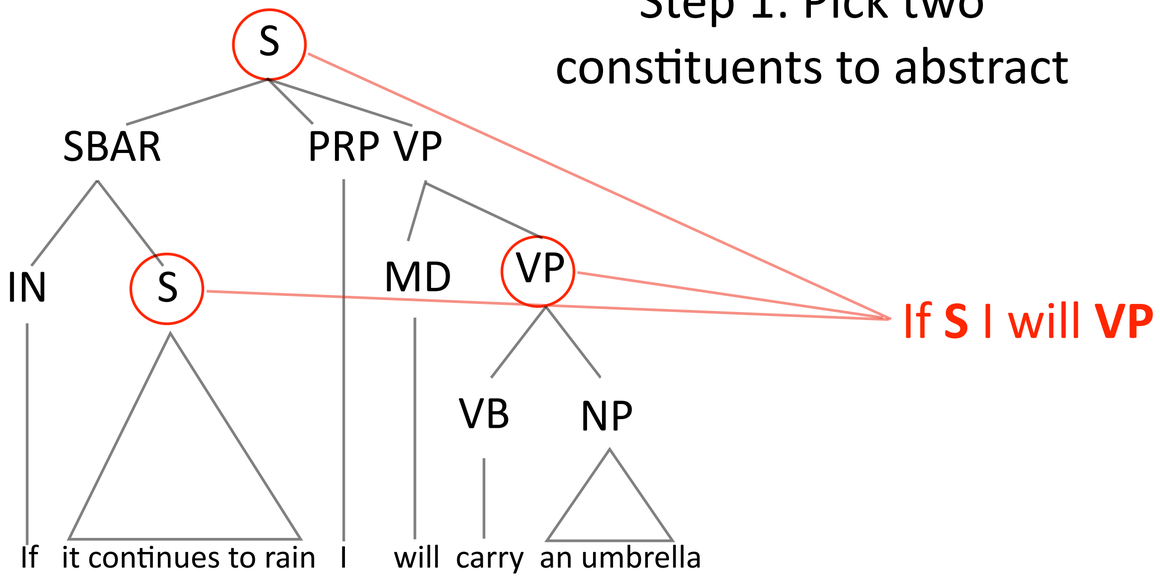
Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

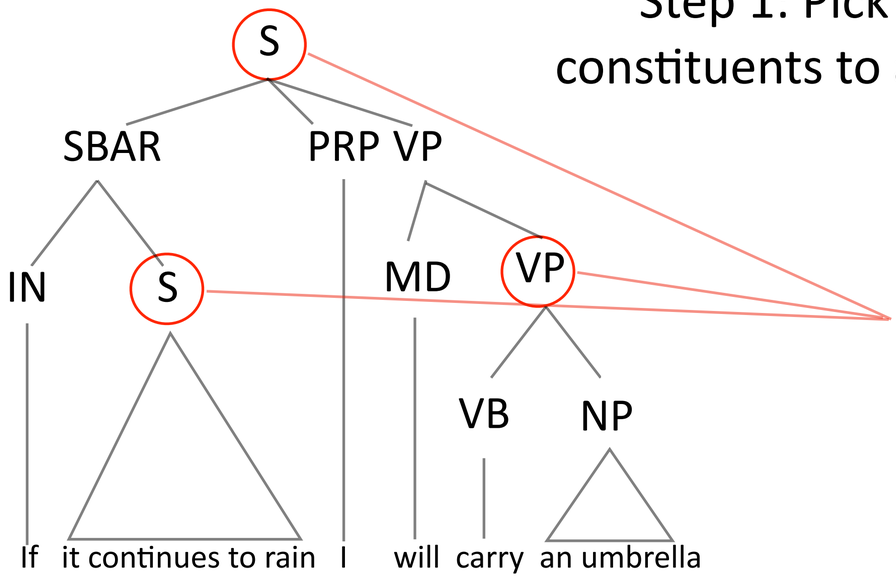
Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

Step 2: Reorder with a seq2seq model

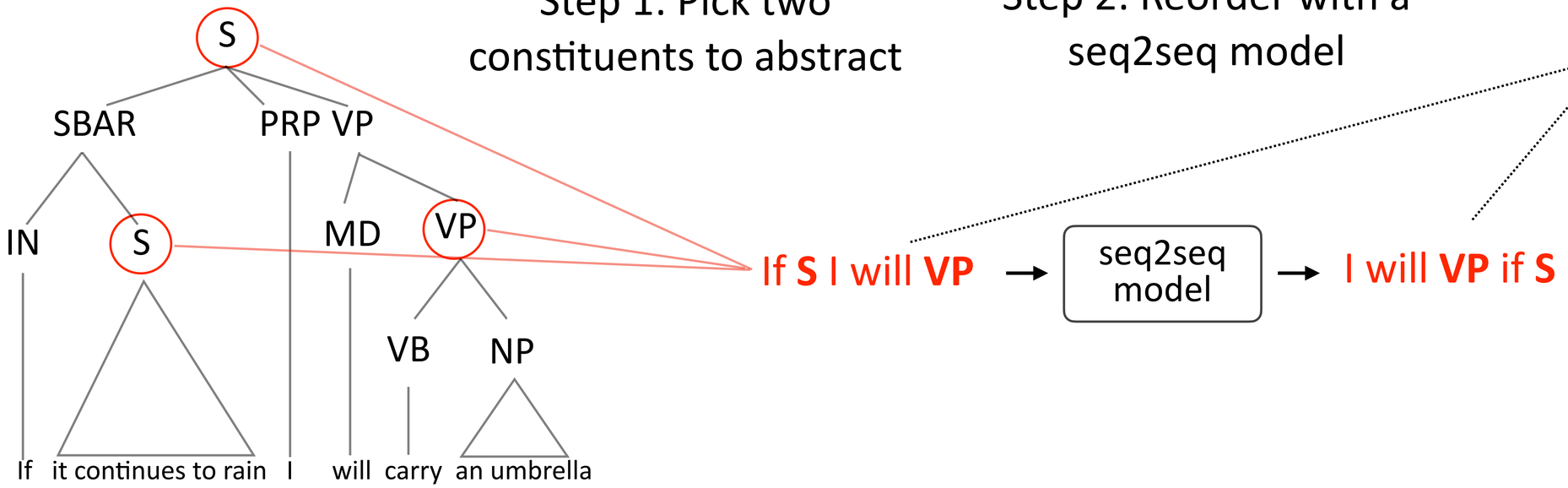
Step 3: Align and get reordering

If **S** I will **VP** → seq2seq model → I will **VP** if **S**

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



Step 1: Pick two constituents to abstract

Step 2: Reorder with a seq2seq model

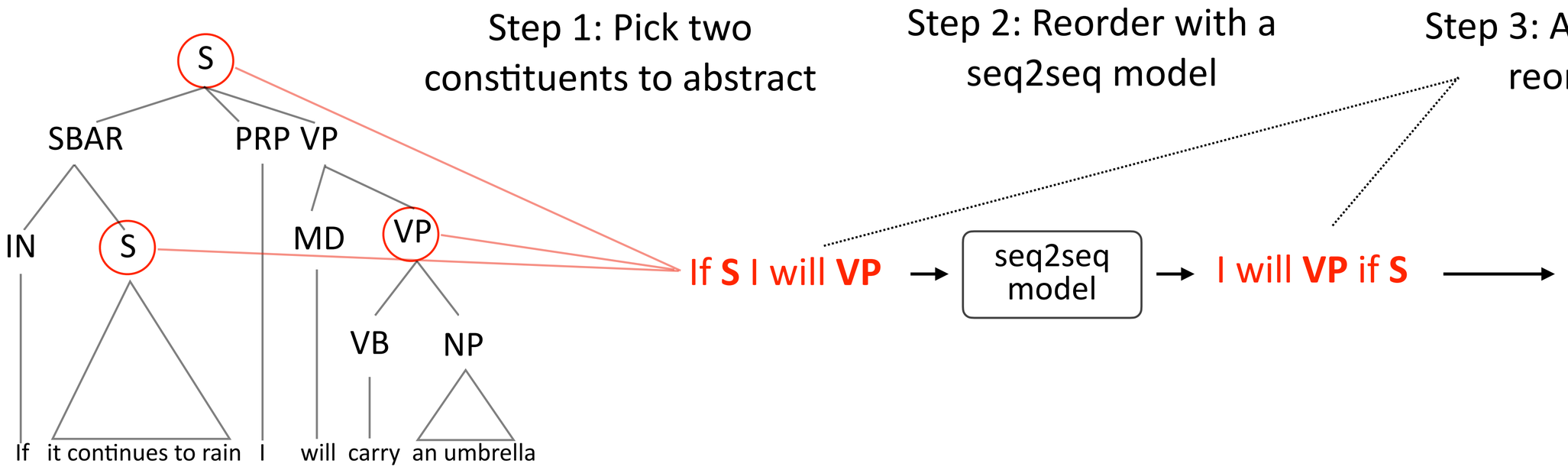
Step 3: Align and get reordering

If **S** I will **VP** → seq2seq model → I will **VP** if **S**

If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

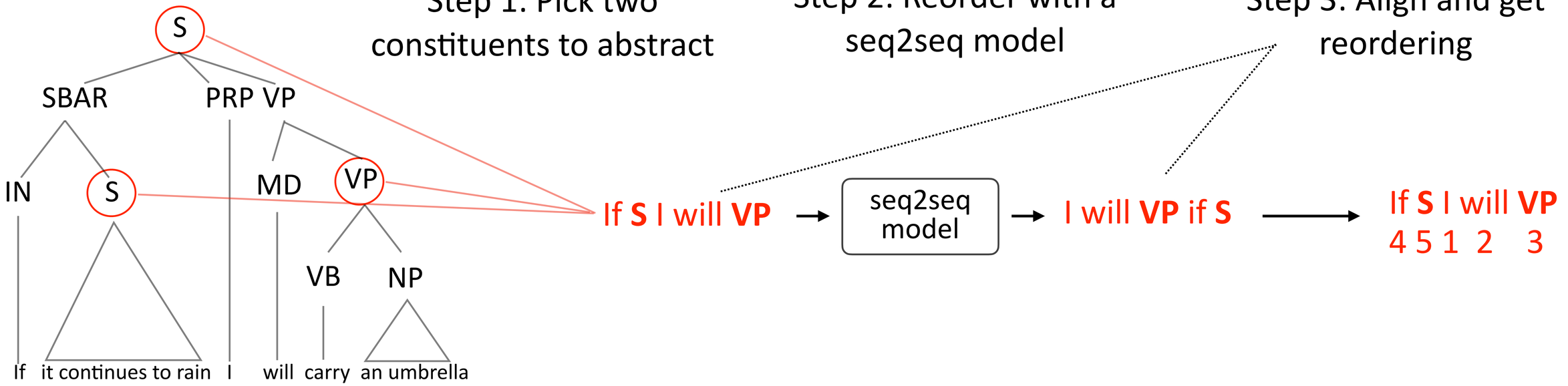
Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

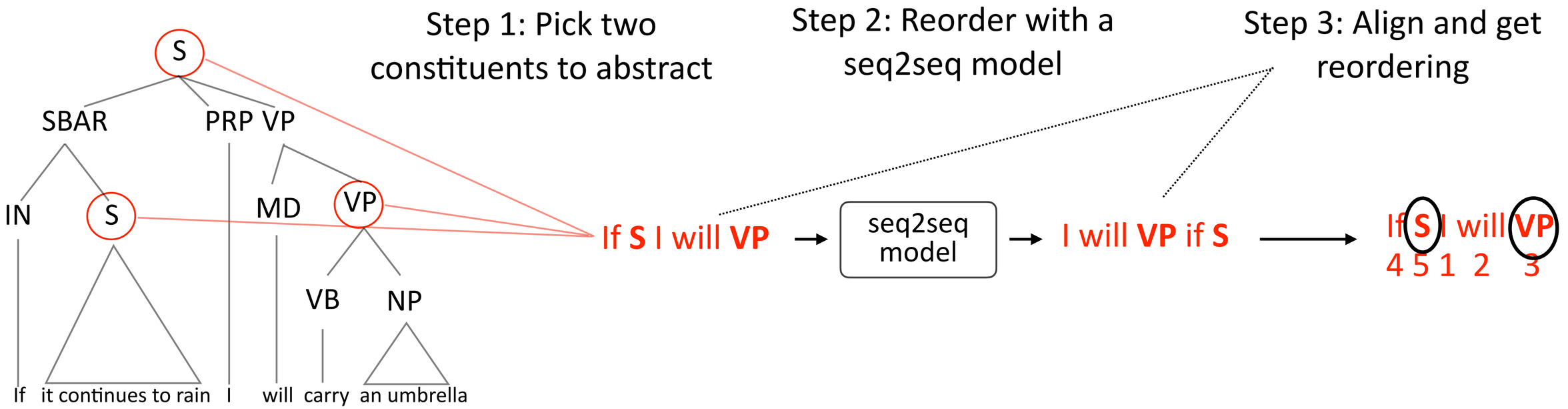
Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

SOW model: Source Order Rewriting

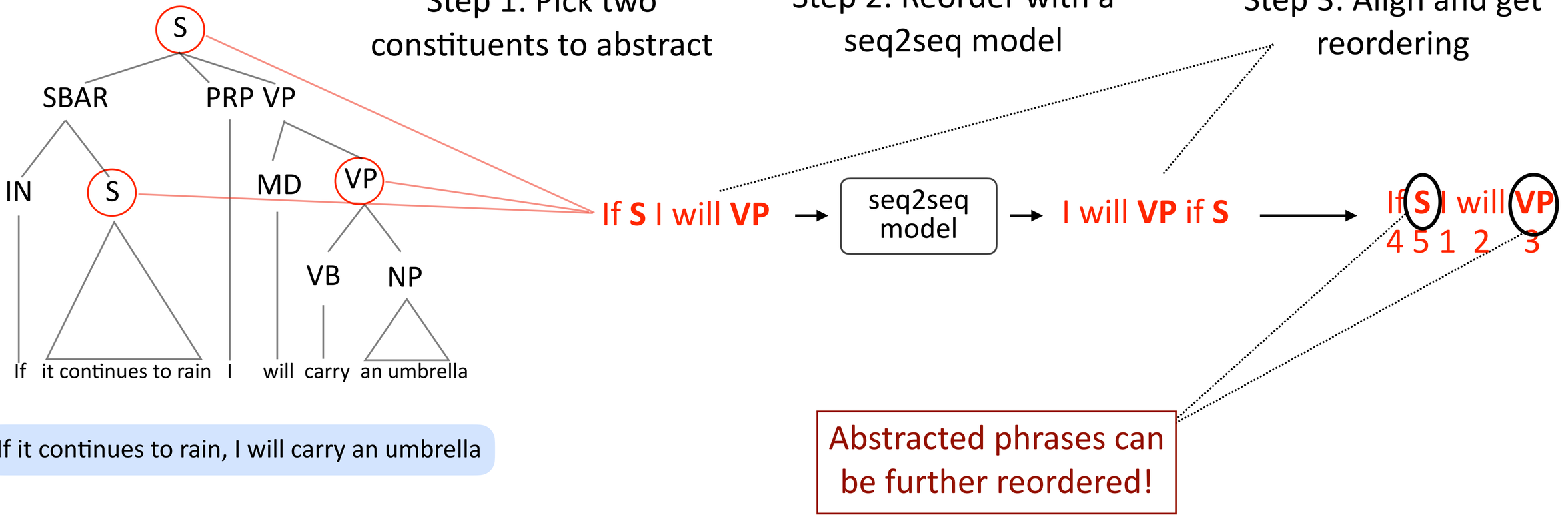
Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

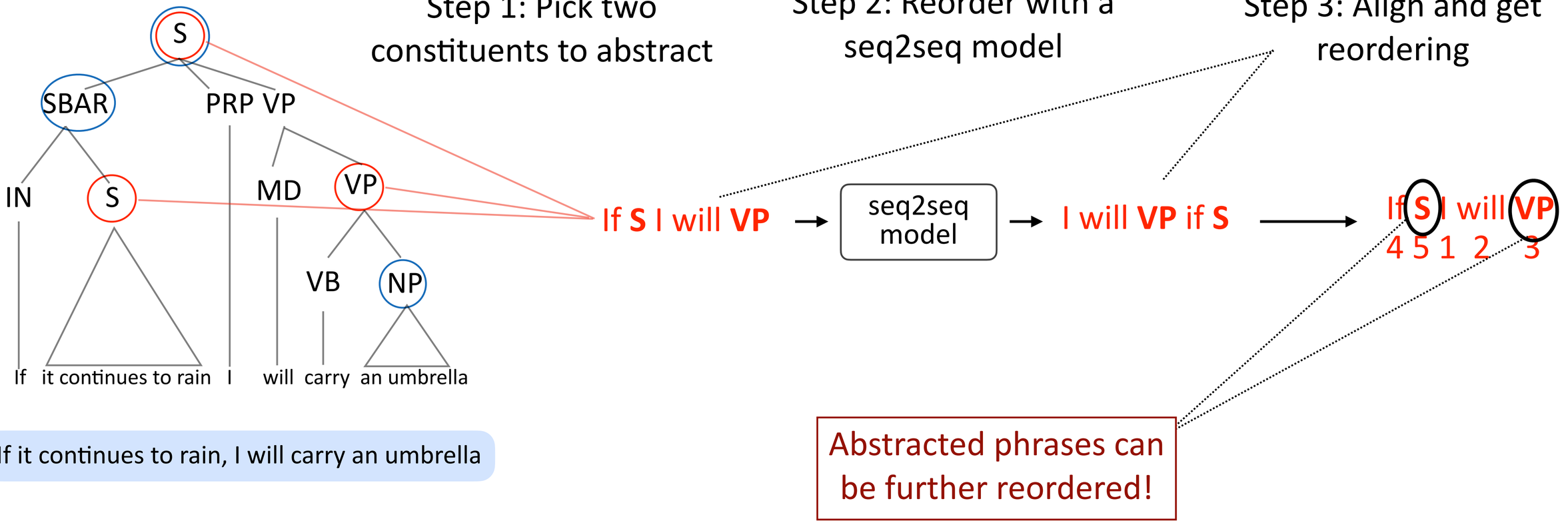
SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



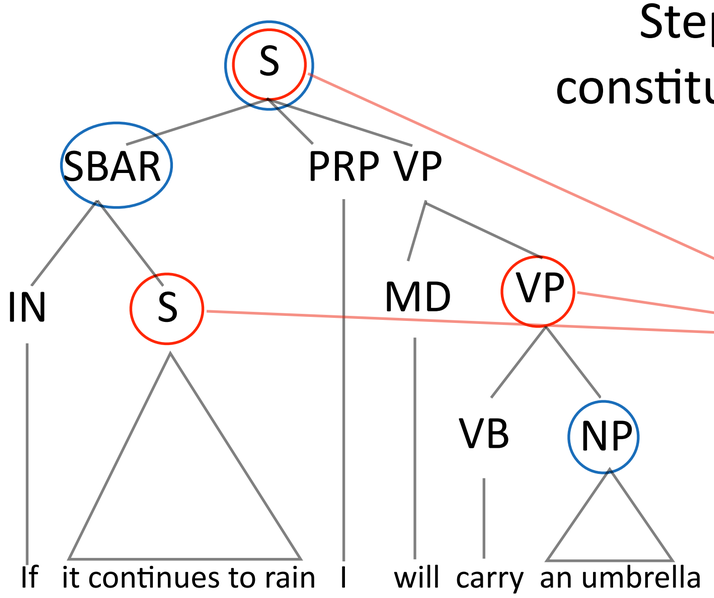
SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

Step 1: Pick two constituents to abstract

Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

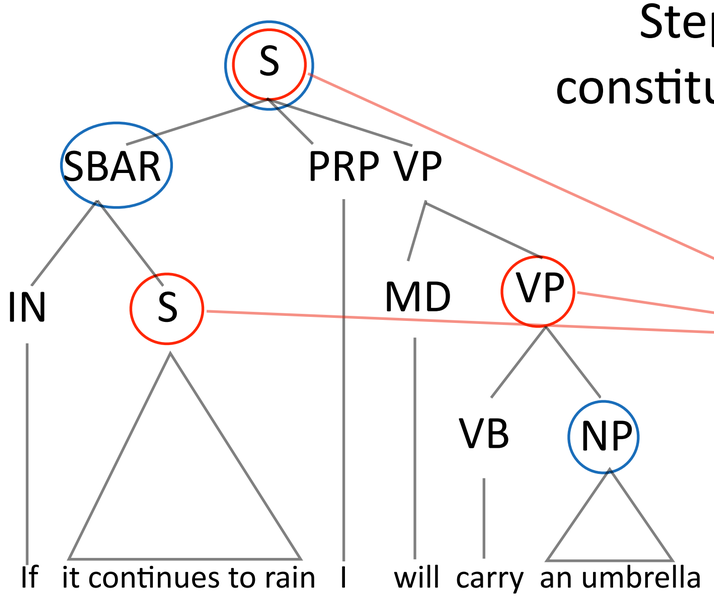
If **S** I will **VP** → seq2seq model → I will **VP** if **S** → If **S** I will **VP** (4 5 1 2 3)

SBAR I will carry **NP** → seq2seq model → **SBAR NP** I carry → **SBAR** I will carry **NP** (1 3 4 5 2)

Abstracted phrases can be further reordered!

SOW model: Source Order Rewriting

Goal: Given an input sentence, generate a set of appropriate reorderings.



If it continues to rain, I will carry an umbrella

Step 1: Pick two constituents to abstract

Step 2: Reorder with a seq2seq model

Step 3: Align and get reordering

If **S** I will **VP** → seq2seq model → I will **VP** if **S** → If **S** I will **VP** (4 5 1 2 3)

SBAR I will carry **NP** → seq2seq model → **SBAR NP** I carry → **SBAR** I will carry **NP** (1 3 4 5 2)

Abstracted phrases can be further reordered!

Output of the SOW model

Training Data: SOW

- We want phrase pairs of the following kind:

If **S** I will **VP** ⟶ I will **VP** if **S**

removing the **NN** from **NP** ⟶ **NP** was excluded from **NN**

Training Data: SOW

- We want phrase pairs of the following kind:

  If **S** I will **VP** ⟶ I will **VP** if **S**

  removing the **NN** from **NP** ⟶ **NP** was excluded from **NN**

- Paraphrase dataset (ParaNMT-50M) to extract training data.

Training Data: SOW

- We want phrase pairs of the following kind:

  If **S** I will **VP** ⟶ I will **VP** if **S**

  removing the **NN** from **NP** ⟶ **NP** was excluded from **NN**

- Paraphrase dataset (ParaNMT-50M) to extract training data.

If it continues to rain I will carry an umbrella

I will carry an umbrella if rain continues

Training Data: SOW

- We want phrase pairs of the following kind:

If **S** I will **VP** ⟶ I will **VP** if **S**

removing the **NN** from **NP** ⟶ **NP** was excluded from **NN**

- Paraphrase dataset (ParaNMT-50M) to extract training data.

If it continues to rain I will carry an umbrella

I will carry an umbrella if rain continues

Training Data: SOW

- We want phrase pairs of the following kind:

If **S** I will **VP** ⟶ I will **VP** if **S**

removing the **NN** from **NP** ⟶ **NP** was excluded from **NN**

- Paraphrase dataset (ParaNMT-50M) to extract training data.

If it continues to rain I will carry an umbrella

I will carry an umbrella if rain continues

If **S** I will **VP** ⟶ I will **VP** if **S**

REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.

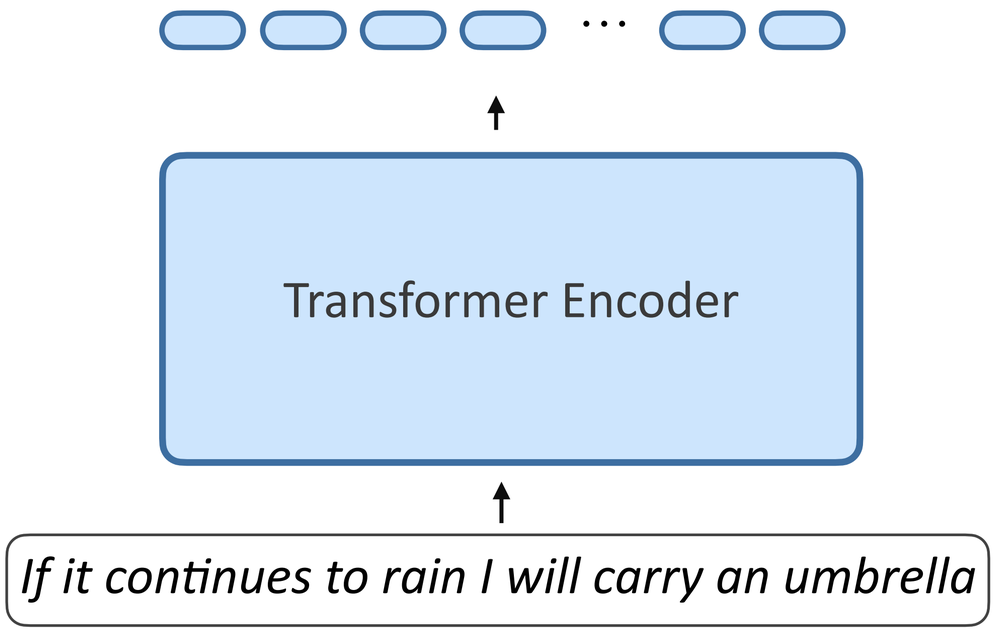
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.

If it continues to rain I will carry an umbrella

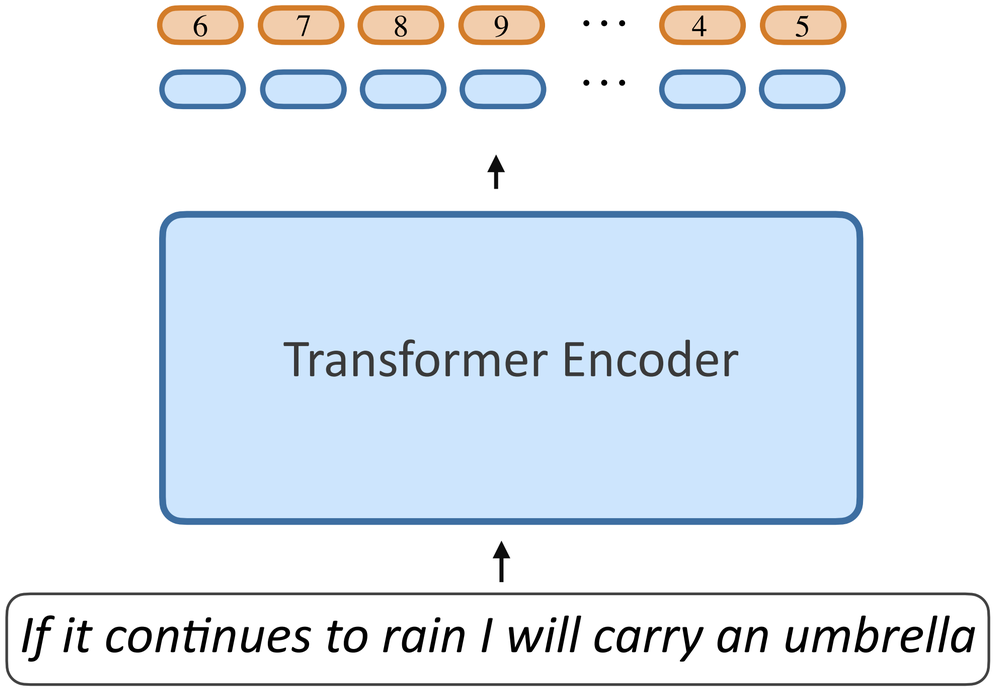
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



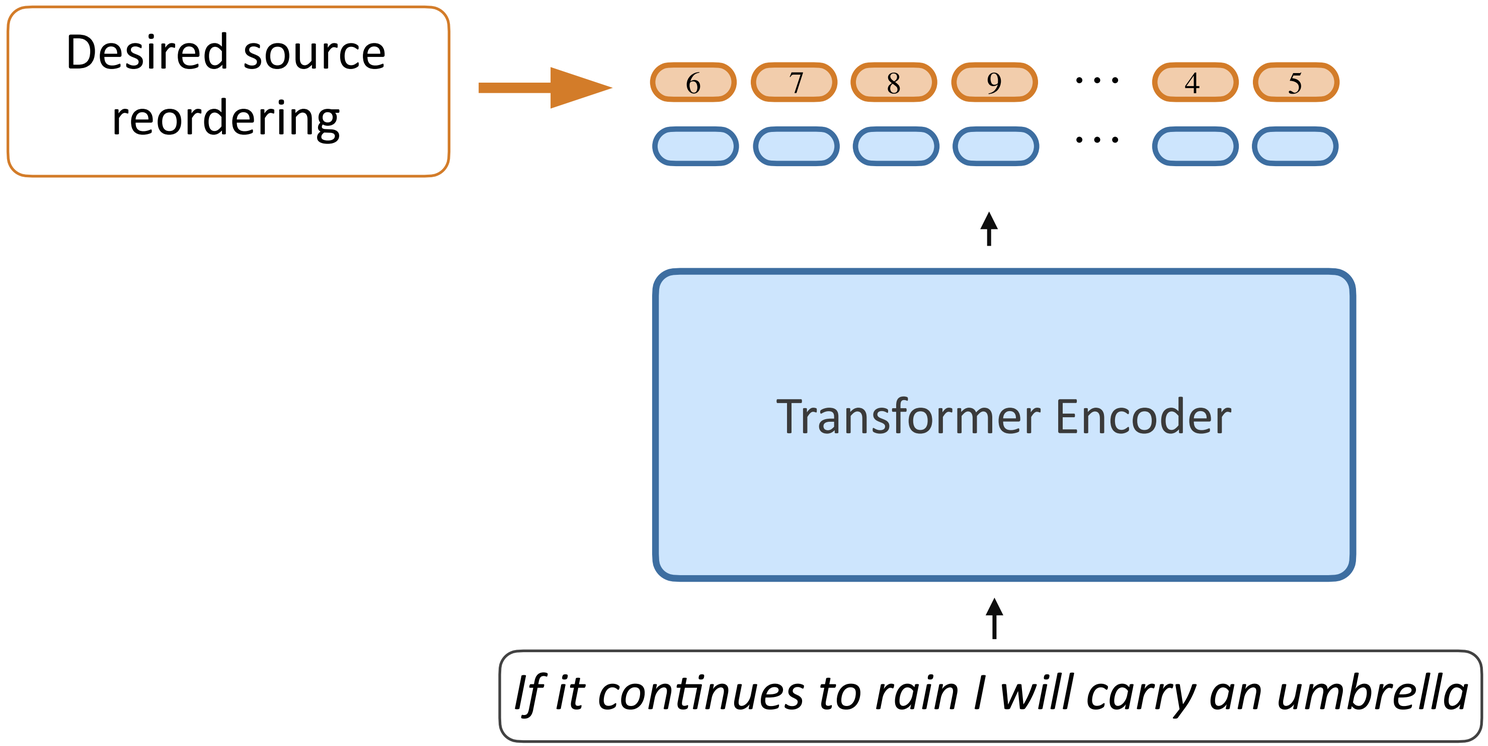
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



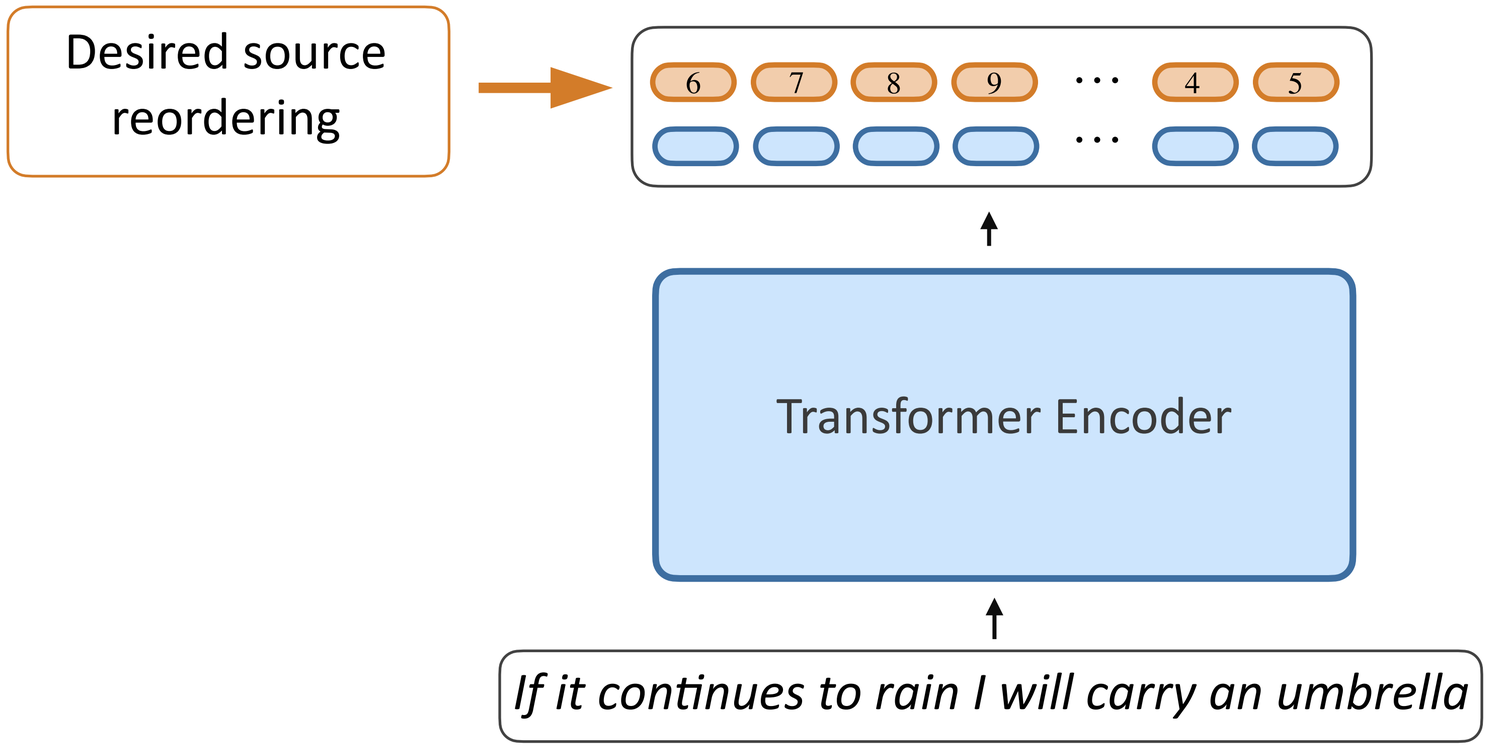
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



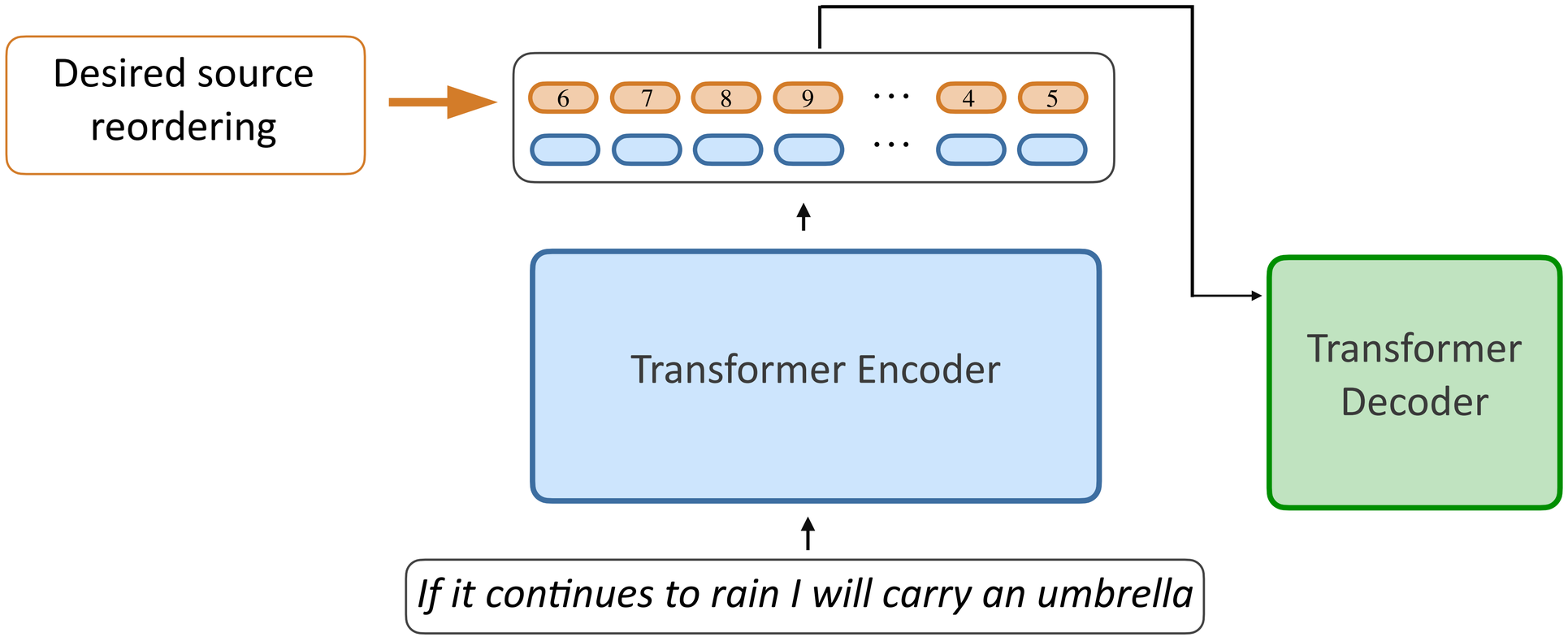
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



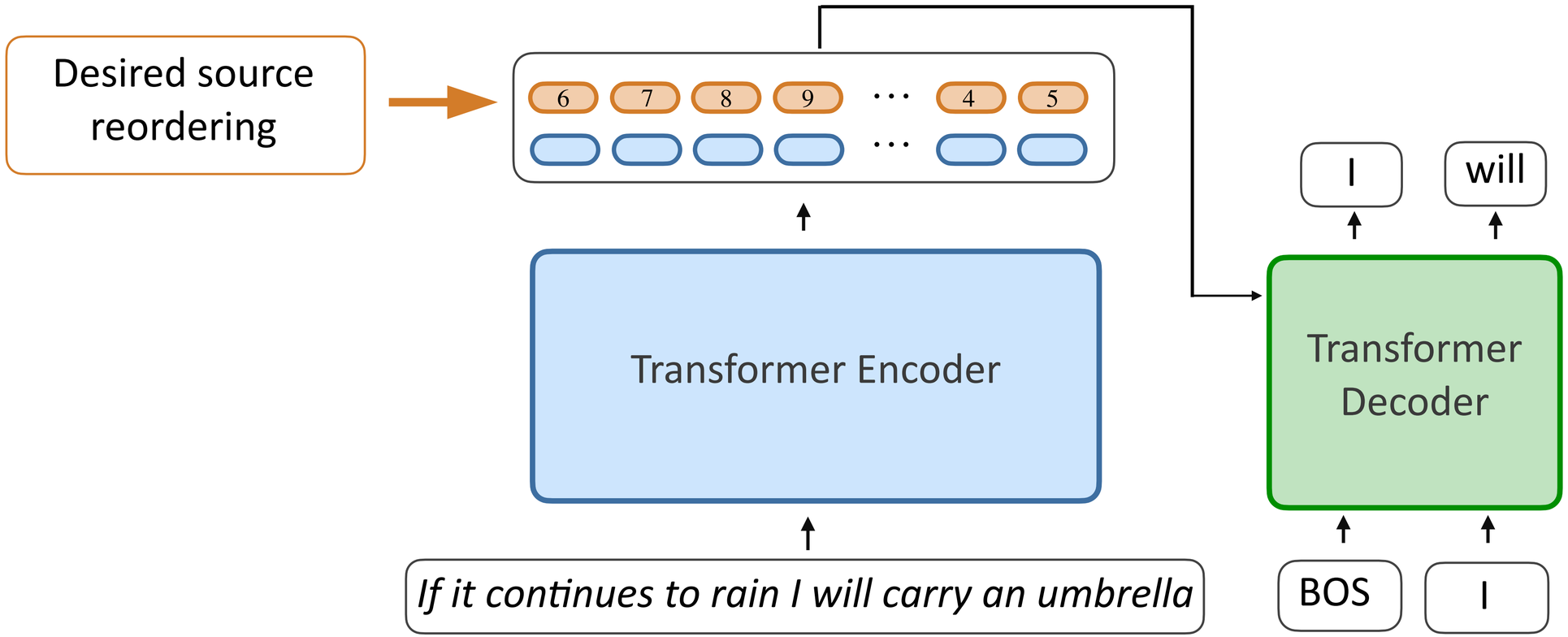
REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



REAP model: Rearrangement Aware Paraphrasing

Goal: Given an input sentence **x** and source reordering **r**, generate a paraphrase **y** roughly following the order specified by **r**.



Training Data: REAP

- We want sentence pairs with reordering information about the source sentence.

Clippers won the game → The game was won by Clippers
4 3 1 2

The sales went up in May → In May, the sales went up
3 4 5 6 1 2

Training Data: REAP

- We want sentence pairs with reordering information about the source sentence.

Clippers won the game → The game was won by Clippers
4 3 1 2

The sales went up in May → In May, the sales went up
3 4 5 6 1 2

- Extract training data from any paraphrase dataset (ParaNMT-50M) using word alignment to gold paraphrases:

Training Data: REAP

- We want sentence pairs with reordering information about the source sentence.

Clippers won the game → The game was won by Clippers
4 3 1 2

The sales went up in May → In May, the sales went up
3 4 5 6 1 2

- Extract training data from any paraphrase dataset (ParaNMT-50M) using word alignment to gold paraphrases:

Clippers won the game

The game was won by Clippers

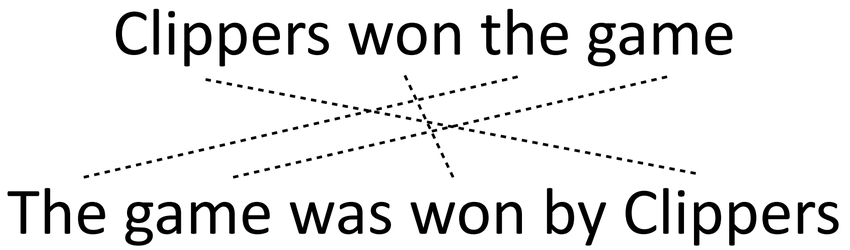
Training Data: REAP

- We want sentence pairs with reordering information about the source sentence.

Clippers won the game → The game was won by Clippers
4 3 1 2

The sales went up in May → In May, the sales went up
3 4 5 6 1 2

- Extract training data from any paraphrase dataset (ParaNMT-50M) using word alignment to gold paraphrases:



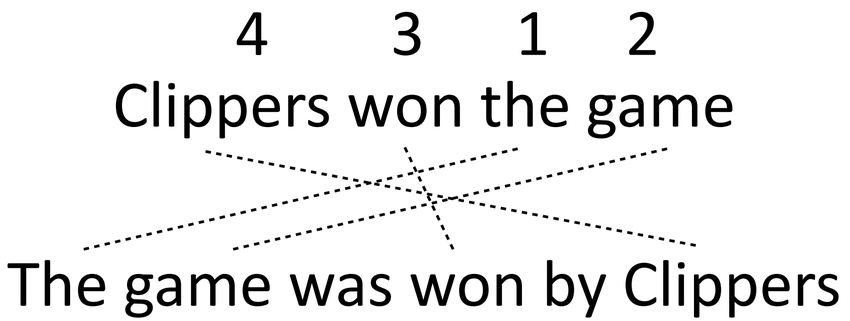
Training Data: REAP

- We want sentence pairs with reordering information about the source sentence.

Clippers won the game → The game was won by Clippers
4 3 1 2

The sales went up in May → In May, the sales went up
3 4 5 6 1 2

- Extract training data from any paraphrase dataset (ParaNMT-50M) using word alignment to gold paraphrases:



Evaluation: Dataset and Training

Evaluation: Dataset and Training

- ParaNMT dataset: English paraphrase pairs constructed using backtranslation.

Evaluation: Dataset and Training

- ParaNMT dataset: English paraphrase pairs constructed using backtranslation.

- Training Data
  SOW (phrase-level pairs): 1.5m
  REAP (sentence-level pairs): 350k

Evaluation: Dataset and Training

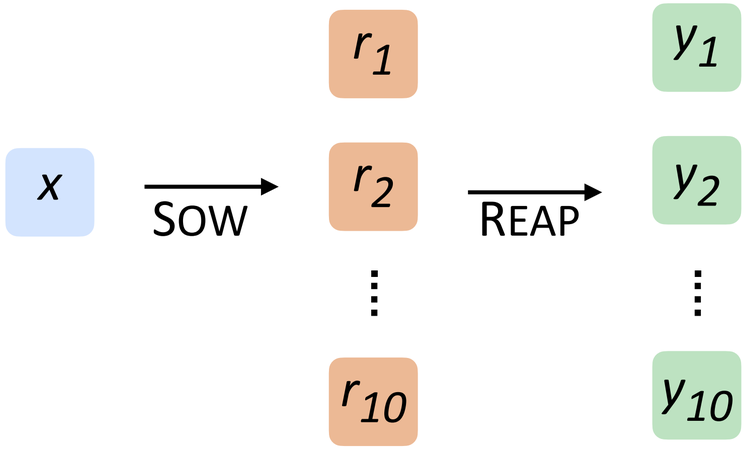
- ParaNMT dataset: English paraphrase pairs constructed using backtranslation.
- Training Data
  SOW (phrase-level pairs): 1.5m
  REAP (sentence-level pairs): 350k
- Testing (10k paraphrase pairs): for each of 10 reorderings, produce a paraphrase. Evaluate **quality** and **diversity** of these generated paraphrases

Evaluation: Dataset and Training

- ParaNMT dataset: English paraphrase pairs constructed using backtranslation.

- Training Data
  SOW (phrase-level pairs): 1.5m
  REAP (sentence-level pairs): 350k

- Testing (10k paraphrase pairs): for each of 10 reorderings, produce a paraphrase. Evaluate **quality** and **diversity** of these generated paraphrases

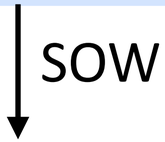


Example of a generated paraphrase

Despite recognition in European legislation, it
has proved difficult to implement the right of consumer associations to representation.

Example of a generated paraphrase

Despite recognition in European legislation, it
has proved difficult to implement the right of consumer associations to representation.



Example of a generated paraphrase

Despite recognition in European legislation, it has proved difficult to implement the right of consumer associations to representation.

SOW

Despite recognition PP, it VP. —> it VP despite recognition PP.

Example of a generated paraphrase

PP

Despite recognition in European legislation, it has proved difficult to implement the right of consumer associations to representation.

SOW

Despite recognition PP, it VP. —> it VP despite recognition PP.

Example of a generated paraphrase

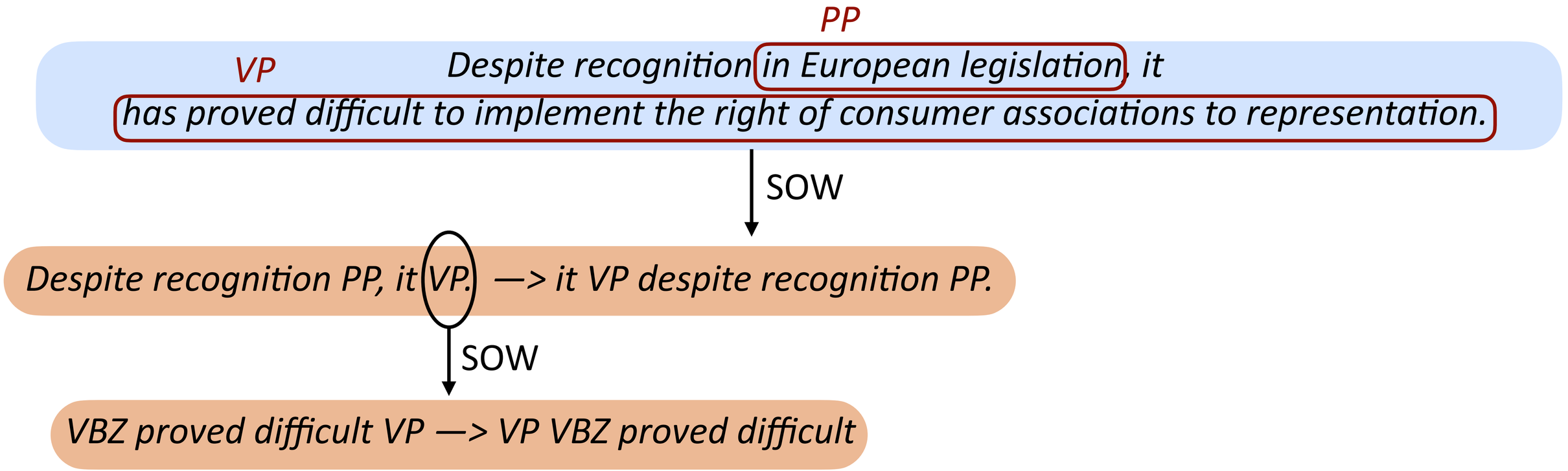
PP

VP *Despite recognition in European legislation, it has proved difficult to implement the right of consumer associations to representation.*

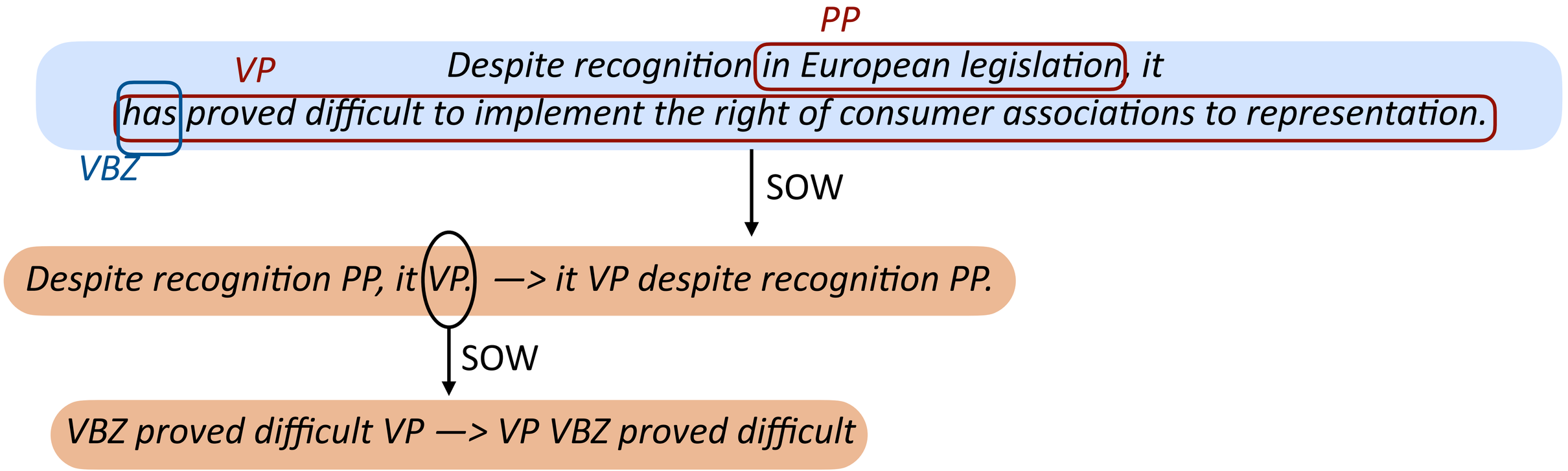
SOW

Despite recognition PP, it VP. —> it VP despite recognition PP.

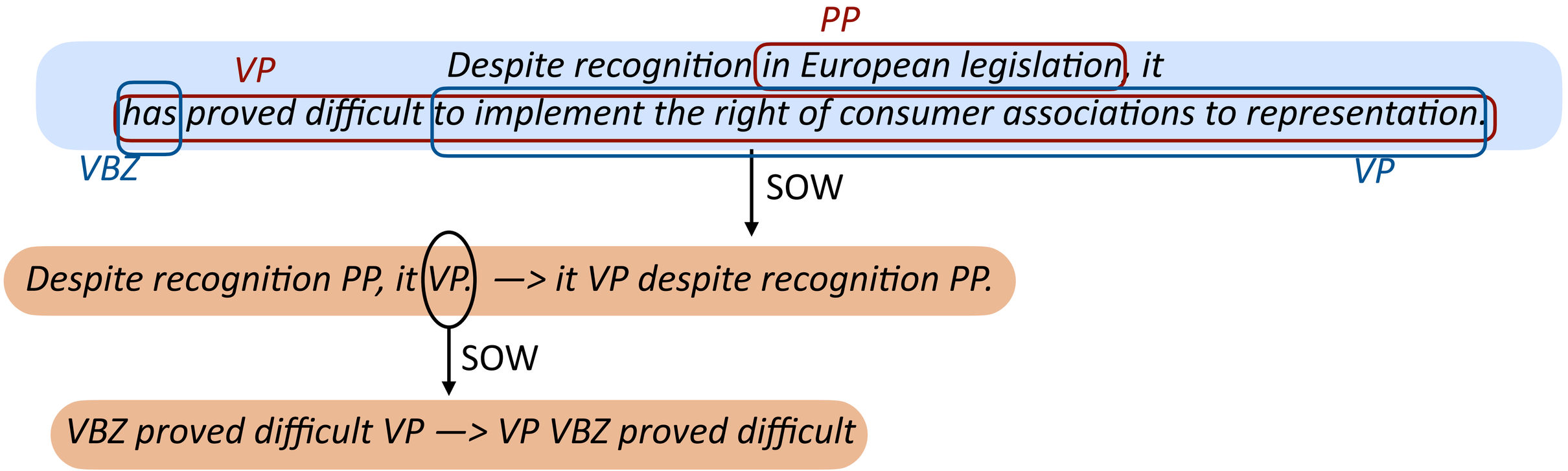
Example of a generated paraphrase



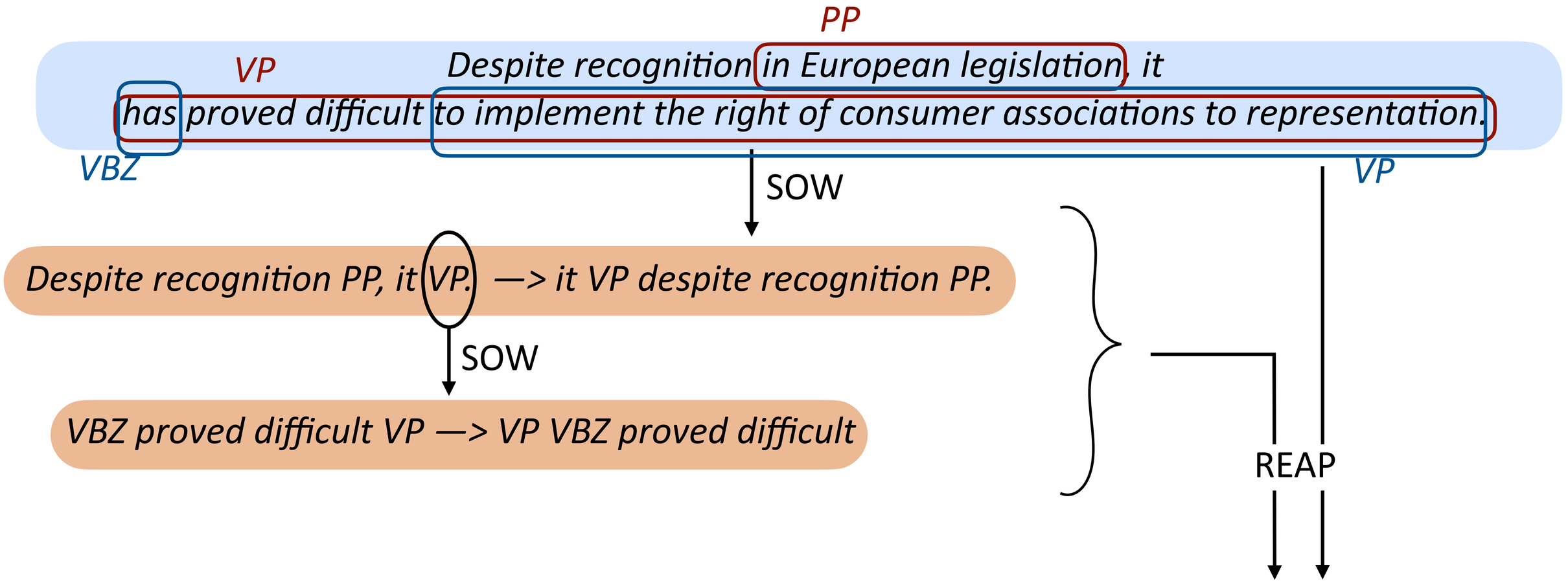
Example of a generated paraphrase



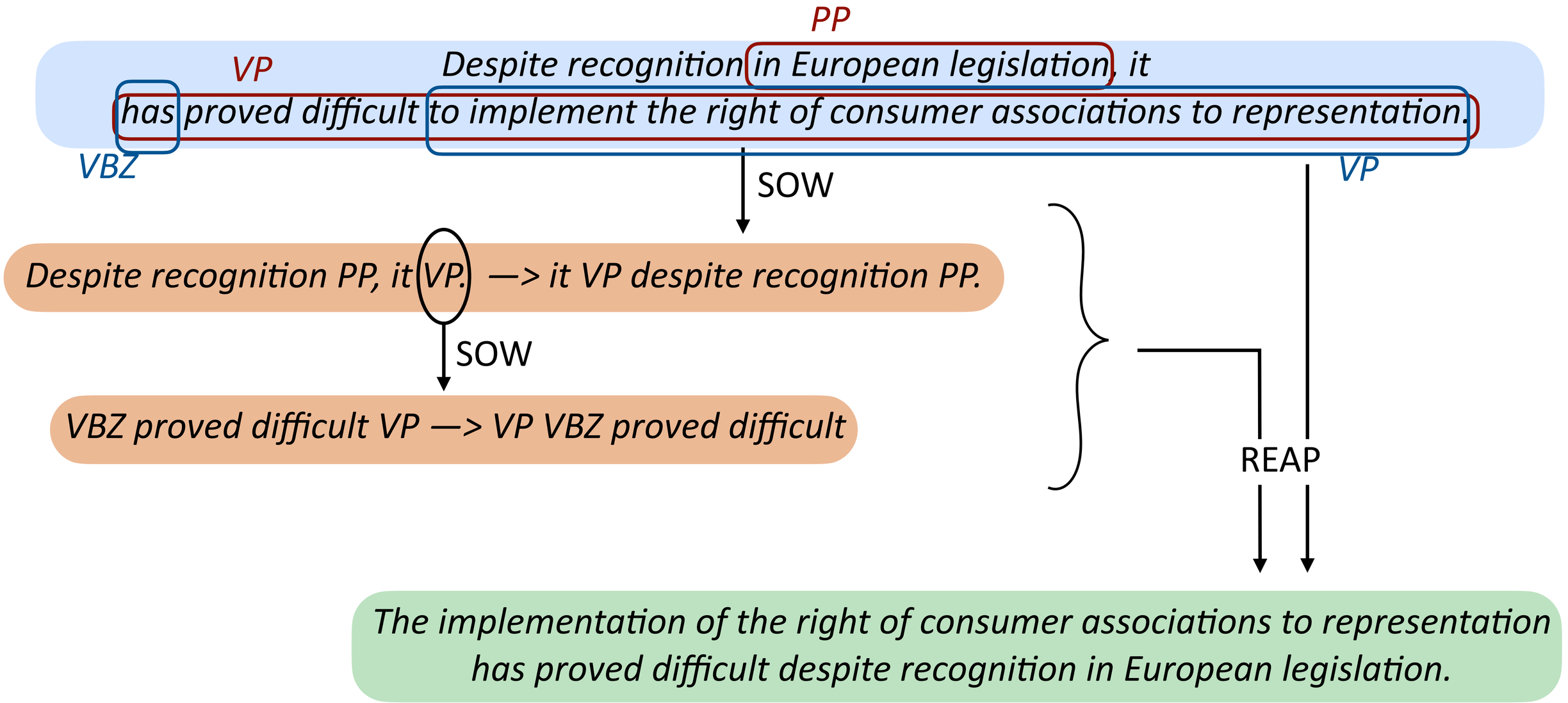
Example of a generated paraphrase



Example of a generated paraphrase



Example of a generated paraphrase



Comparison with prior work

Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

SCPN
(Iyyer et al. NAACL'18)

Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Comparison with prior work

Quality (higher is better)

Homogeneity (lower is better)

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Comparison with prior work

Quality (higher is better)

oracle-BLEU

best BLEU score w.r.t. to the target among the 10 paraphrases

Homogeneity (lower is better)

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Comparison with prior work

	Quality (higher is better)	Homogeneity (lower is better)
	oracle-BLEU best BLEU score w.r.t. to the target among the 10 paraphrases	self-BLEU pair-wise BLEU between generated paraphrases for the same input
Transformer seq2seq		
Top-k decoding		
Diverse-decoding (Kumar et al. NAACL'19)		
SCPN (Iyyer et al. NAACL'18)		
SOW-REAP		

Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

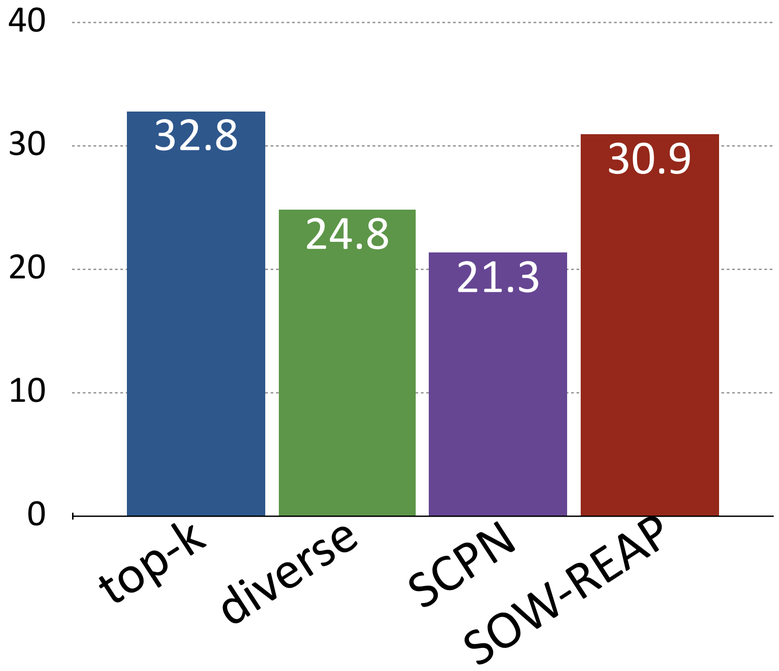
SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Quality (higher is better)

oracle-BLEU

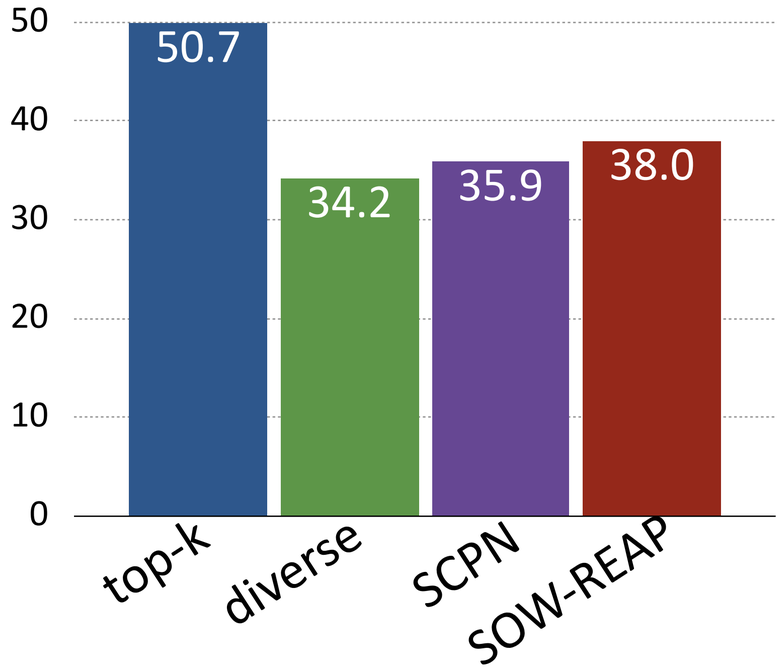
best BLEU score w.r.t. to the target among the 10 paraphrases



Homogeneity (lower is better)

self-BLEU

pair-wise BLEU between generated paraphrases for the same input



Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

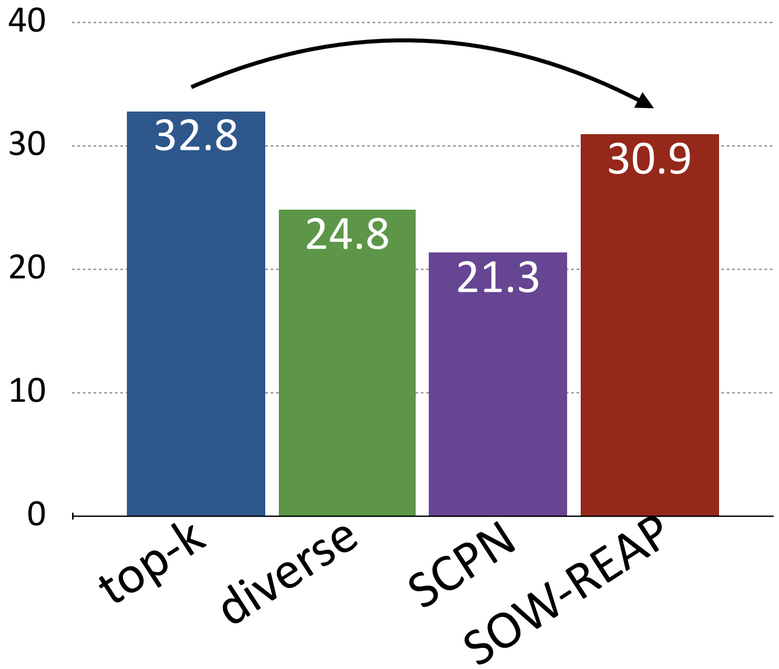
SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Quality (higher is better)

oracle-BLEU

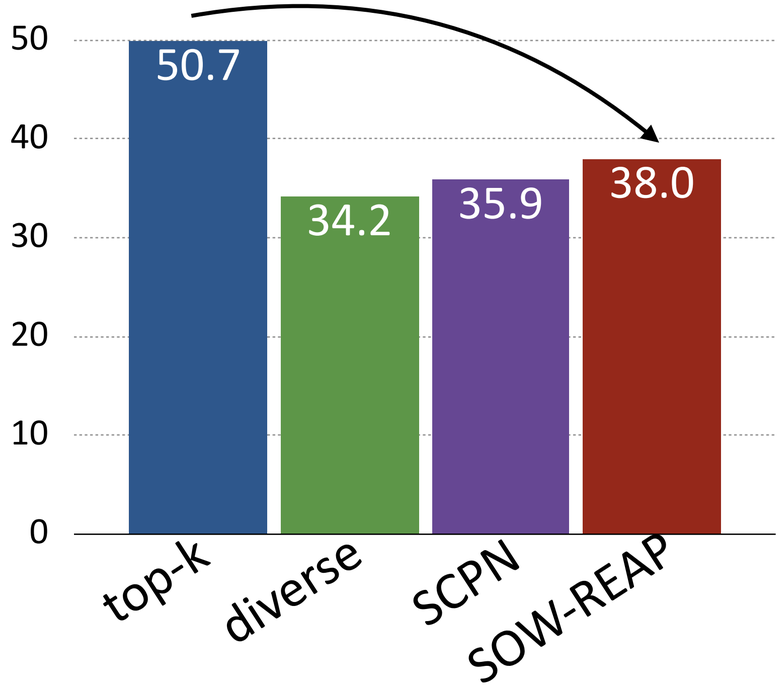
best BLEU score w.r.t. to the target among the 10 paraphrases



Homogeneity (lower is better)

self-BLEU

pair-wise BLEU between generated paraphrases for the same input



Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

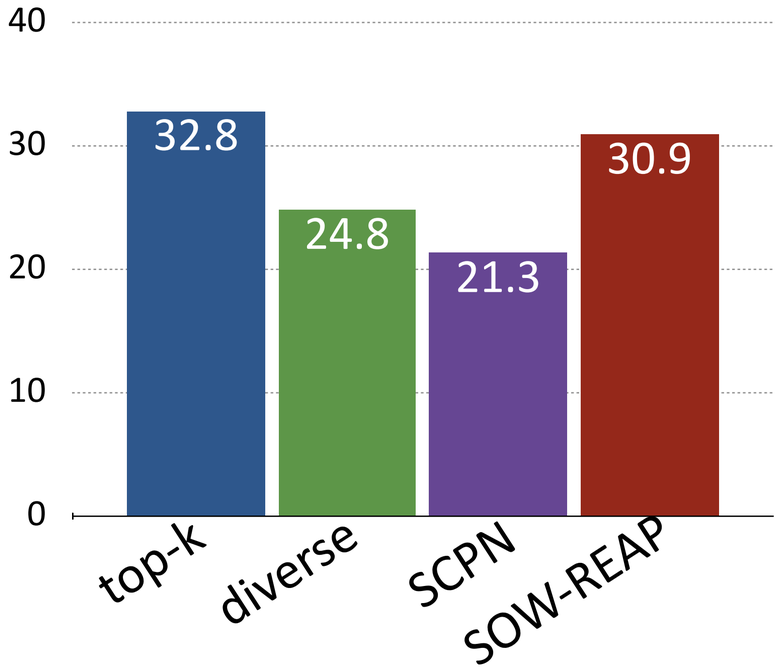
SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Quality (higher is better)

oracle-BLEU

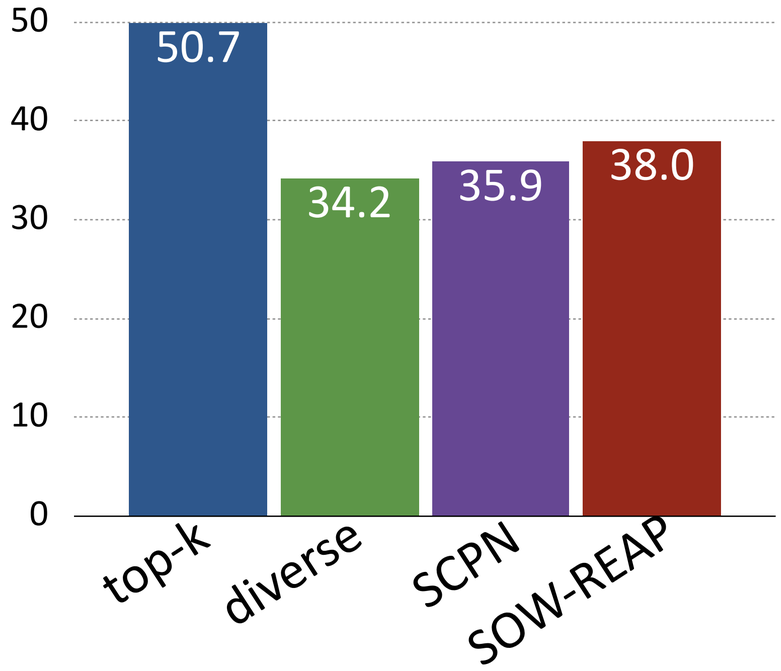
best BLEU score w.r.t. to the target among the 10 paraphrases



Homogeneity (lower is better)

self-BLEU

pair-wise BLEU between generated paraphrases for the same input



Comparison with prior work

Transformer seq2seq

Top-k decoding

Diverse-decoding
(Kumar et al. NAACL'19)

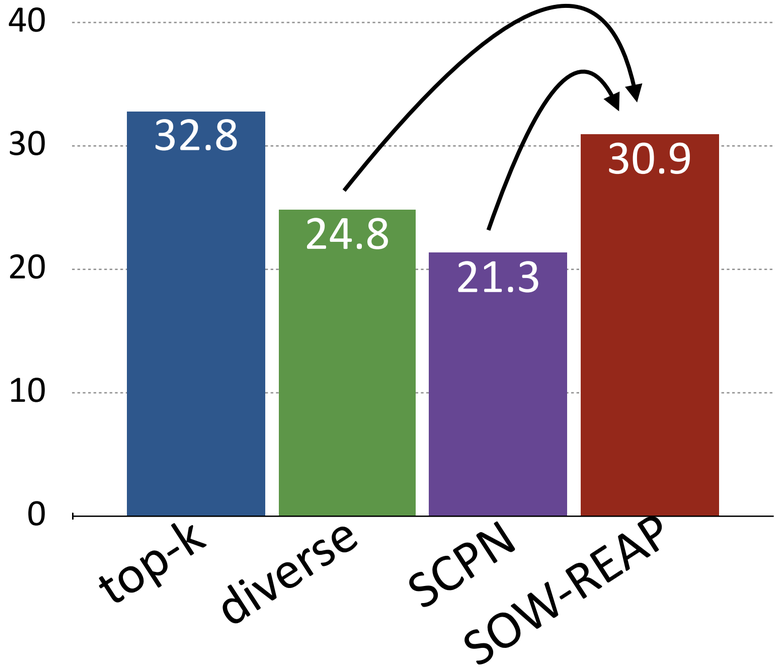
SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

Quality (higher is better)

oracle-BLEU

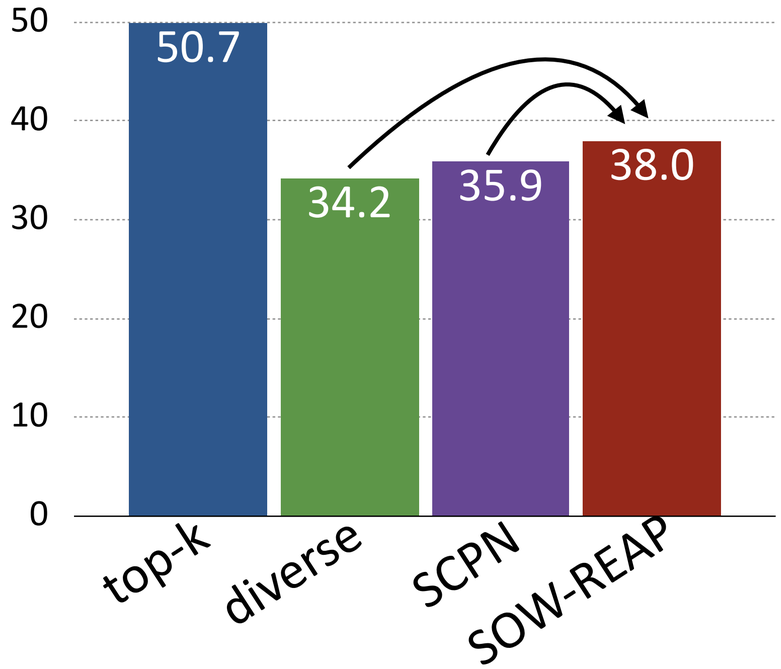
best BLEU score w.r.t. to the target among the 10 paraphrases



Homogeneity (lower is better)

self-BLEU

pair-wise BLEU between generated paraphrases for the same input



Comparison with prior work

Transformer seq2seq

Top-k decoding

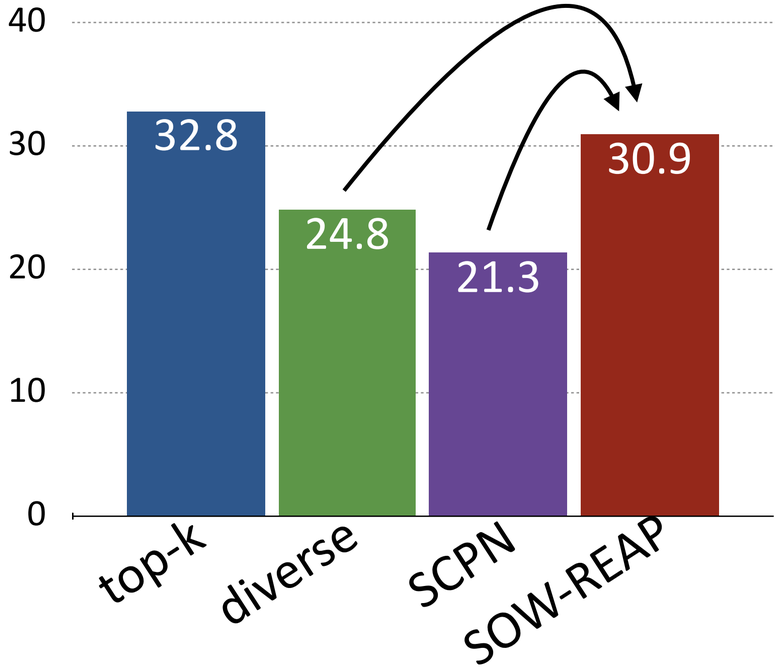
Diverse-decoding
(Kumar et al. NAACL'19)

SCPN
(Iyyer et al. NAACL'18)

SOW-REAP

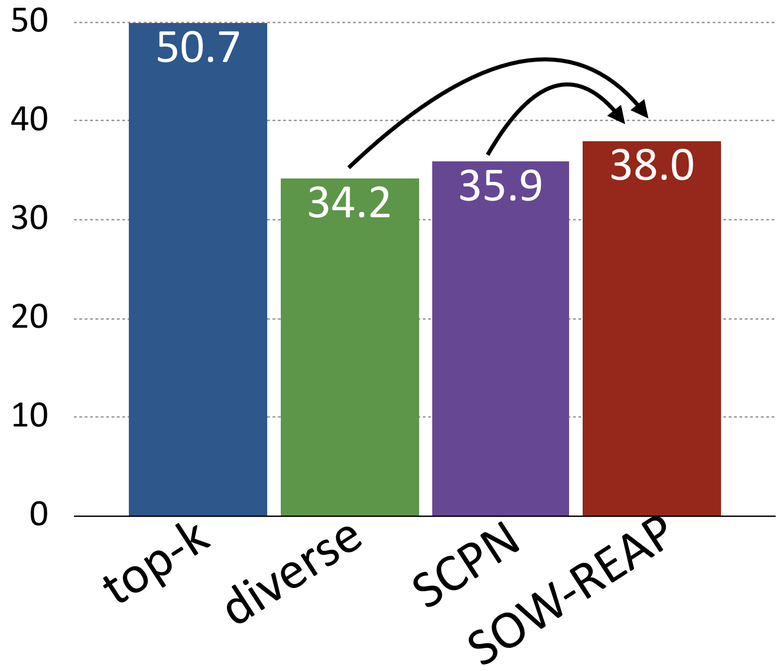
Quality (higher is better)

oracle-BLEU
best BLEU score w.r.t. to the target among the 10 paraphrases



Homogeneity (lower is better)

self-BLEU
pair-wise BLEU between generated paraphrases for the same input



SOW-REAP achieves a better quality-diversity tradeoff compared to baselines!

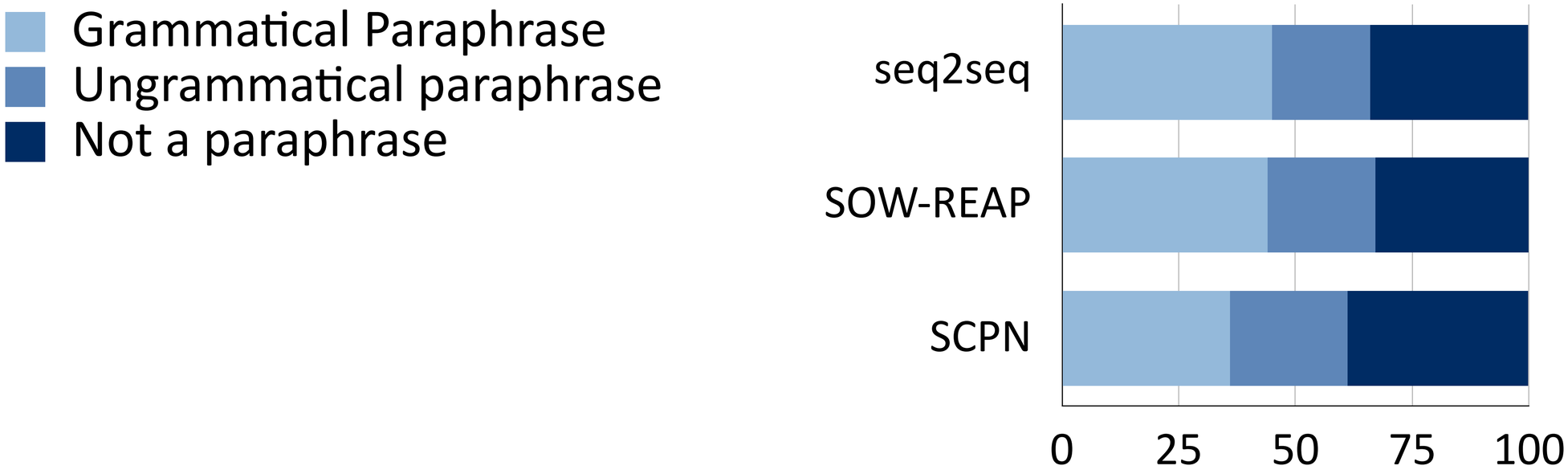
Human Evaluation

Human Evaluation

- Rate 100 sampled paraphrases from each model on a 3-point quality scale with MTurk

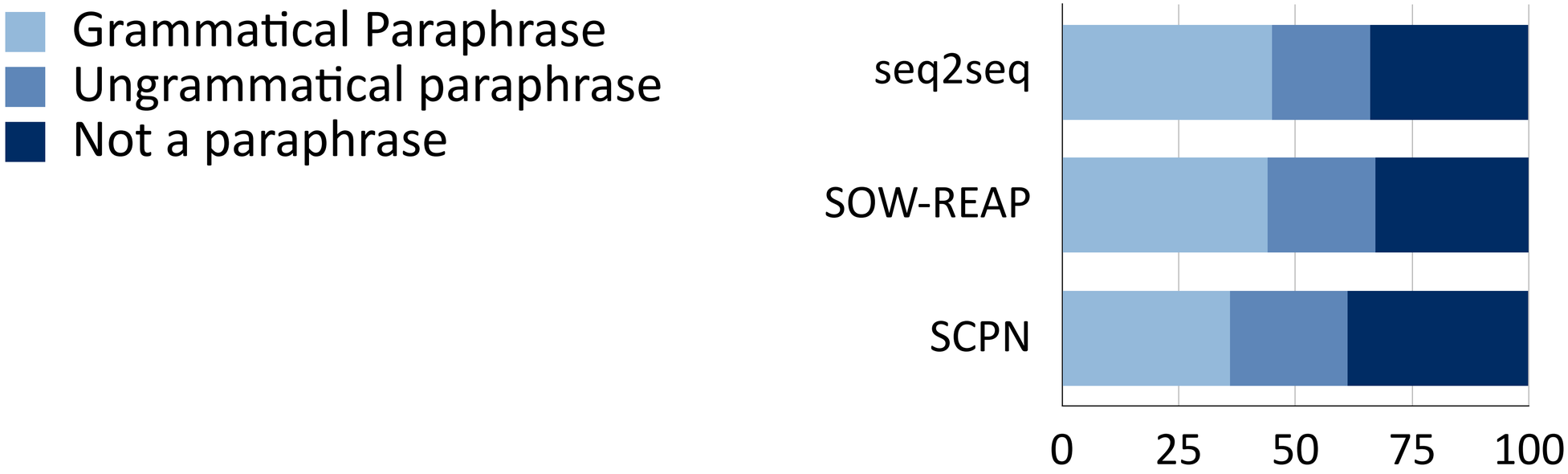
Human Evaluation

- Rate 100 sampled paraphrases from each model on a 3-point quality scale with MTurk



Human Evaluation

- Rate 100 sampled paraphrases from each model on a 3-point quality scale with MTurk



- Quality of ours is **as high as** the basic seq2seq model

Takeaways

Takeaways

- Learning syntactic transformations provide a flexible way to enumerate different paraphrasing phenomenon.

Takeaways

- Learning syntactic transformations provide a flexible way to enumerate different paraphrasing phenomenon.
- Neural models can be trained to follow these syntactic guides and produce diverse generations.

Takeaways

- Learning syntactic transformations provide a flexible way to enumerate different paraphrasing phenomenon.
- Neural models can be trained to follow these syntactic guides and produce diverse generations.

Thanks!