Improving dependency parsing using three methods

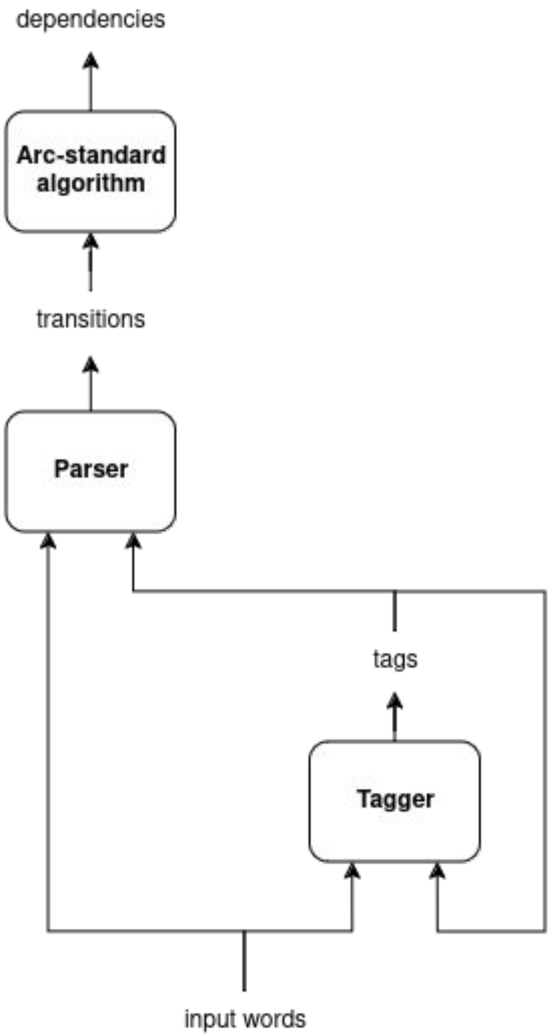
Intro

- Non-projective dependency parsing
- Beam search
- Attention mechanisms

Background

Baseline pipeline

- Part-of-speech tagging model
- Dependency parsing model
- Arc-standard algorithm



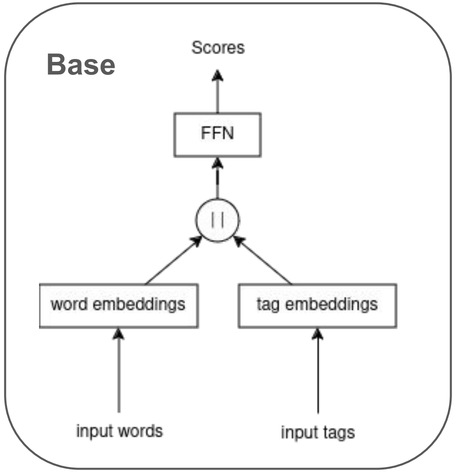
Background

Tagger and parser, same base model

Fixed window

Embeddings

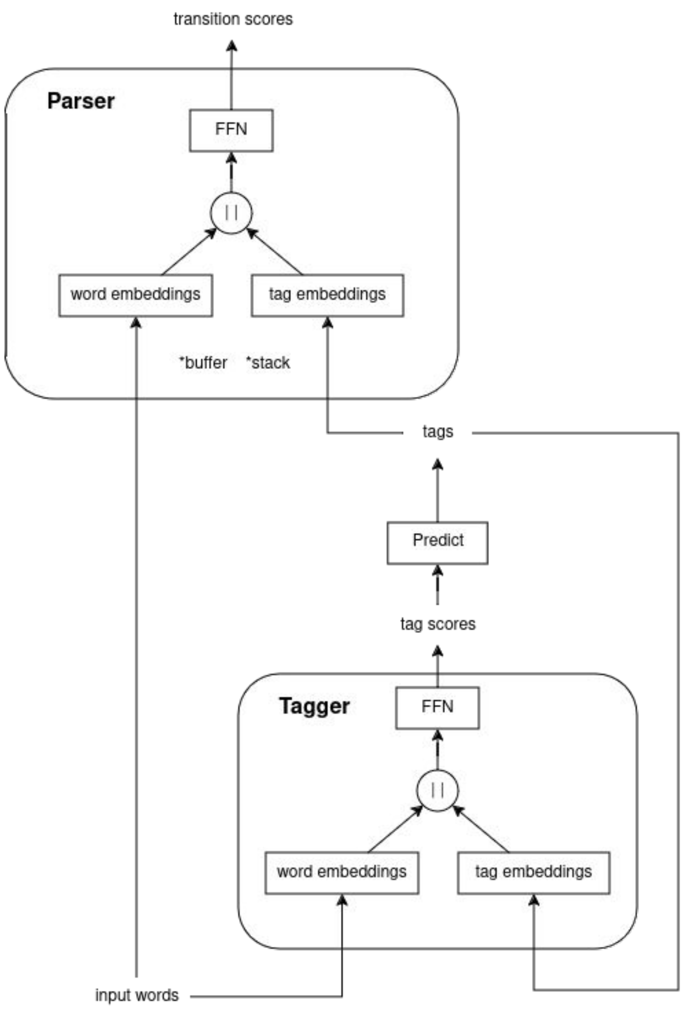
Feed-forward network



Background

Tagger and parser, same base model

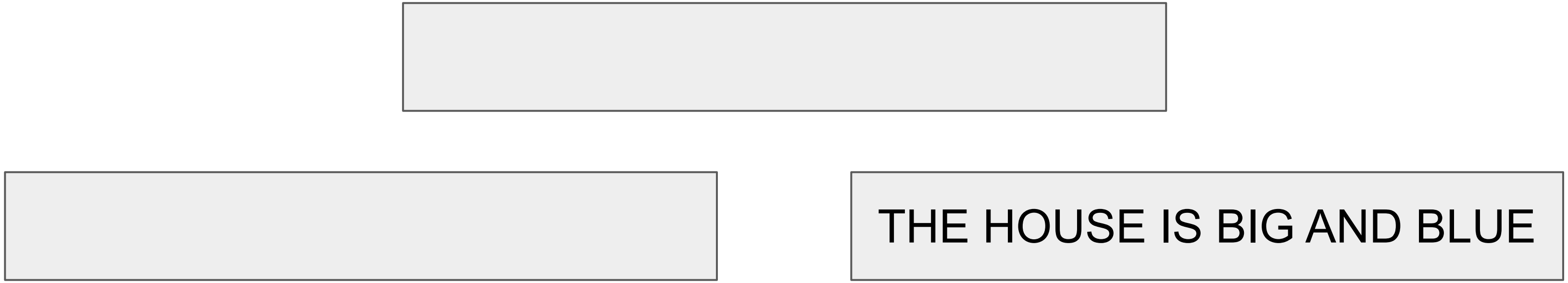
- Fixed window
- Embeddings
- Feed-forward network



DEPENDENCY TREE

STACK

BUFFER



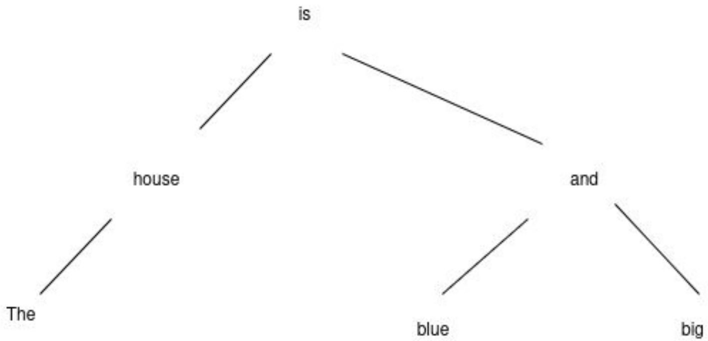
SH : Shift transition - Shift a word from buffer to stack

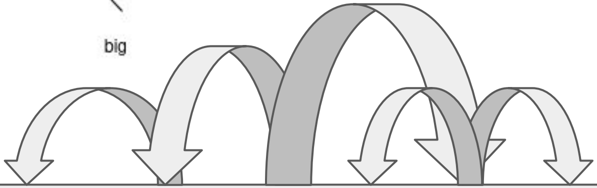
LA : Left Arc - Makes top of stack head of second and pops second

RA : Right Arc - Makes second of stack head of top and pops top

THE HOUSE IS BIG AND BLUE

THE HOUSE IS BIG AND BLUE





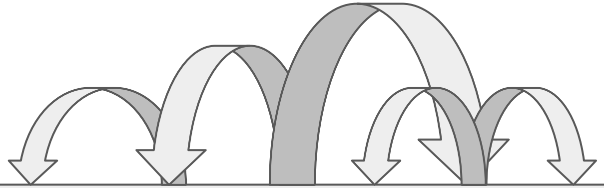
THE HOUSE IS BIG AND BLUE

THE HOUSE IS BIG AND BLUE

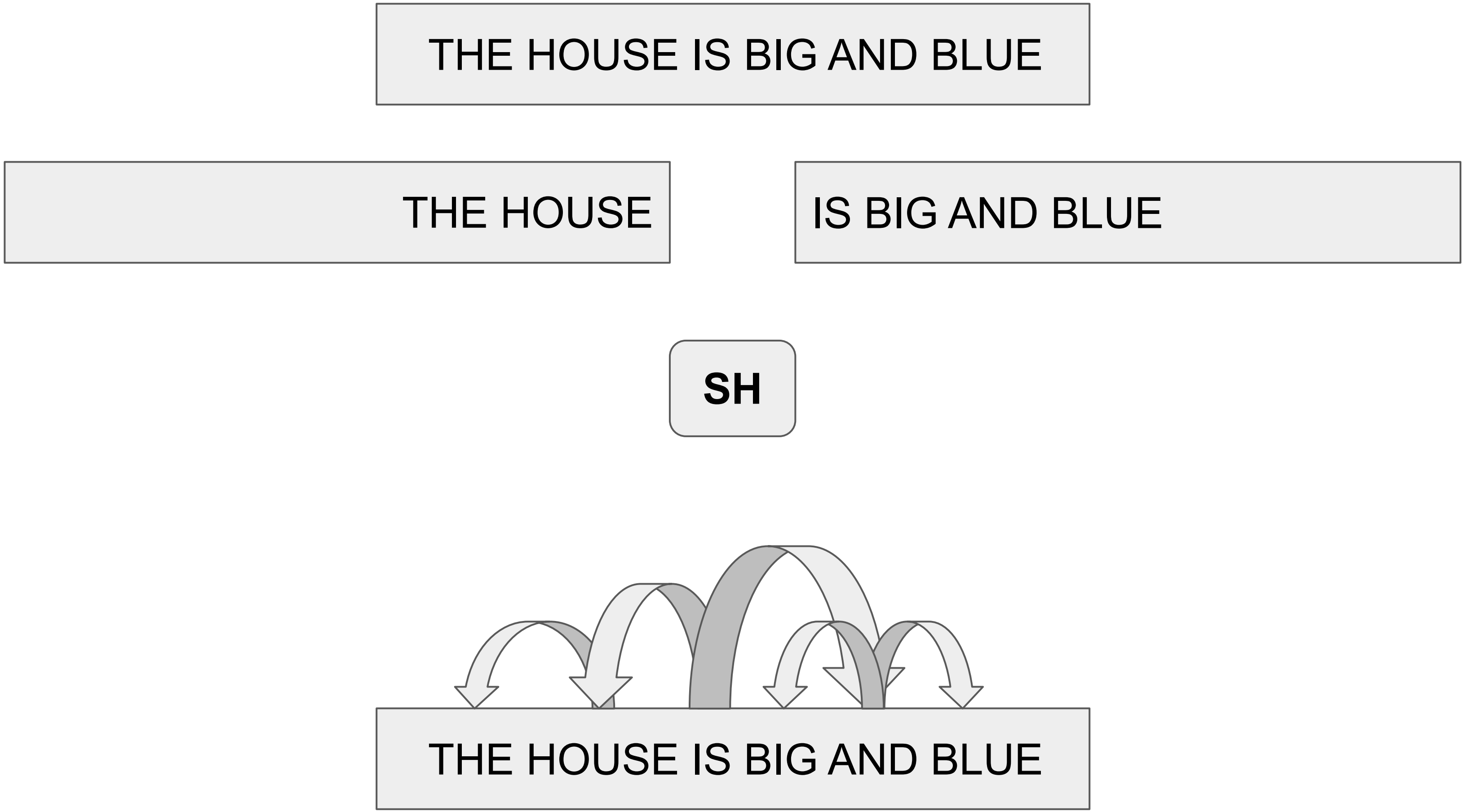
THE

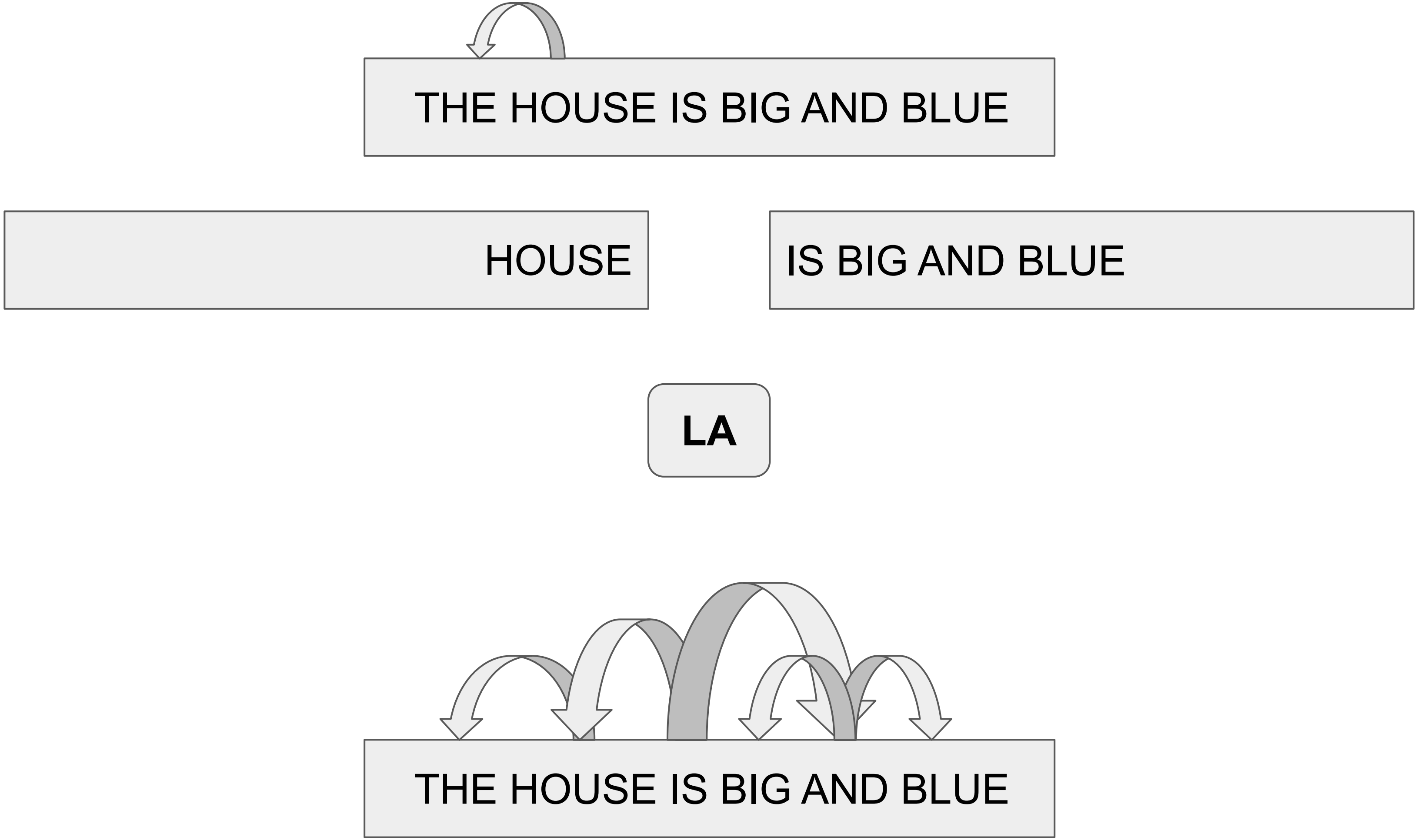
HOUSE IS BIG AND BLUE

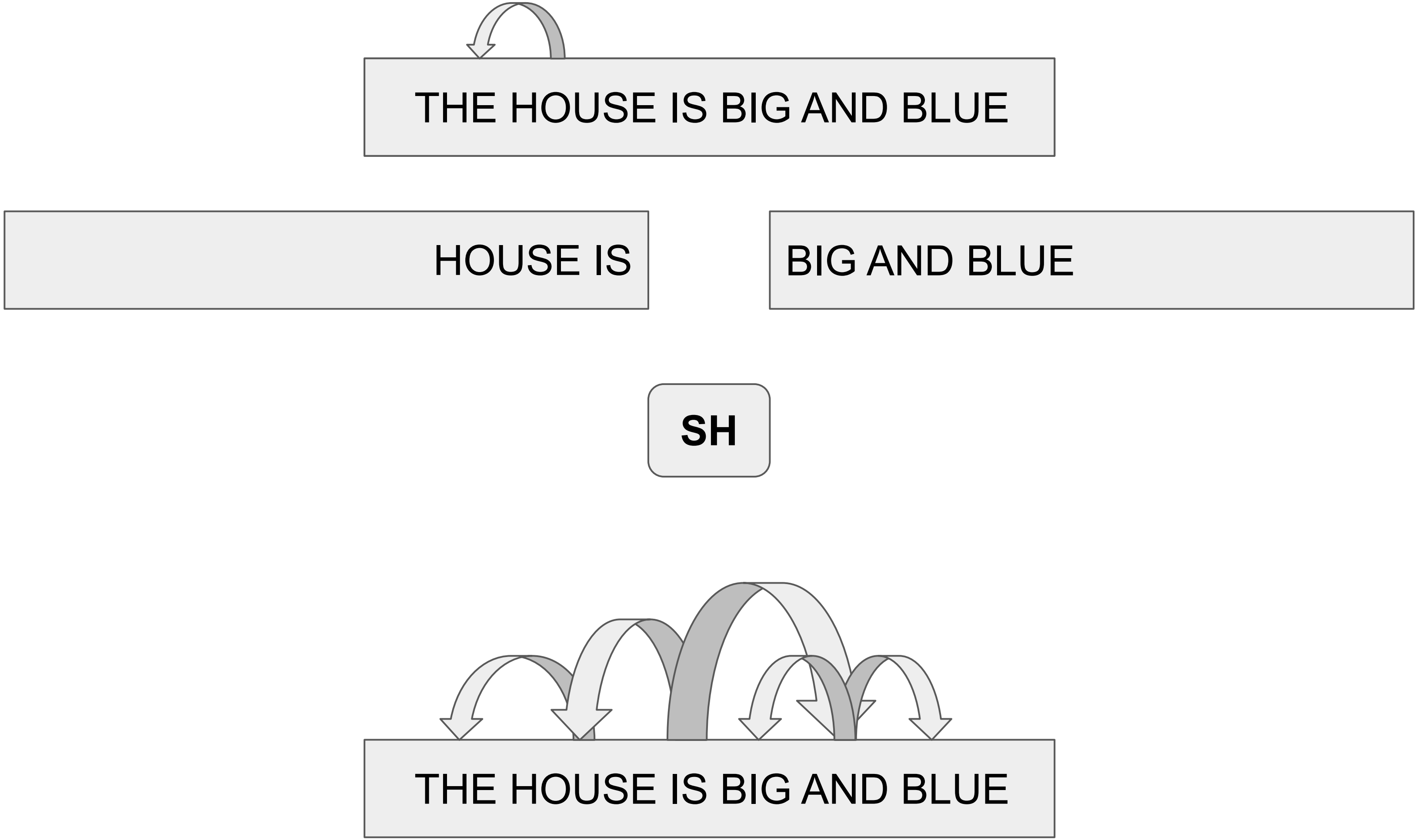
SH

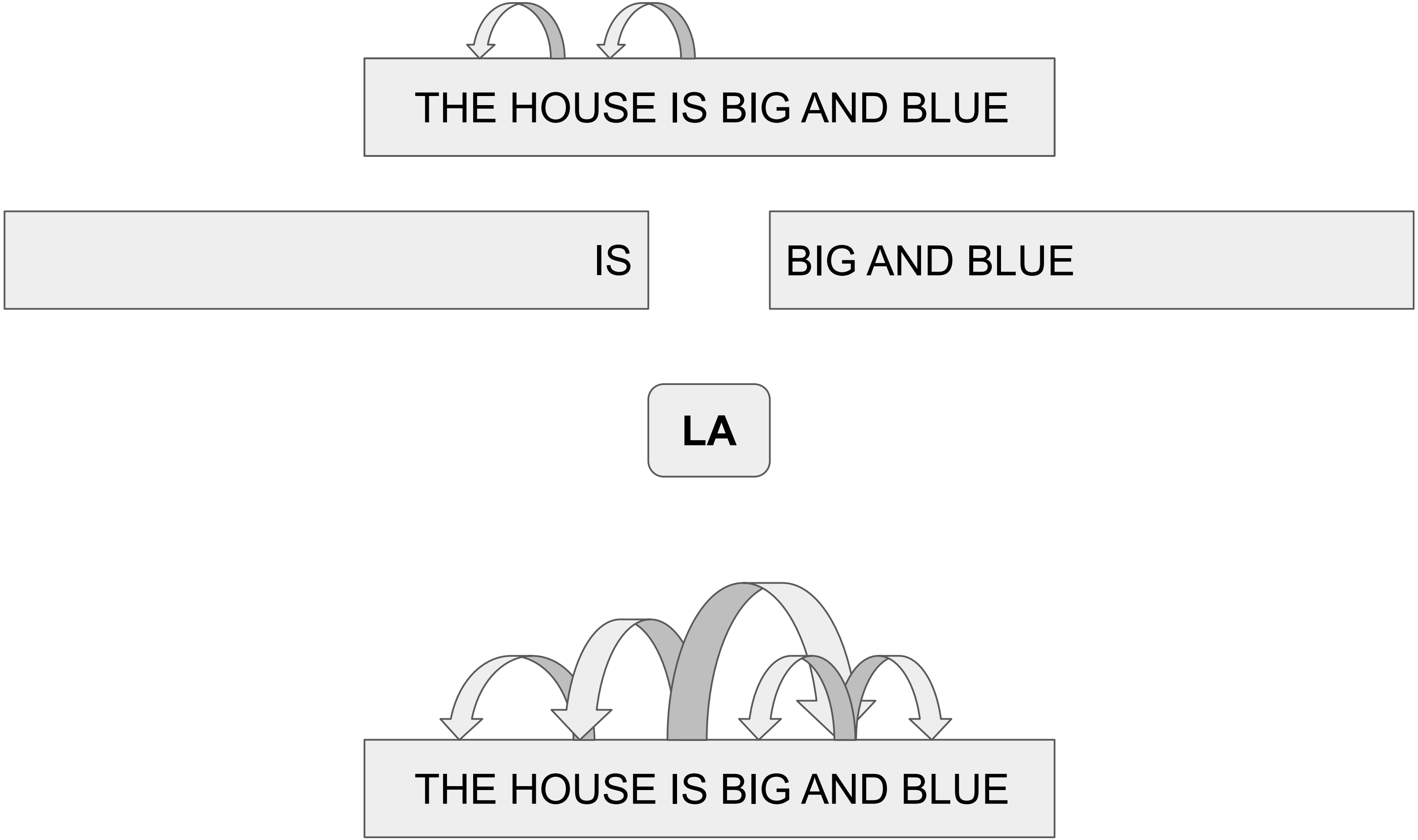


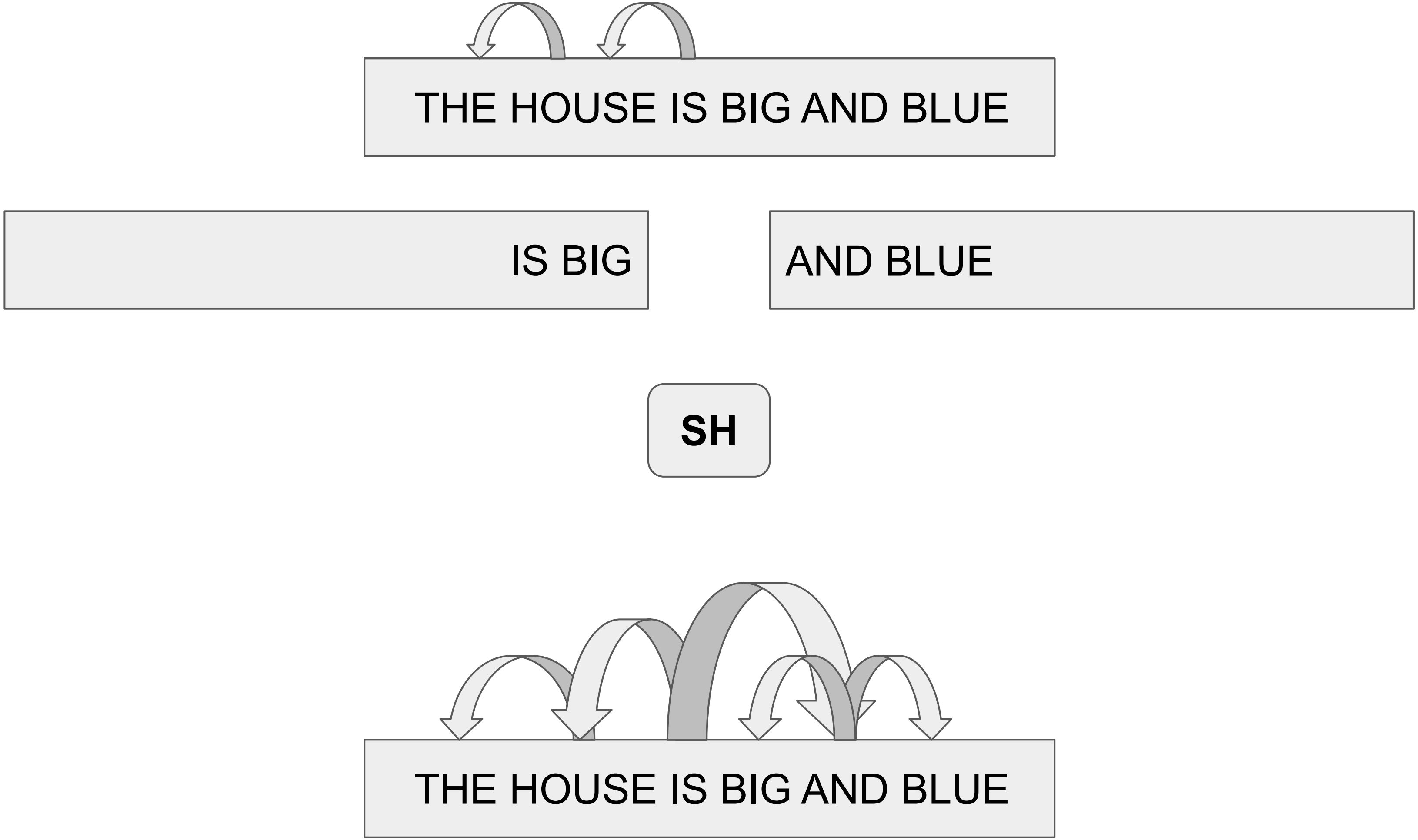
THE HOUSE IS BIG AND BLUE

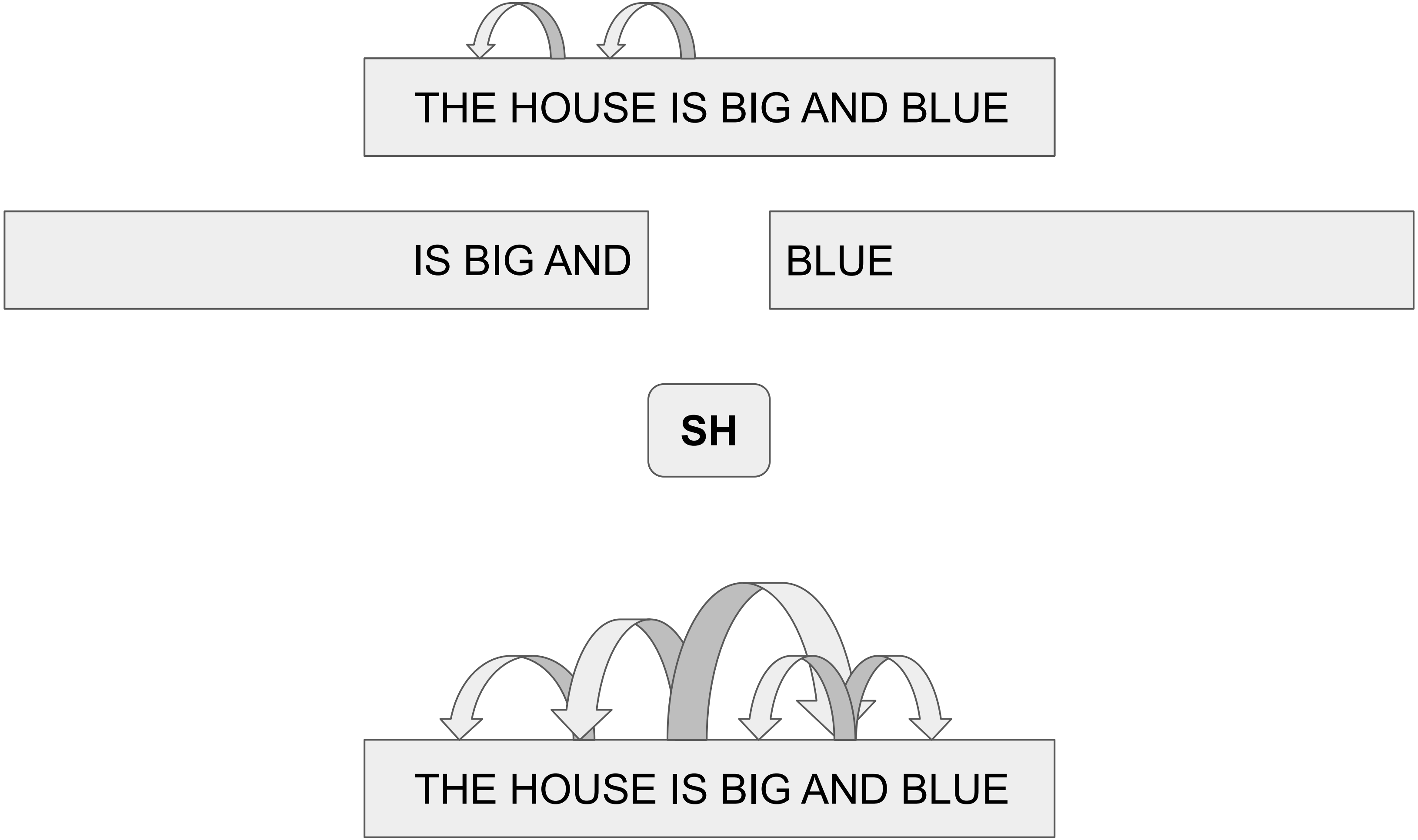

THE HOUSE IS BIG AND BLUE
THE HOUSE
IS BIG AND BLUE
SH
THE HOUSE IS BIG AND BLUE

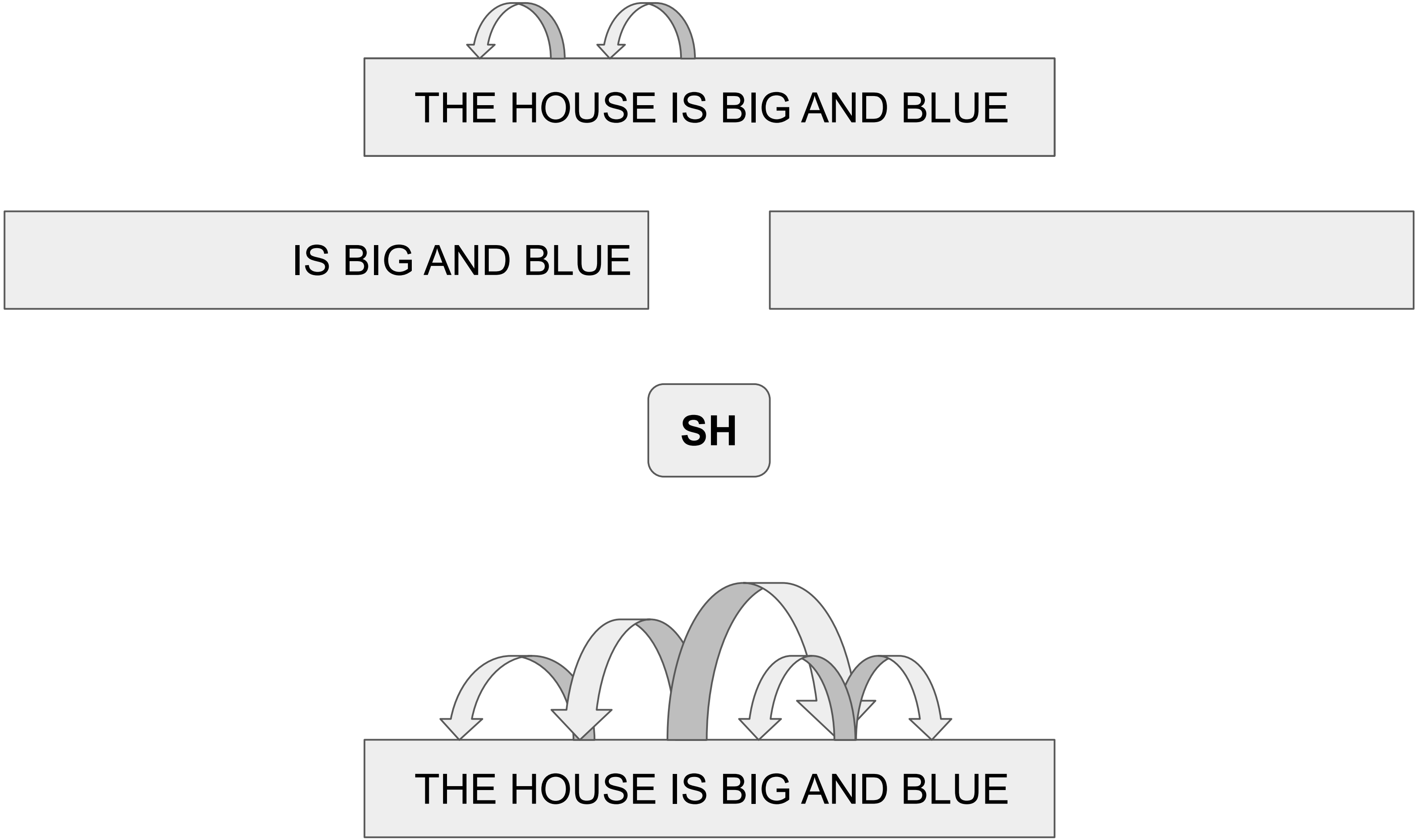

THE HOUSE IS BIG AND BLUE
HOUSE
IS BIG AND BLUE
LA
THE HOUSE IS BIG AND BLUE

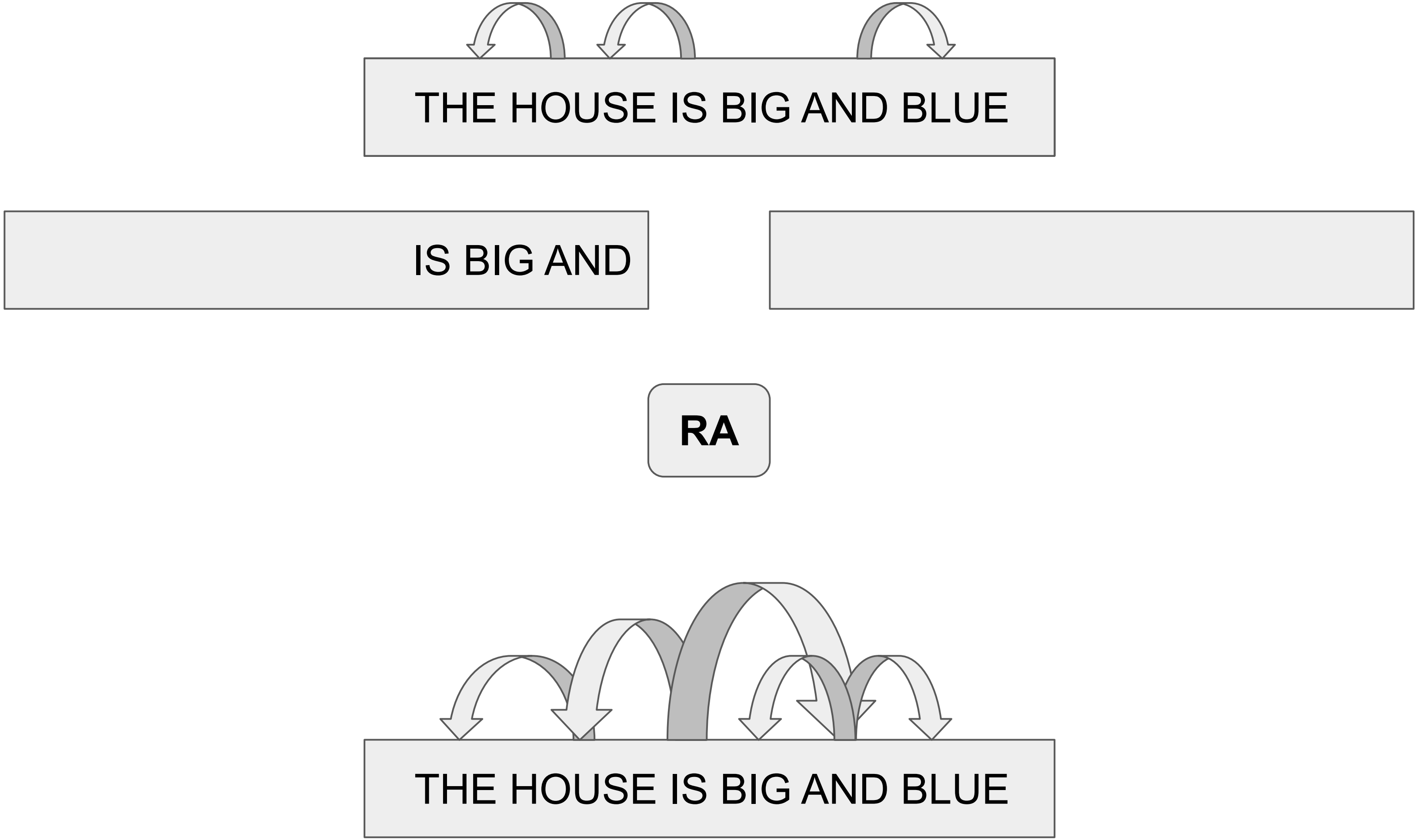

THE HOUSE IS BIG AND BLUE
HOUSE IS
BIG AND BLUE
SH
THE HOUSE IS BIG AND BLUE

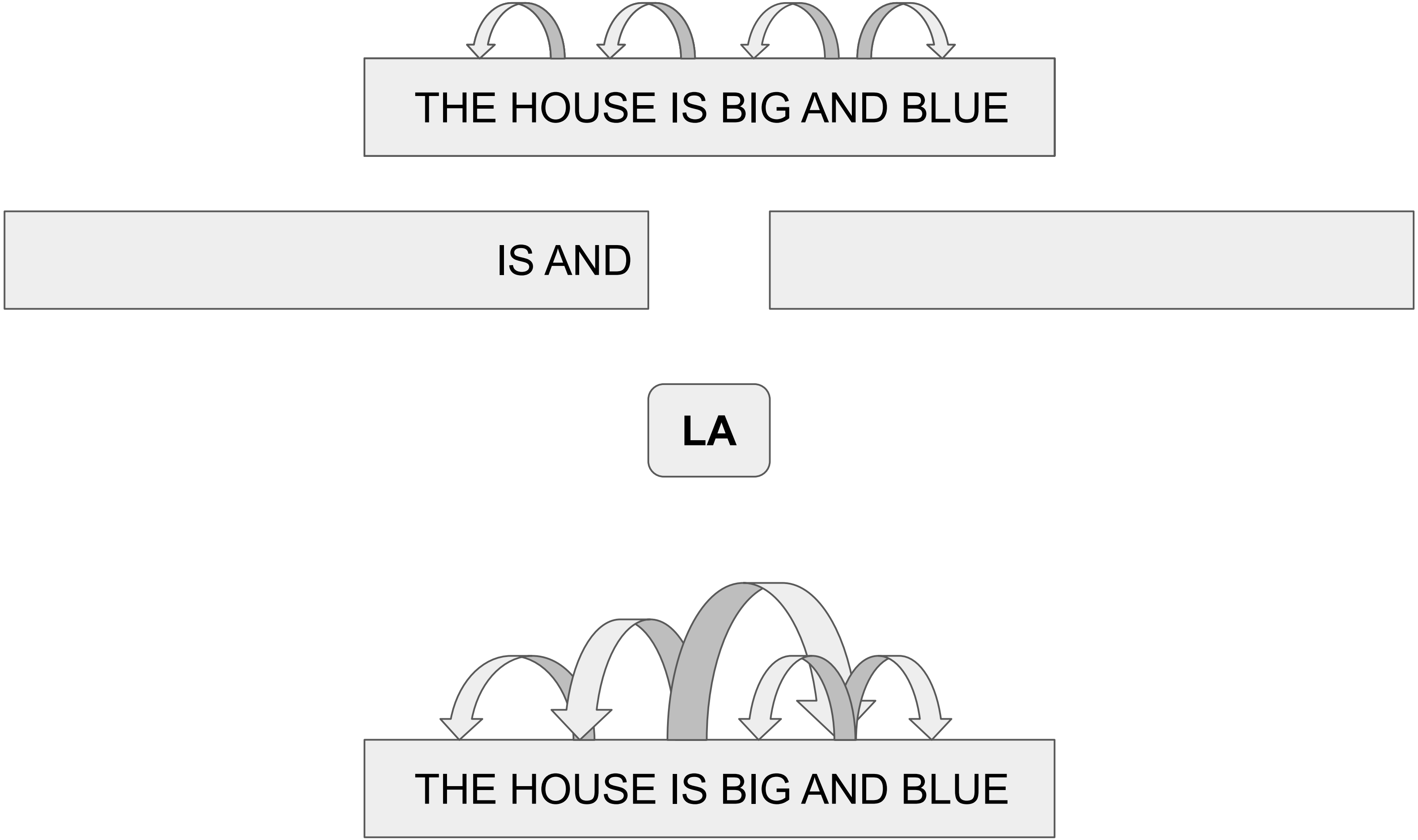

THE HOUSE IS BIG AND BLUE
IS
BIG AND BLUE
LA
THE HOUSE IS BIG AND BLUE

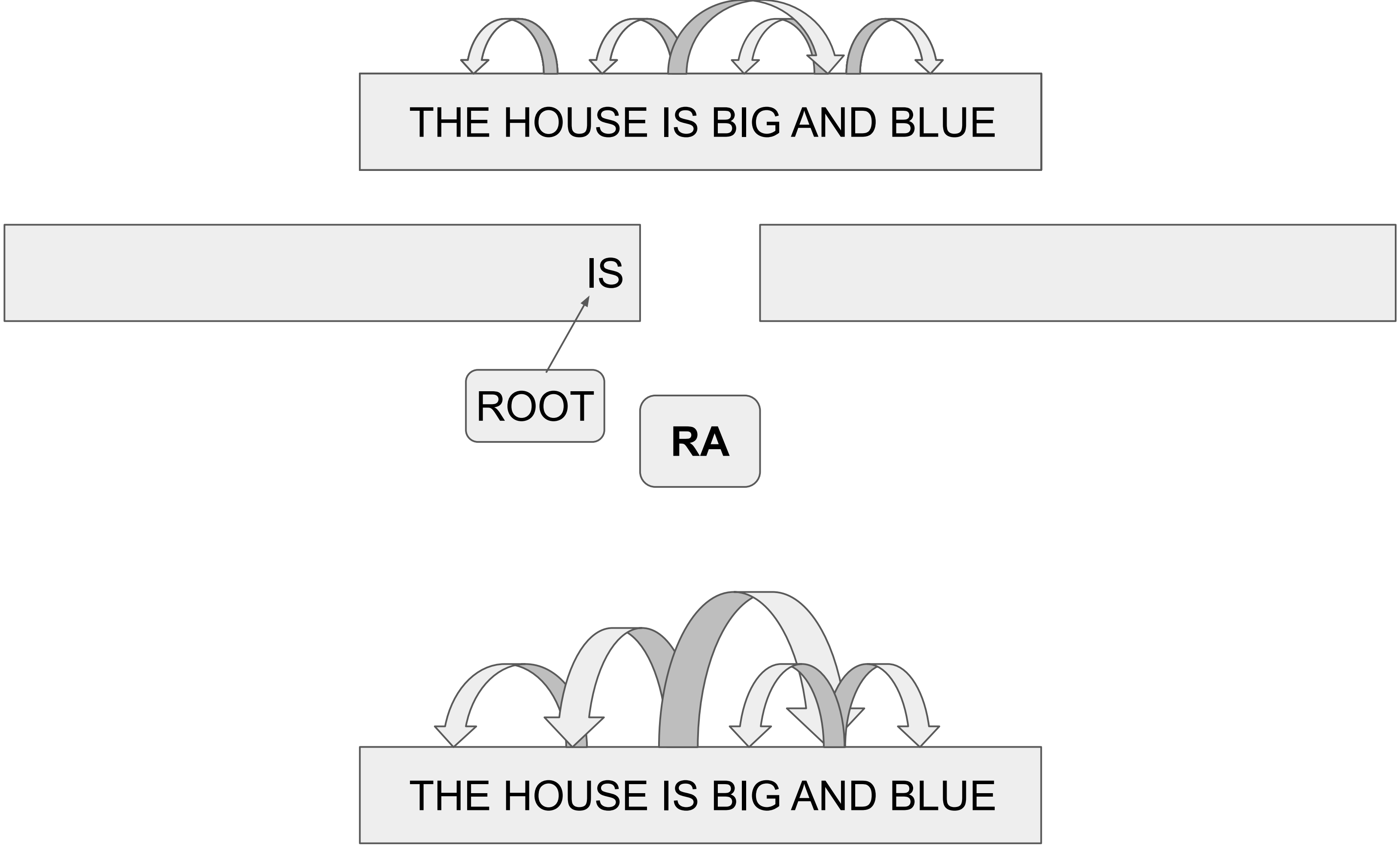

THE HOUSE IS BIG AND BLUE
IS BIG
AND BLUE
SH
THE HOUSE IS BIG AND BLUE


THE HOUSE IS BIG AND BLUE
IS BIG AND
BLUE
SH
THE HOUSE IS BIG AND BLUE


THE HOUSE IS BIG AND BLUE
IS BIG AND BLUE
SH
THE HOUSE IS BIG AND BLUE


THE HOUSE IS BIG AND BLUE
IS BIG AND
RA
THE HOUSE IS BIG AND BLUE


THE HOUSE IS BIG AND BLUE
IS AND
LA
THE HOUSE IS BIG AND BLUE

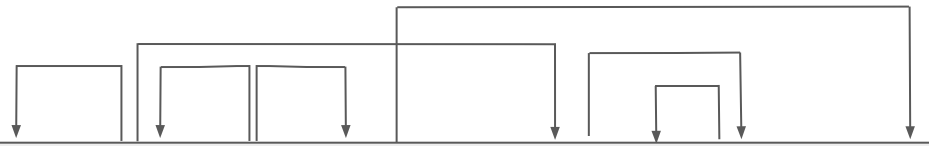

THE HOUSE IS BIG AND BLUE
IS
ROOT
RA
THE HOUSE IS BIG AND BLUE

Training

- Perform Left-Arc transition if it creates a gold arc, and all arcs from the "popped" word are completed.
- Else perform Right-Arc if the same restrictions as above are met.
- Else perform shift transition.

A HEARING IS SCHEDULED ON THE ISSUE TODAY

A HEARING IS SCHEDULED ON THE ISSUE TODAY



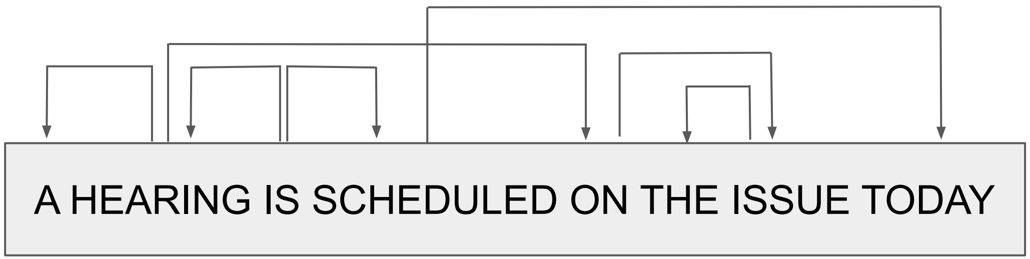
A HEARING IS SCHEDULED ON THE ISSUE TODAY

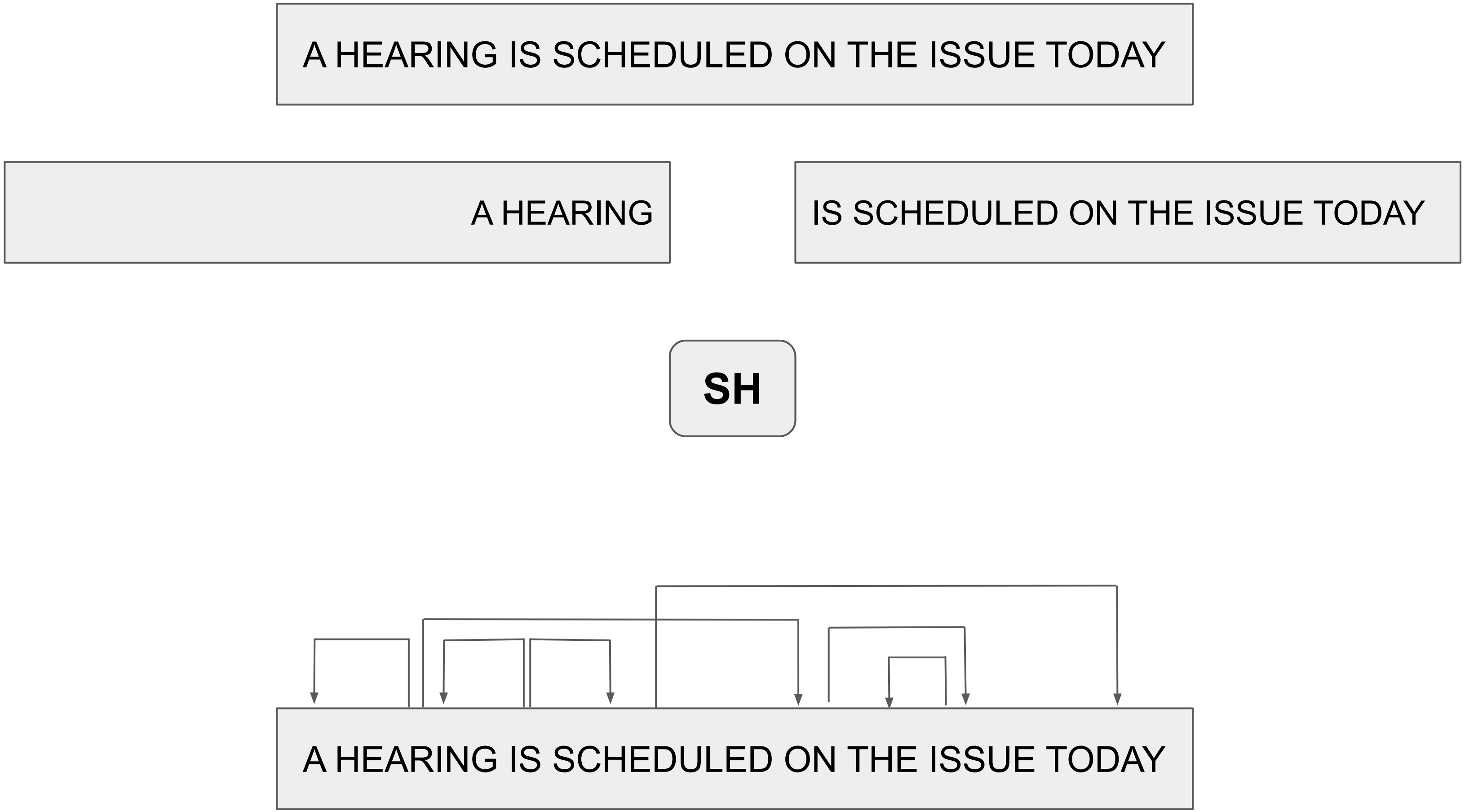
A HEARING IS SCHEDULED ON THE ISSUE TODAY

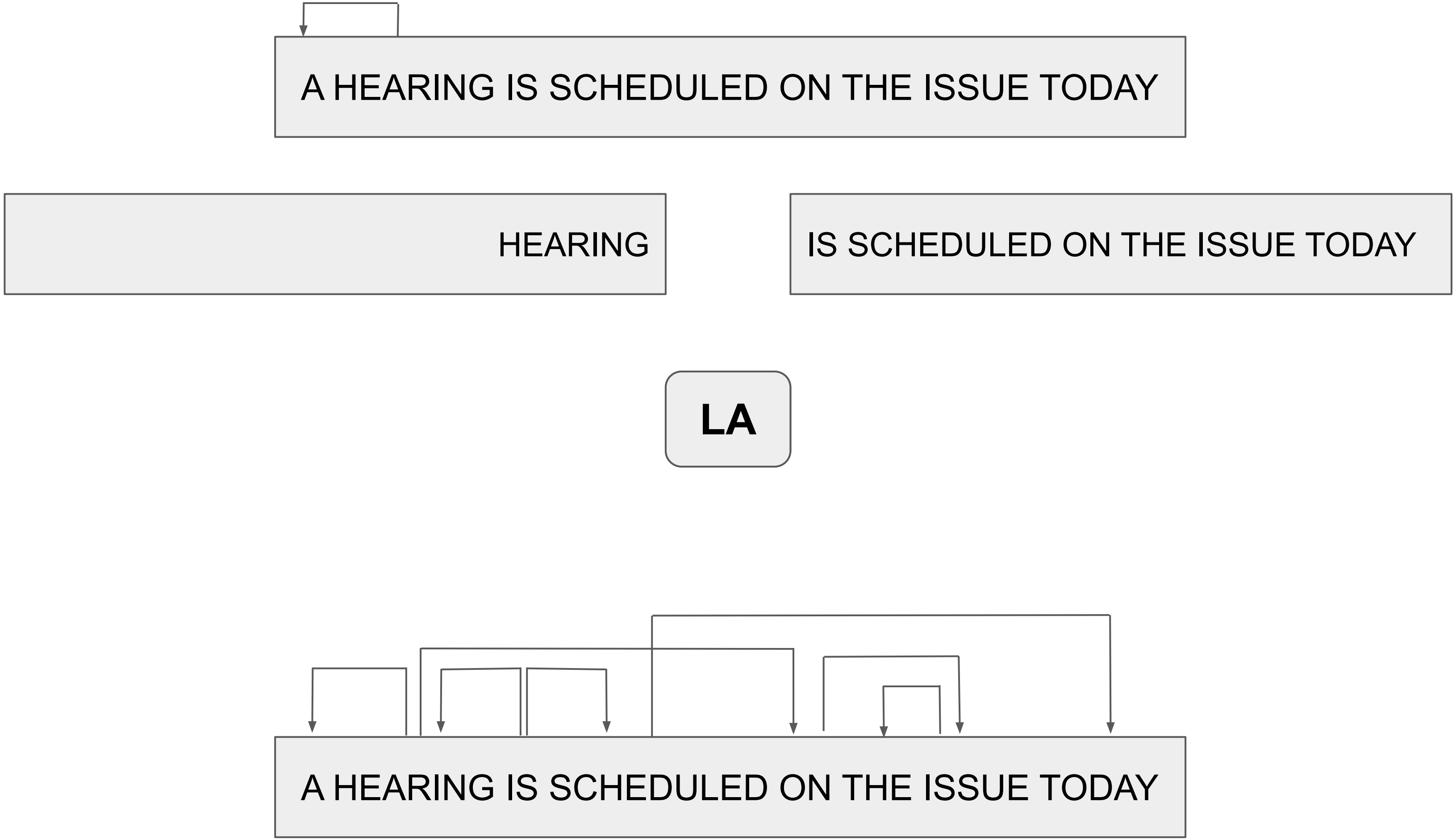
A

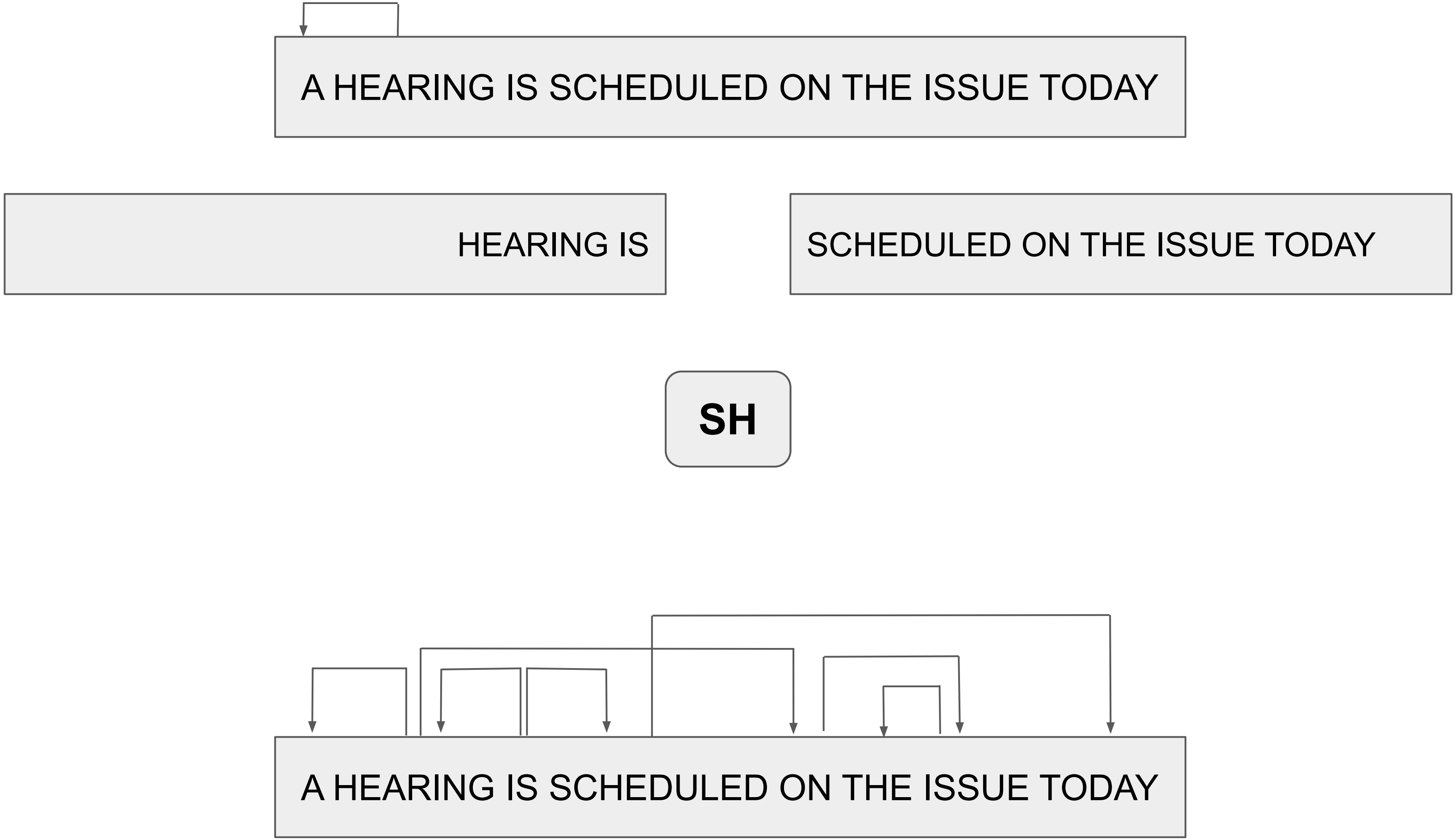
HEARING IS SCHEDULED ON THE ISSUE TODAY

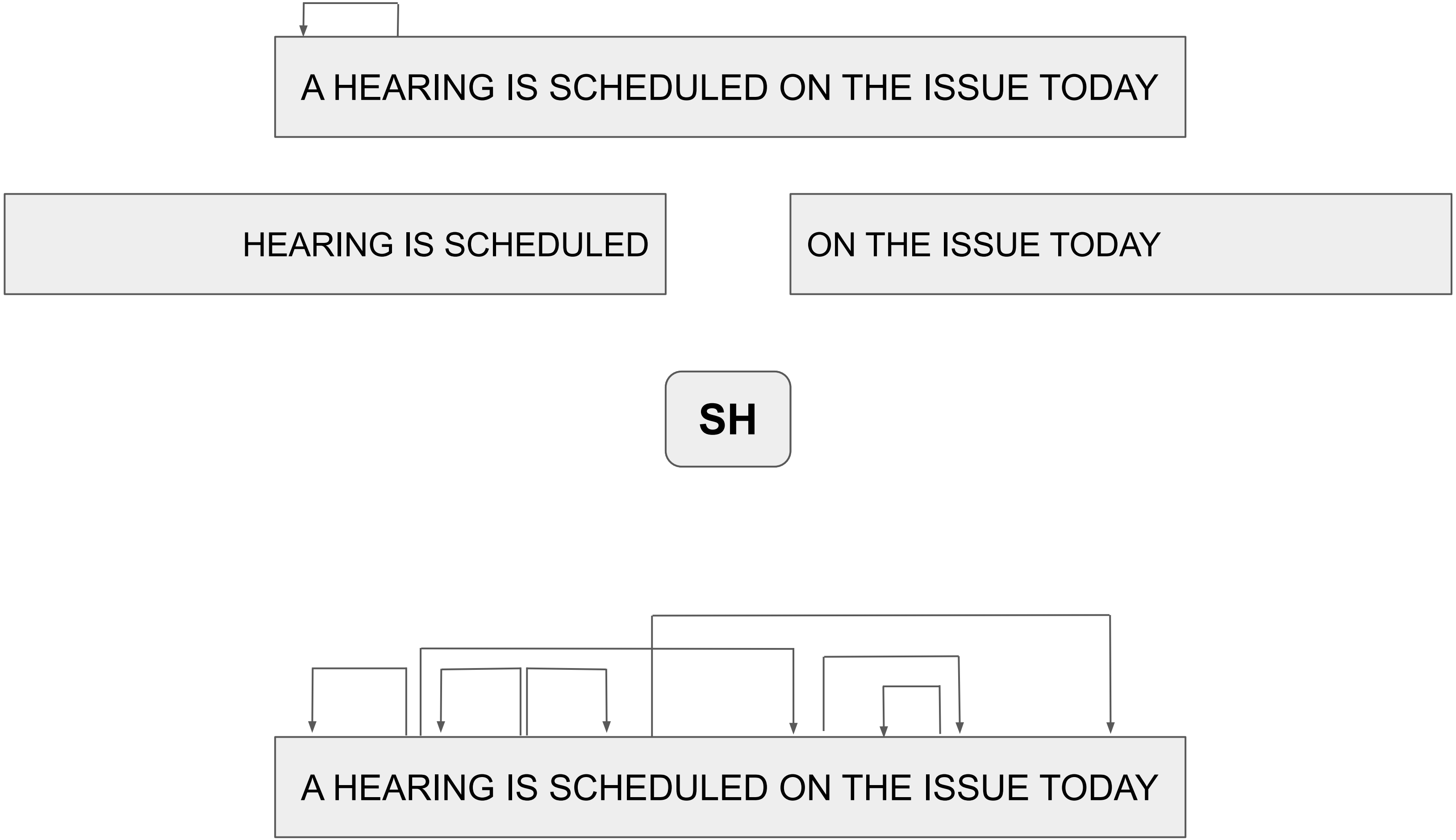
SH

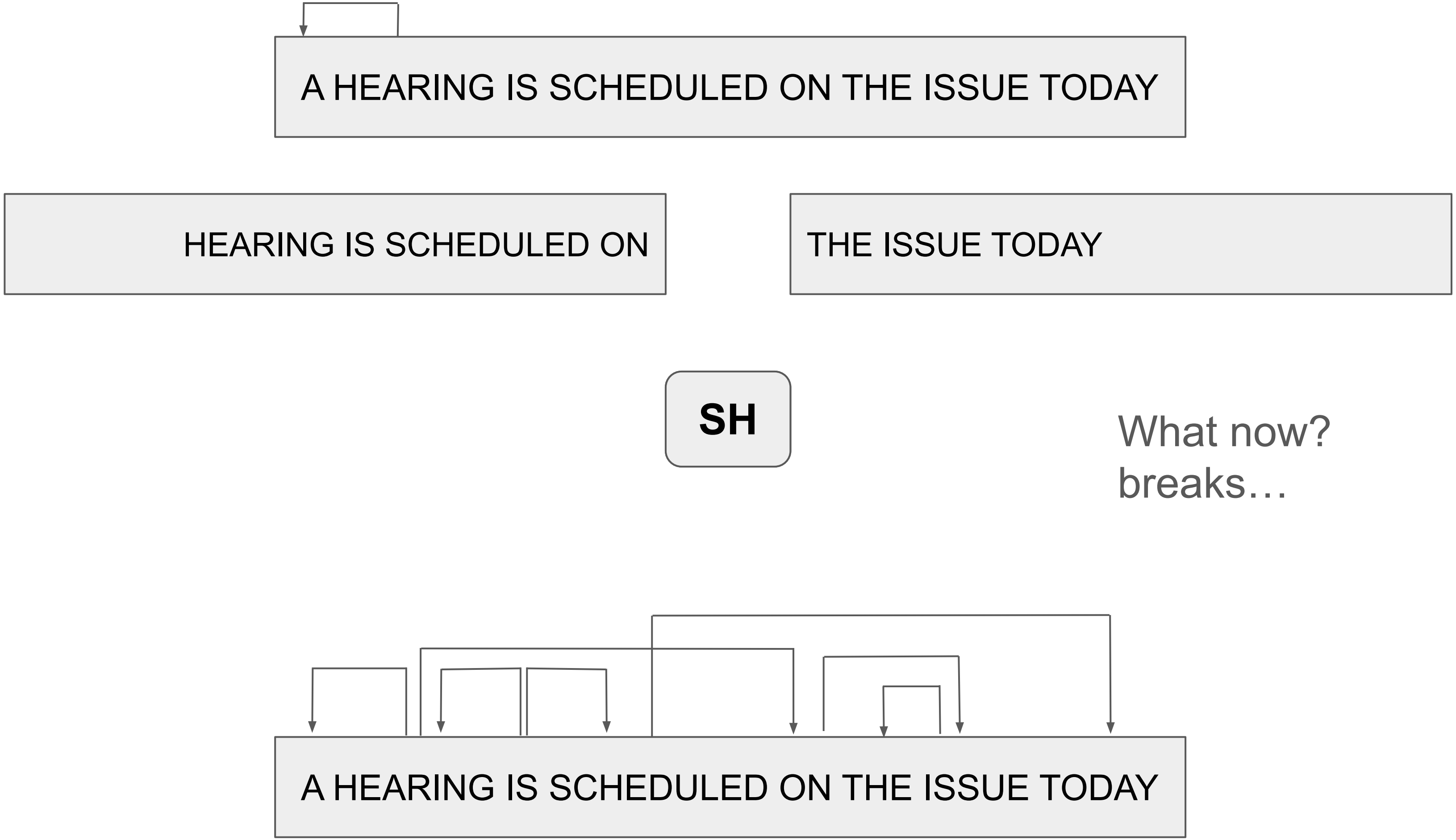


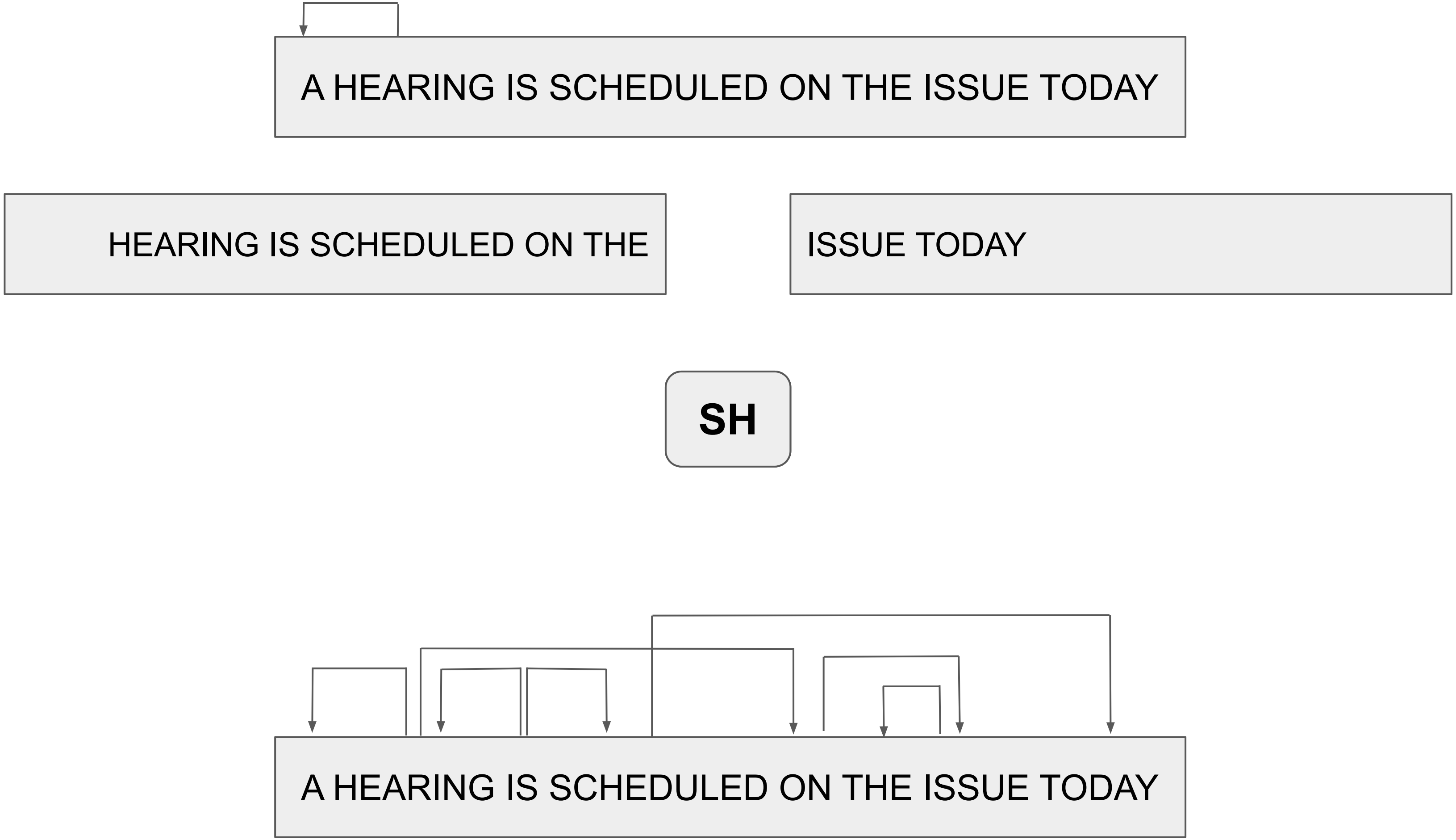

A HEARING IS SCHEDULED ON THE ISSUE TODAY
A HEARING
IS SCHEDULED ON THE ISSUE TODAY
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY

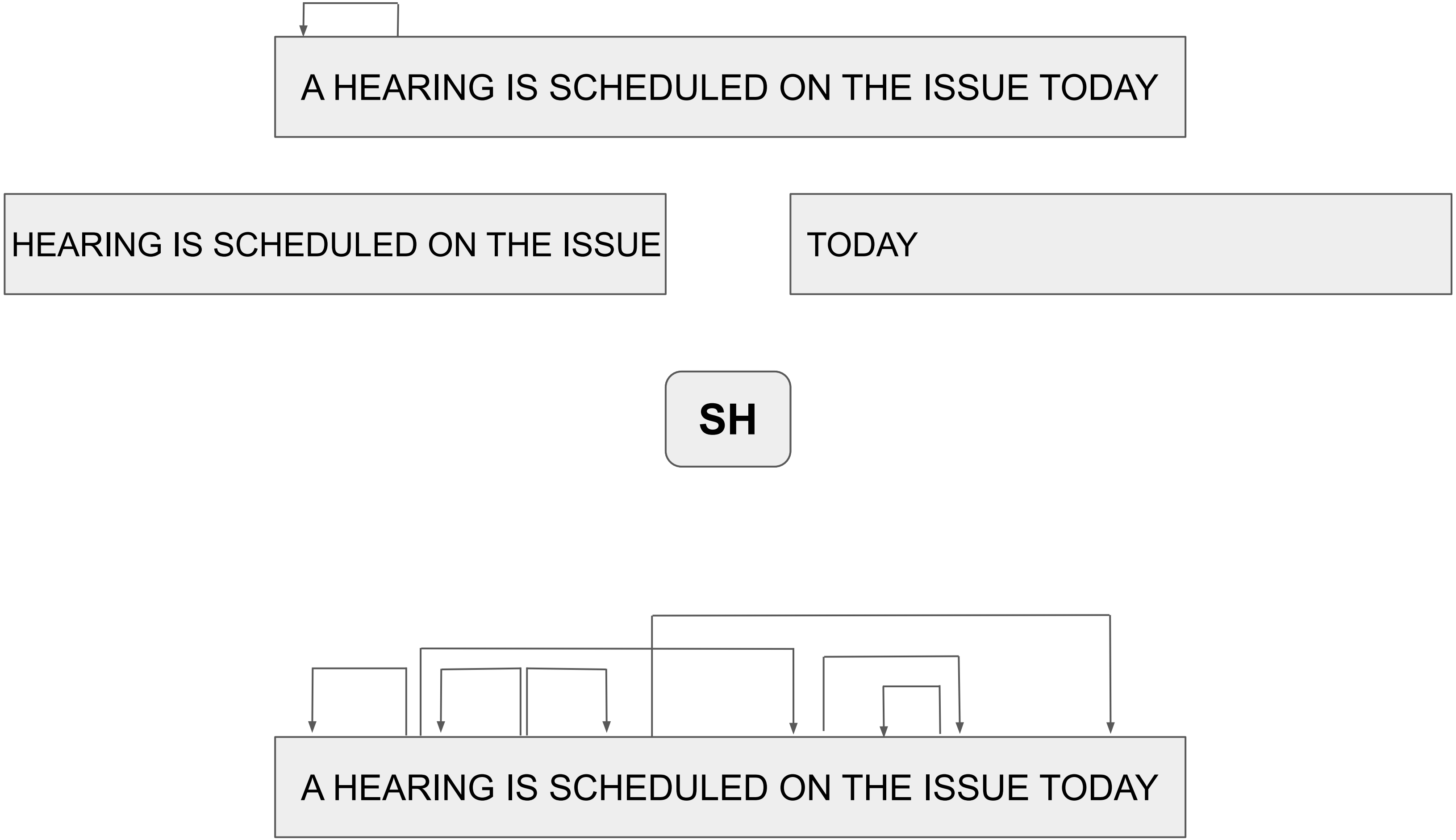

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING
IS SCHEDULED ON THE ISSUE TODAY
LA
A HEARING IS SCHEDULED ON THE ISSUE TODAY

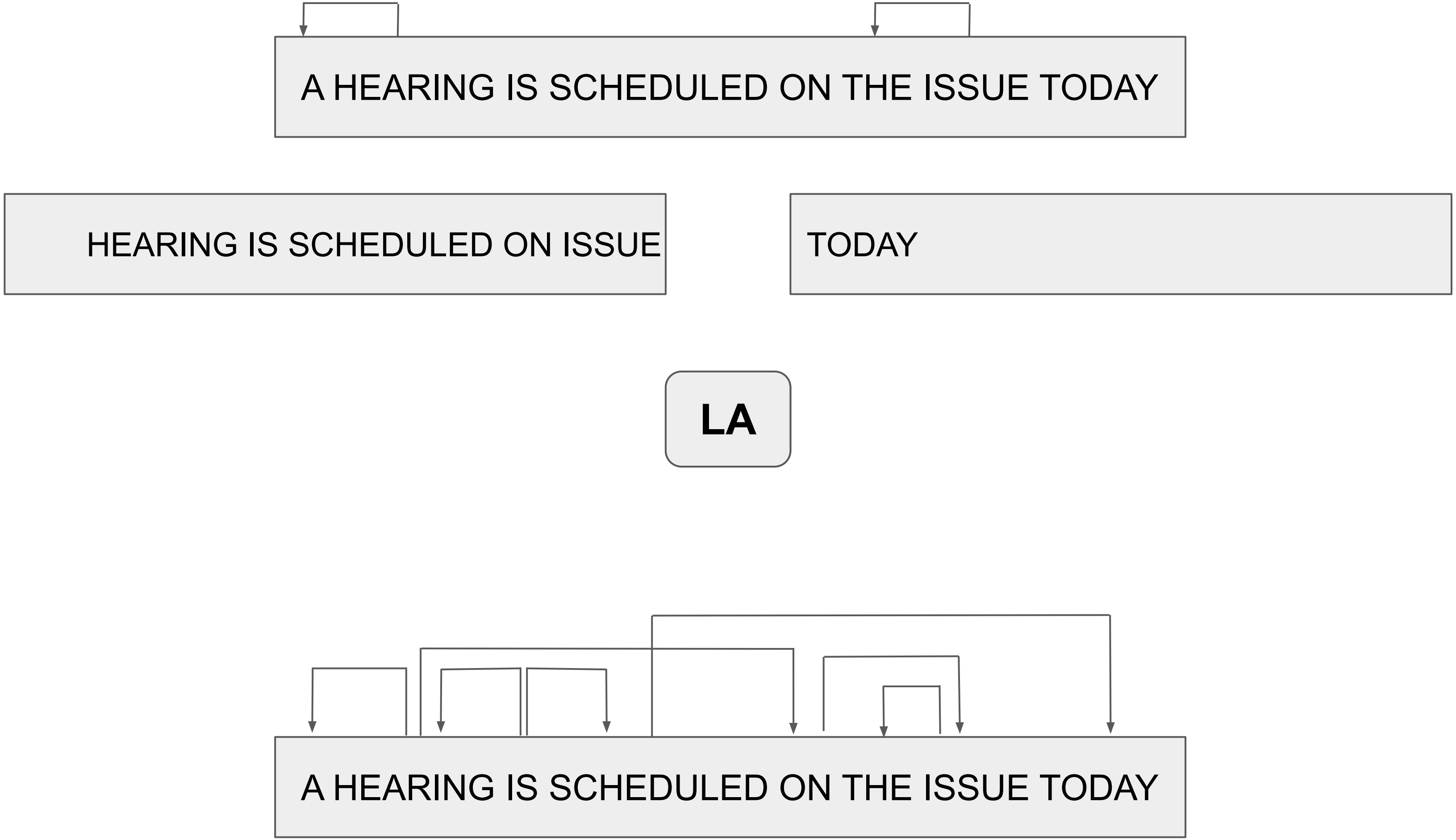

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS
SCHEDULED ON THE ISSUE TODAY
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY

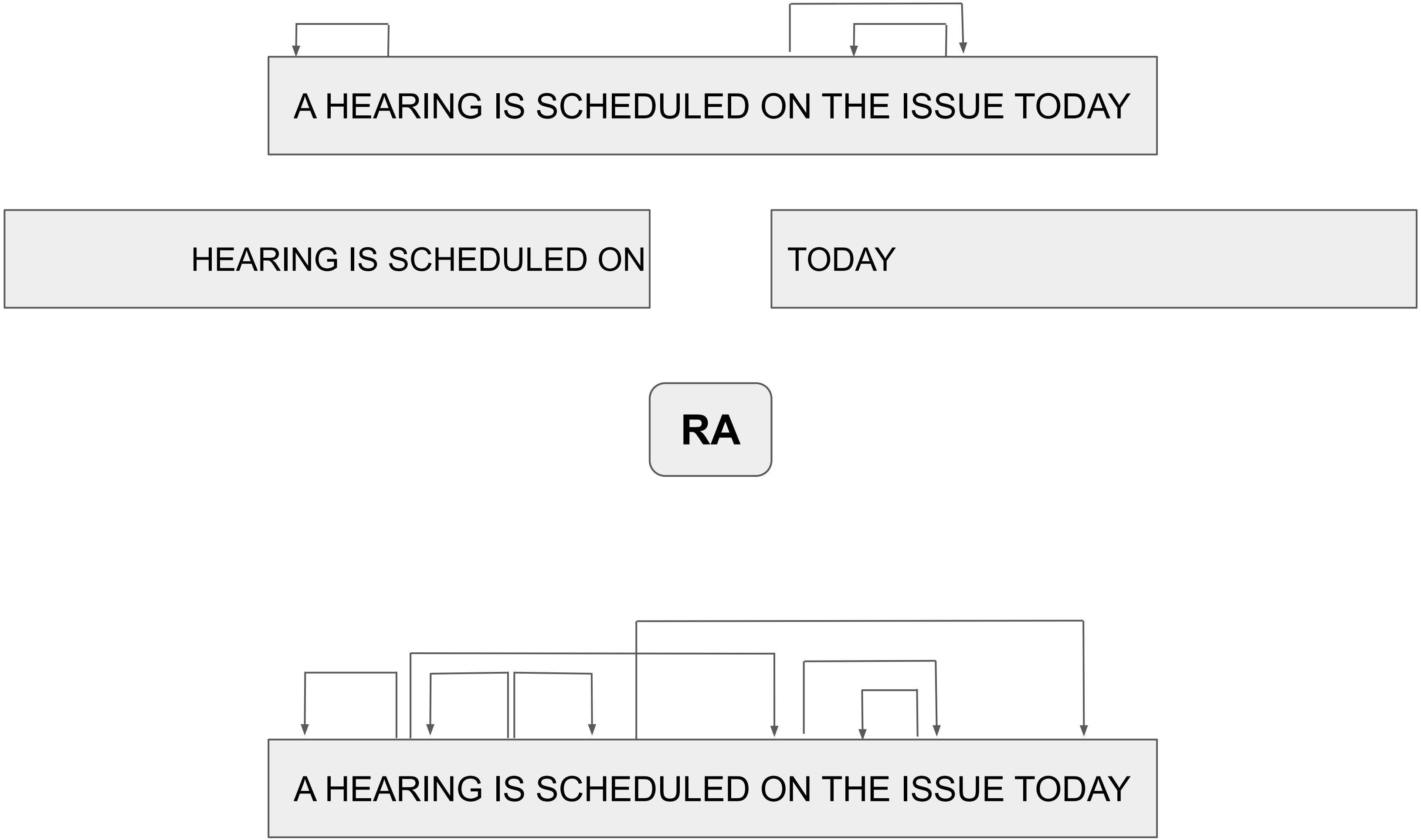

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED
ON THE ISSUE TODAY
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY

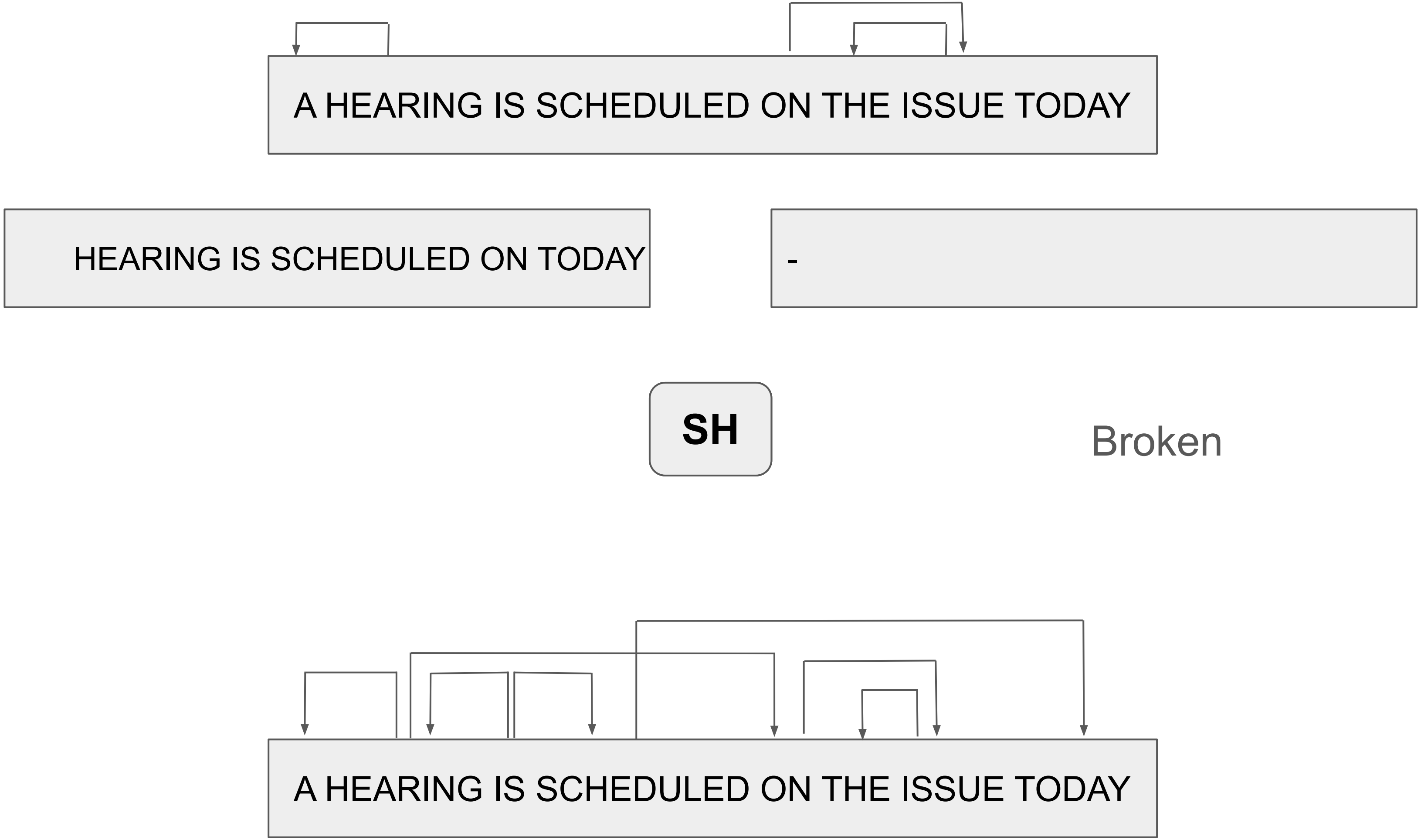

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON
THE ISSUE TODAY
SH
What now?
breaks…
A HEARING IS SCHEDULED ON THE ISSUE TODAY


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON THE
ISSUE TODAY
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON THE ISSUE
TODAY
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON ISSUE
TODAY
LA
A HEARING IS SCHEDULED ON THE ISSUE TODAY


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON
TODAY
RA
A HEARING IS SCHEDULED ON THE ISSUE TODAY

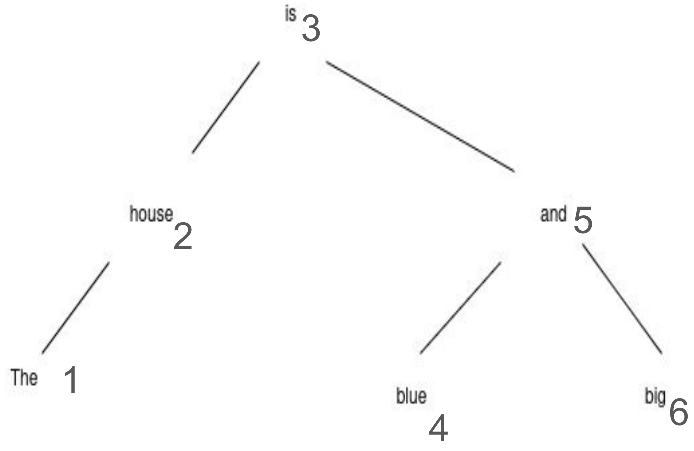

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON TODAY
-
SH
Broken
A HEARING IS SCHEDULED ON THE ISSUE TODAY

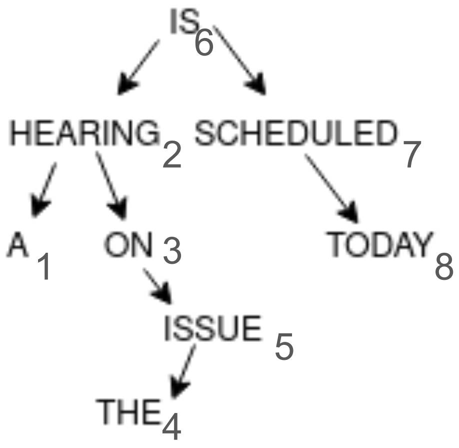
Introduction of additional transition, SWAP

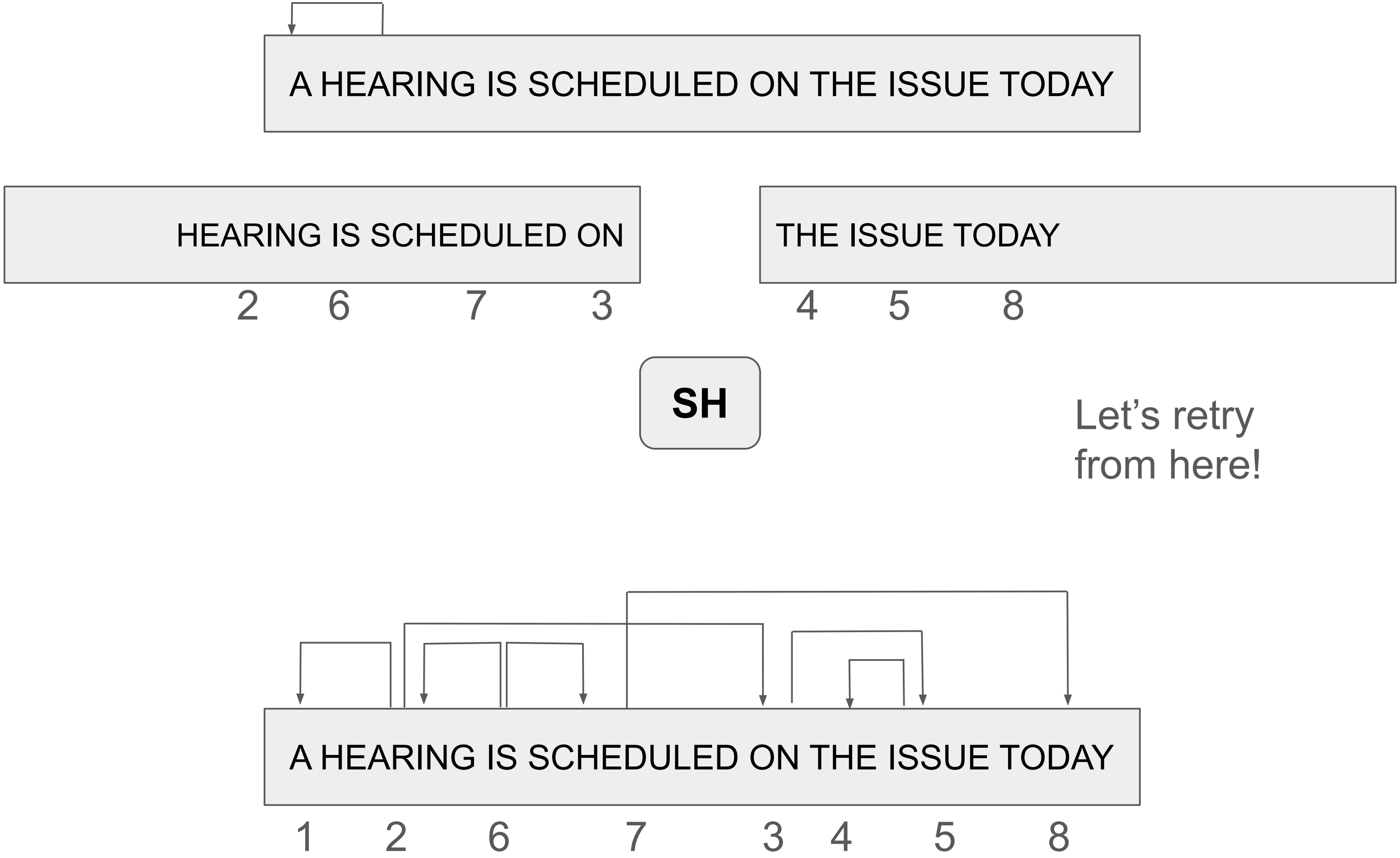
SWAP moves the second-topmost word on the stack back to the buffer.

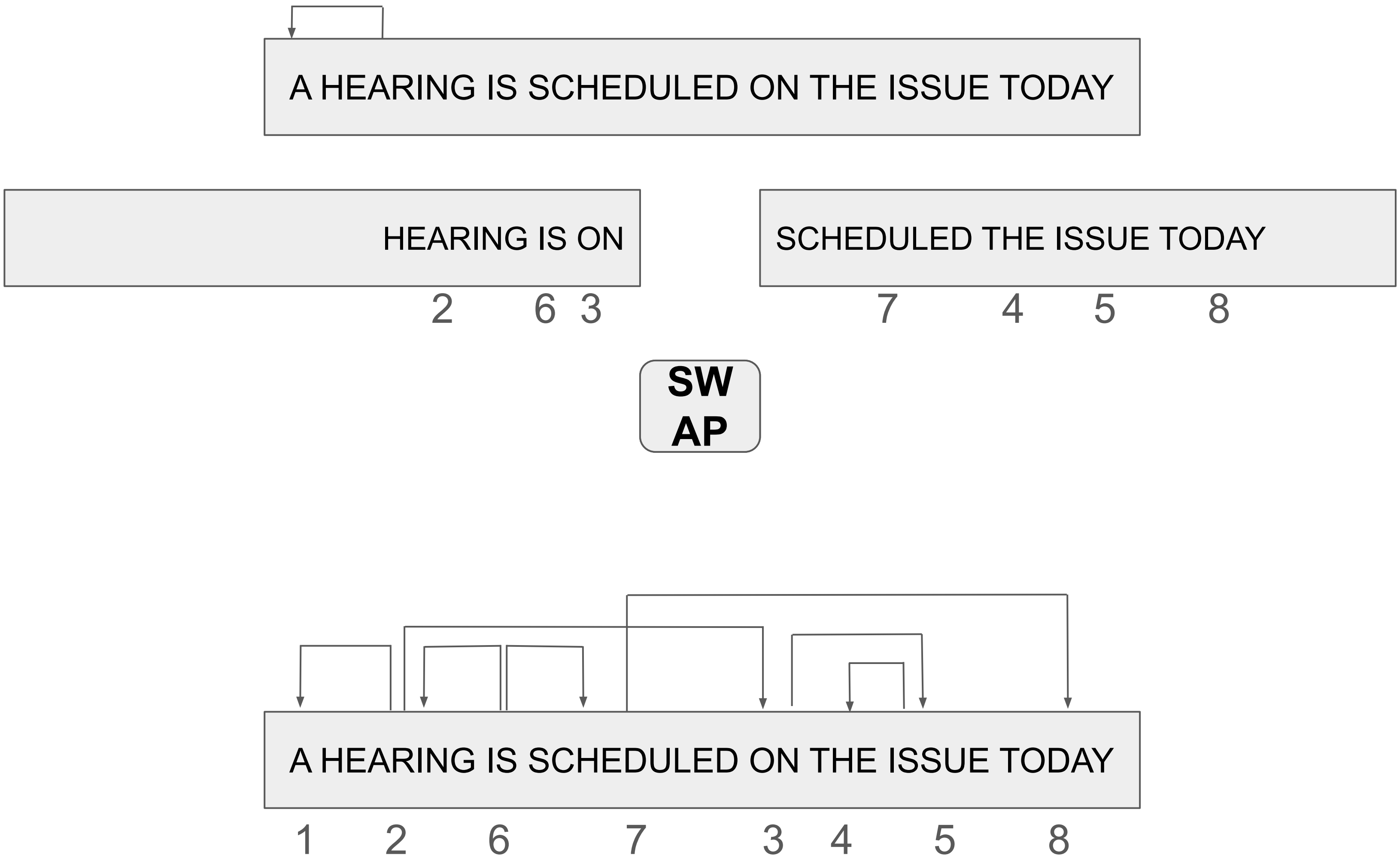
Instead of shift if the two topmost words on the stack aren’t in their “projective order”

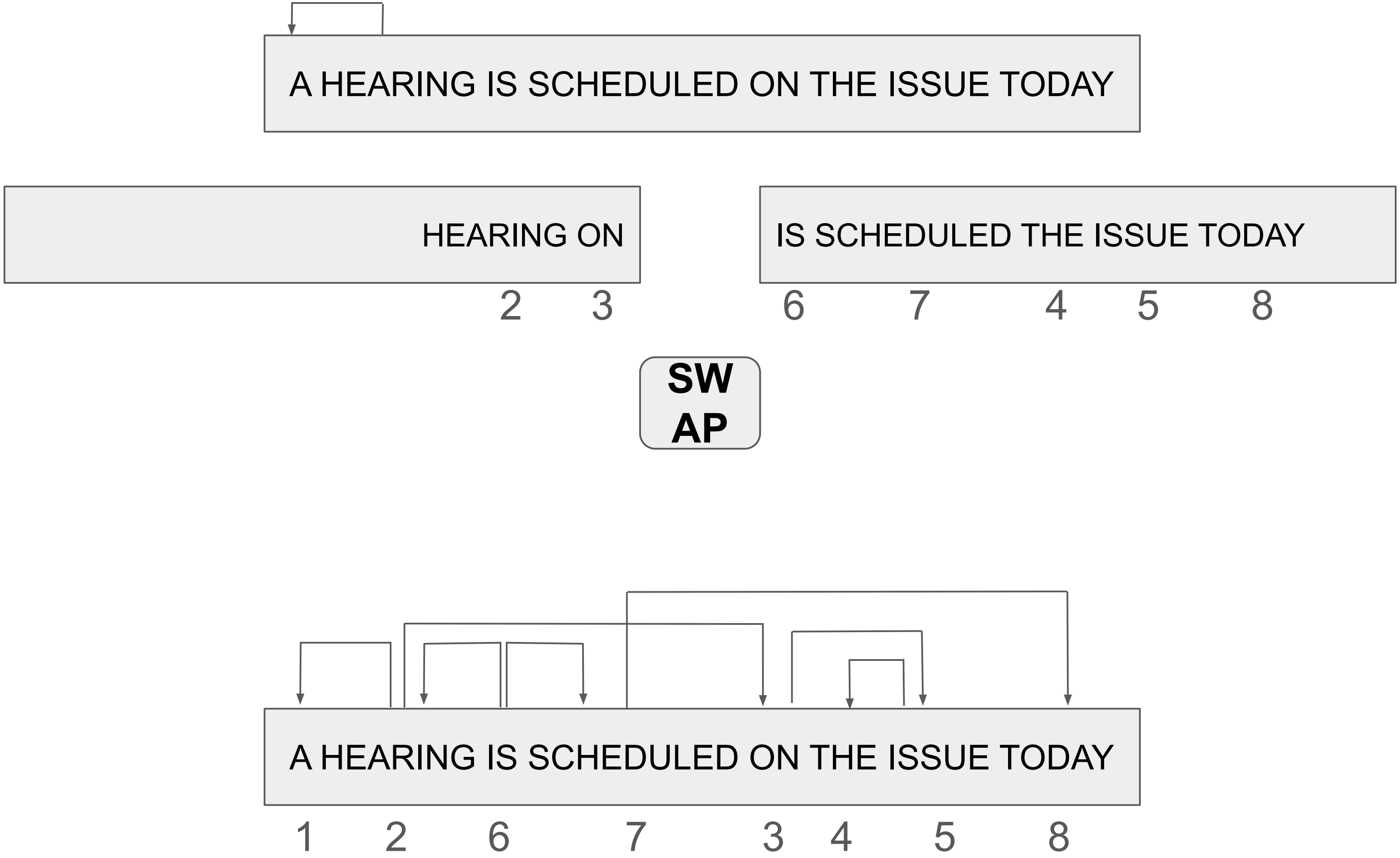
Introduced in the article “Non-Projective Dependency Parsing in Expected Linear Time”, Nivre, ACL-IJCNLP 2009.

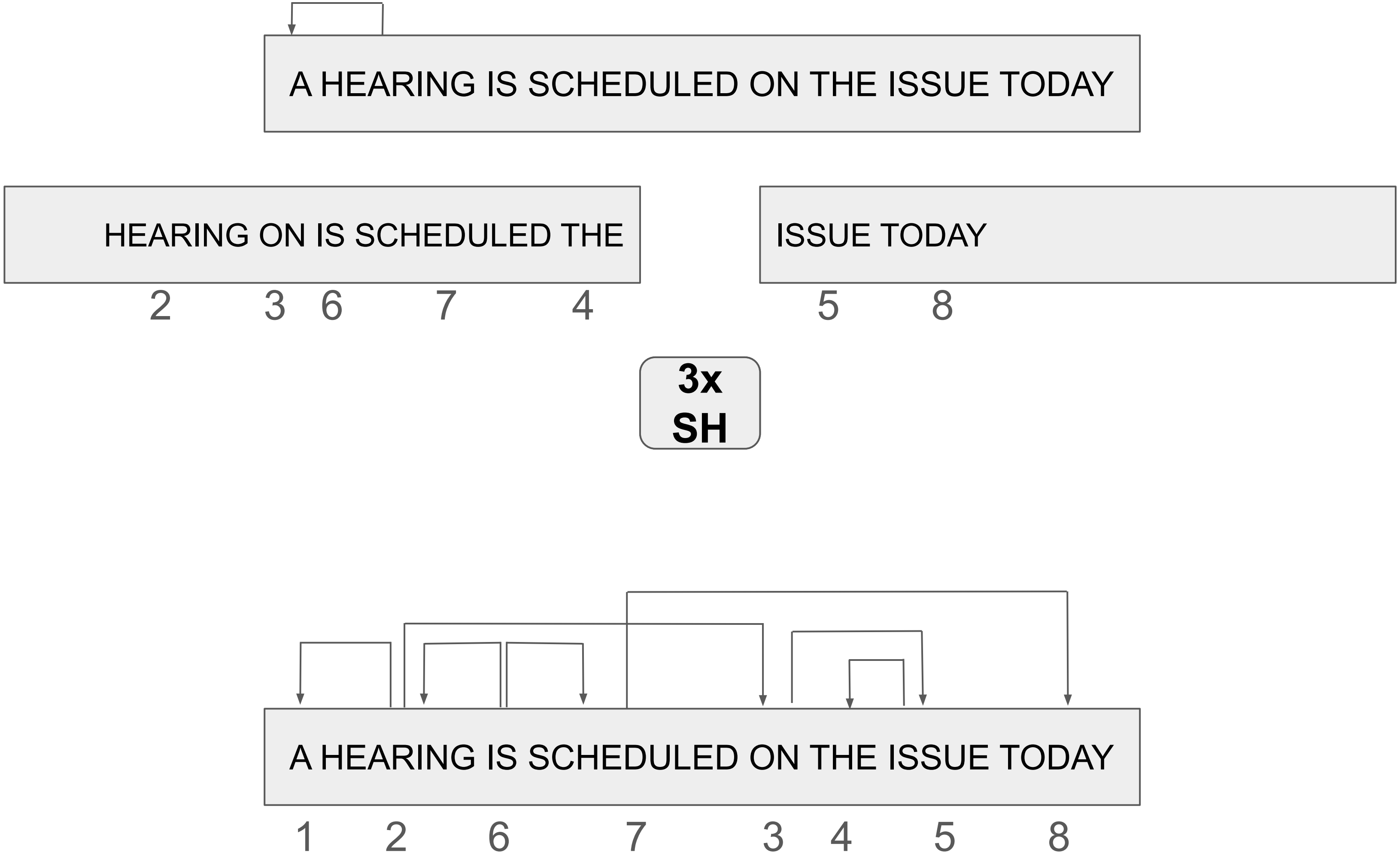


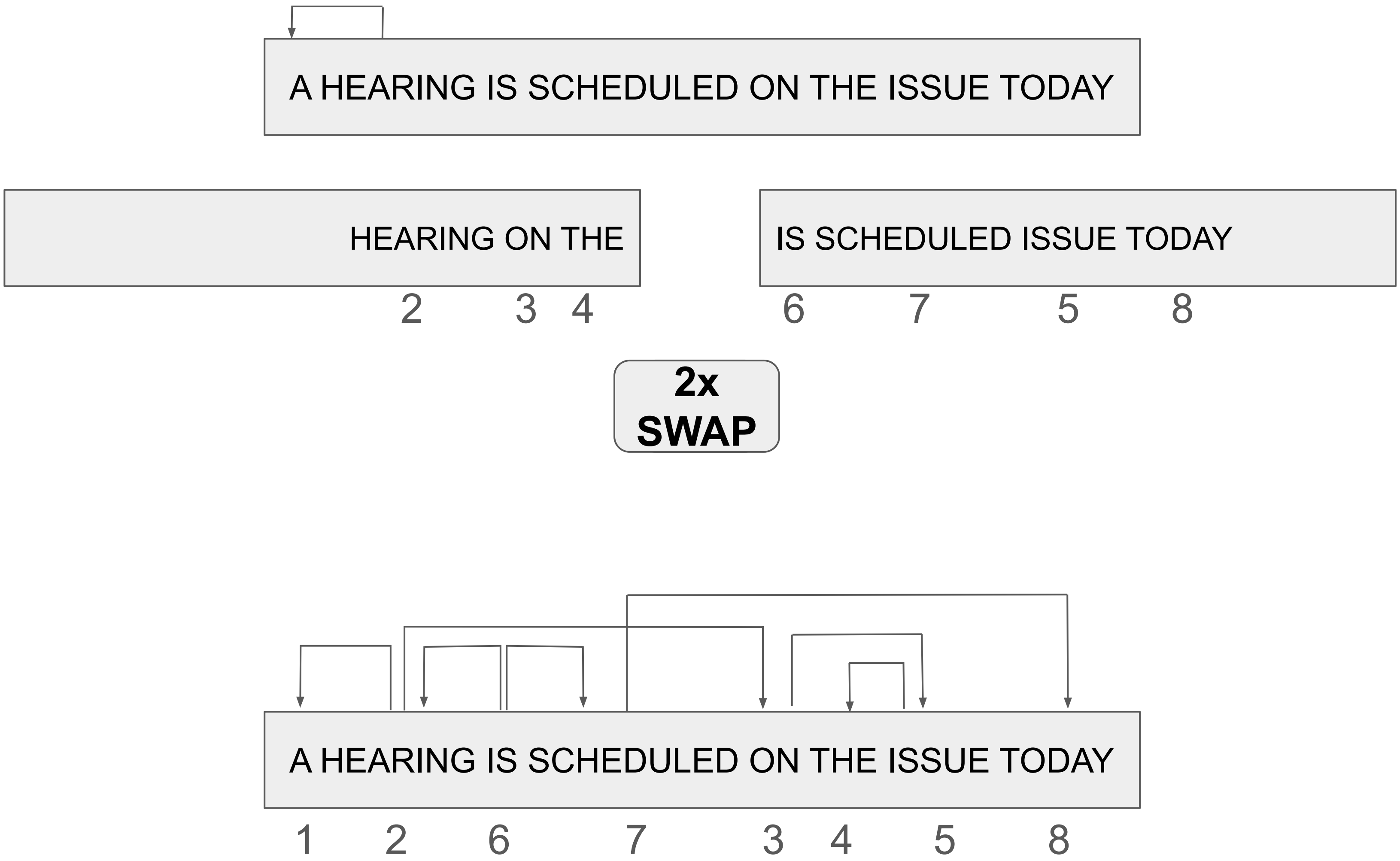


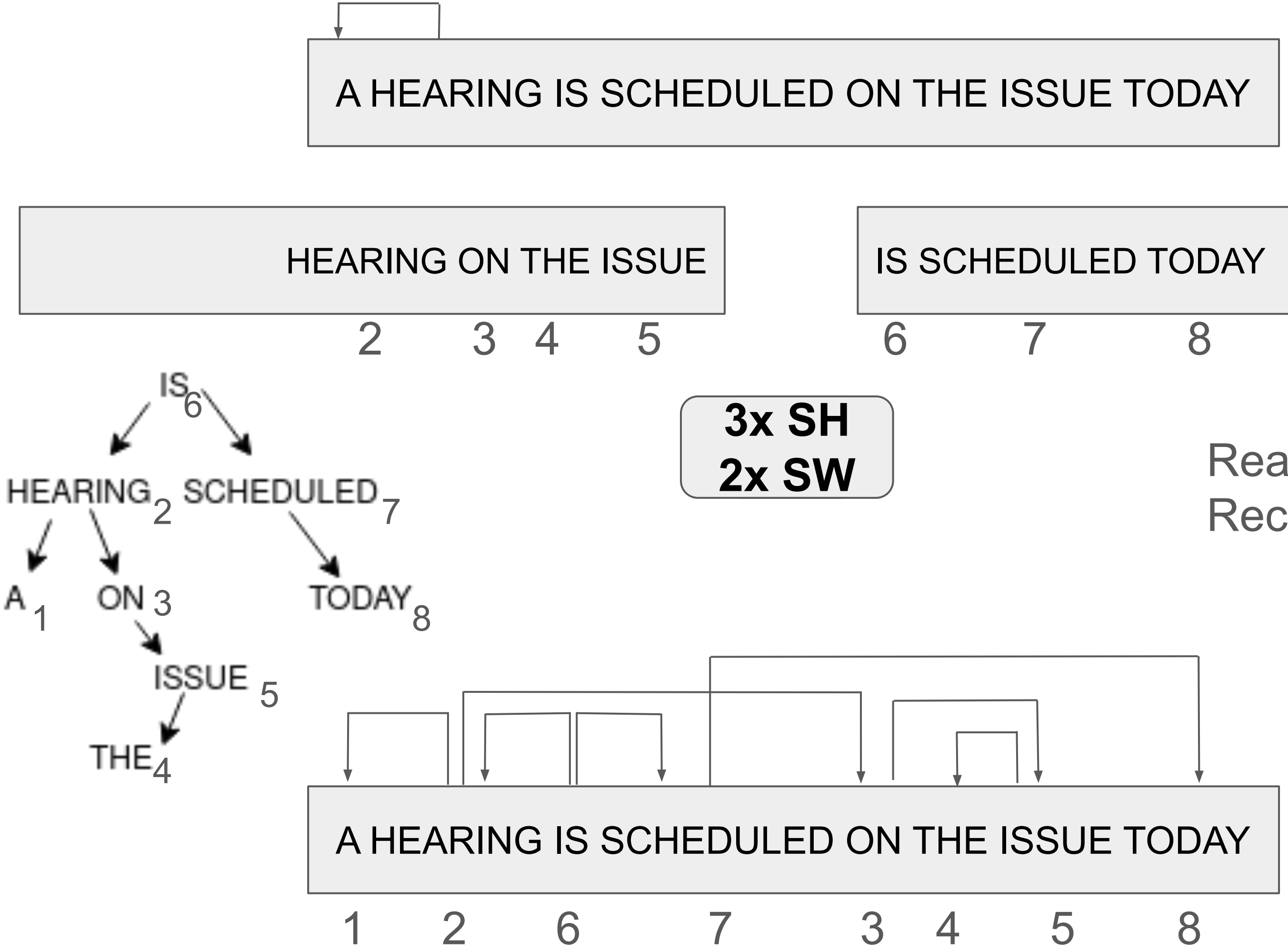

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS SCHEDULED ON
2 6 7 3
THE ISSUE TODAY
4 5 8
SH
Let's retry
from here!
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS ON
SCHEDULED THE ISSUE TODAY
2 6 3
7 4 5 8
SWAP
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8

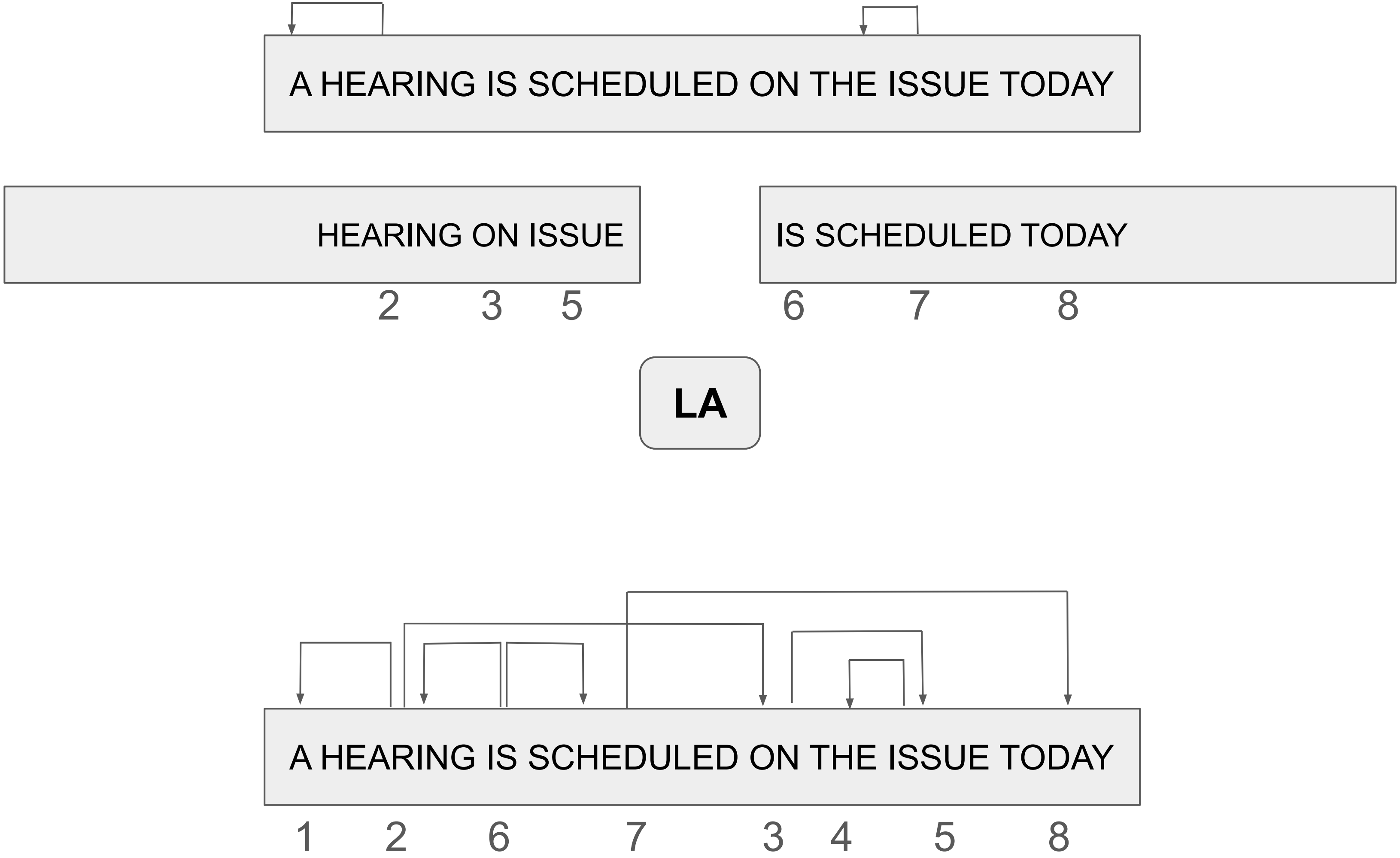

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING ON
2 3
IS SCHEDULED THE ISSUE TODAY
6 7 4 5 8
SWAP
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8

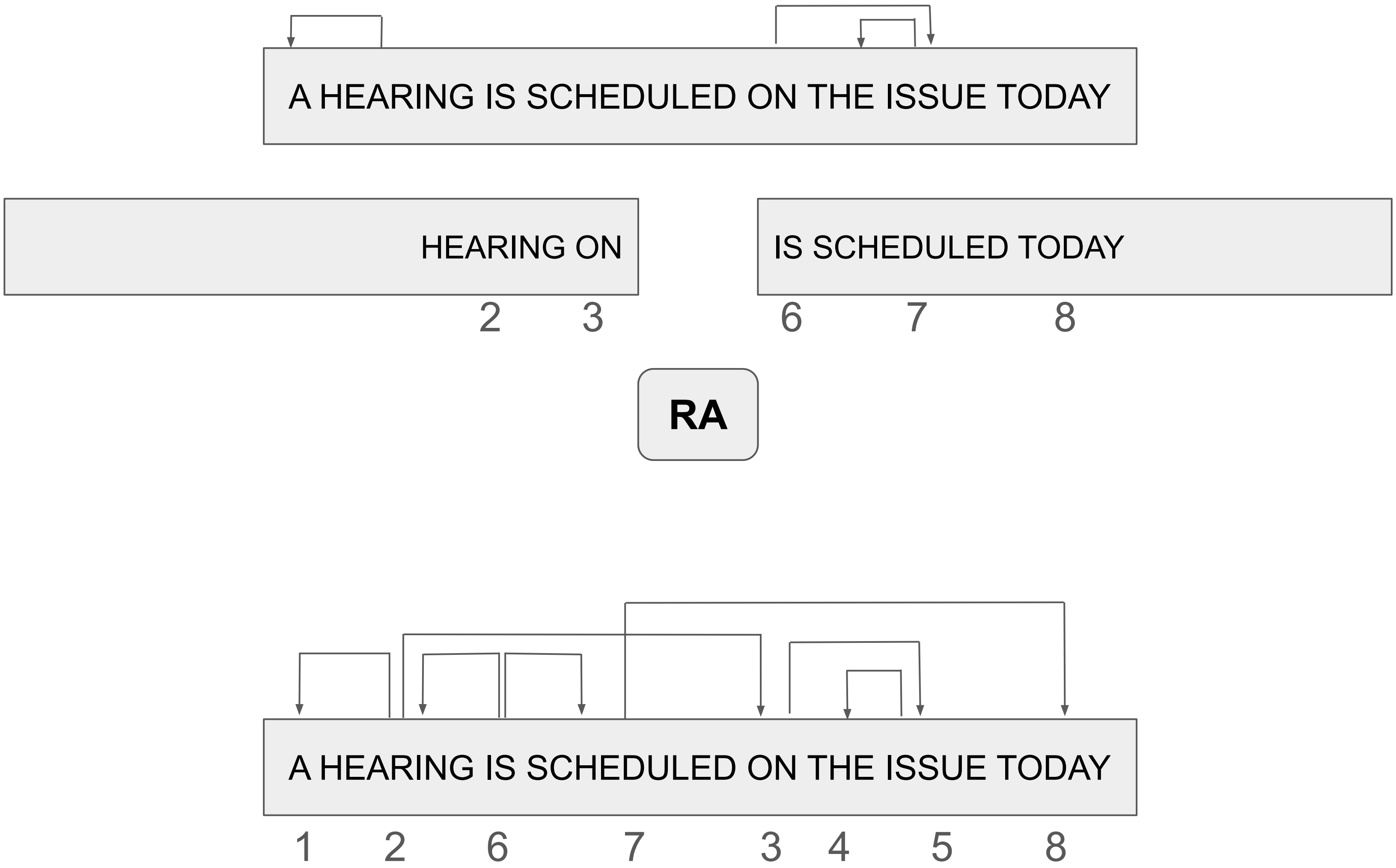

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING ON IS SCHEDULED THE
ISSUE TODAY
2 3 6 7 4
5 8
3x
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8

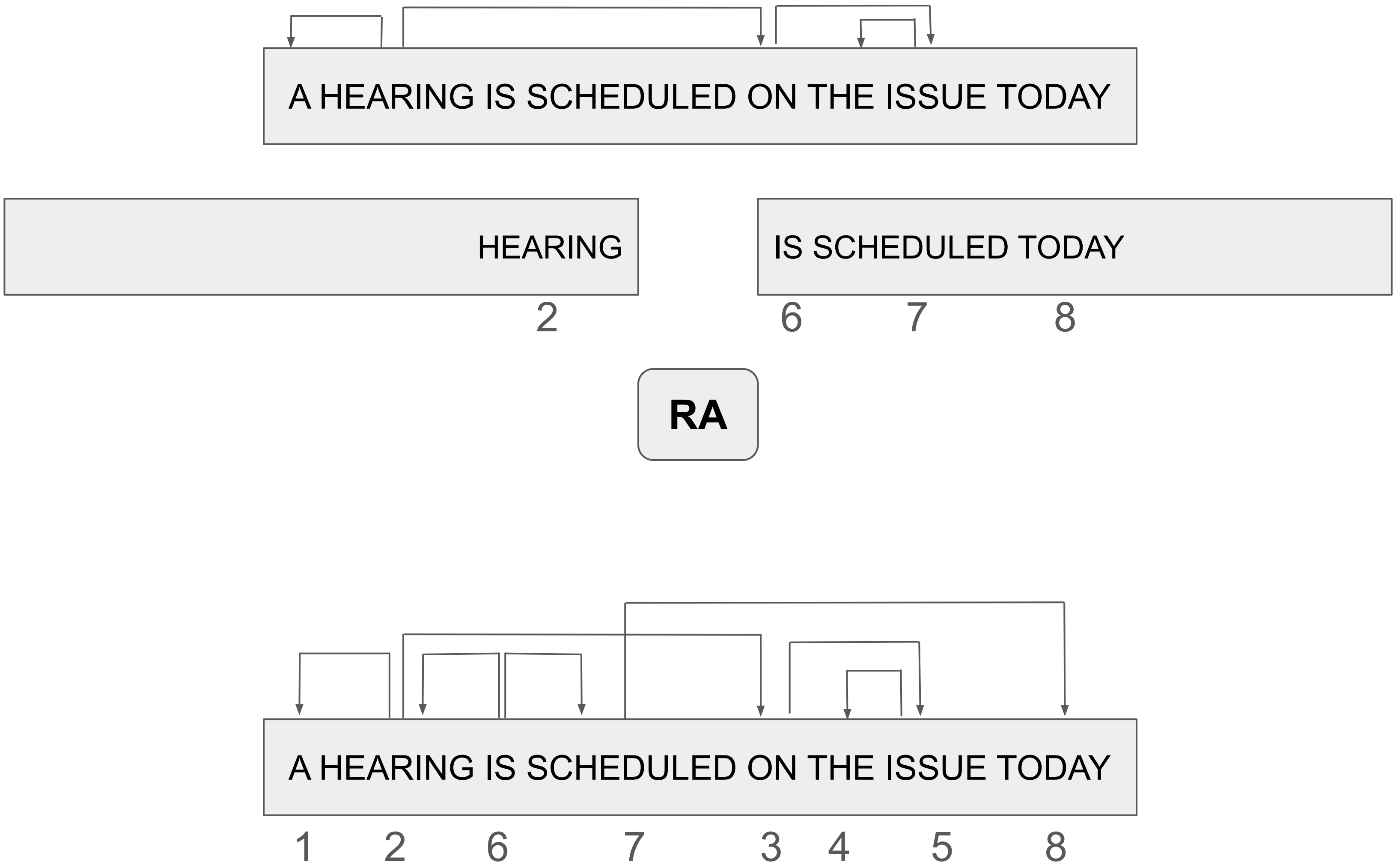

A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING ON THE
IS SCHEDULED ISSUE TODAY
2 3 4
6 7 5 8
2x SWAP
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8



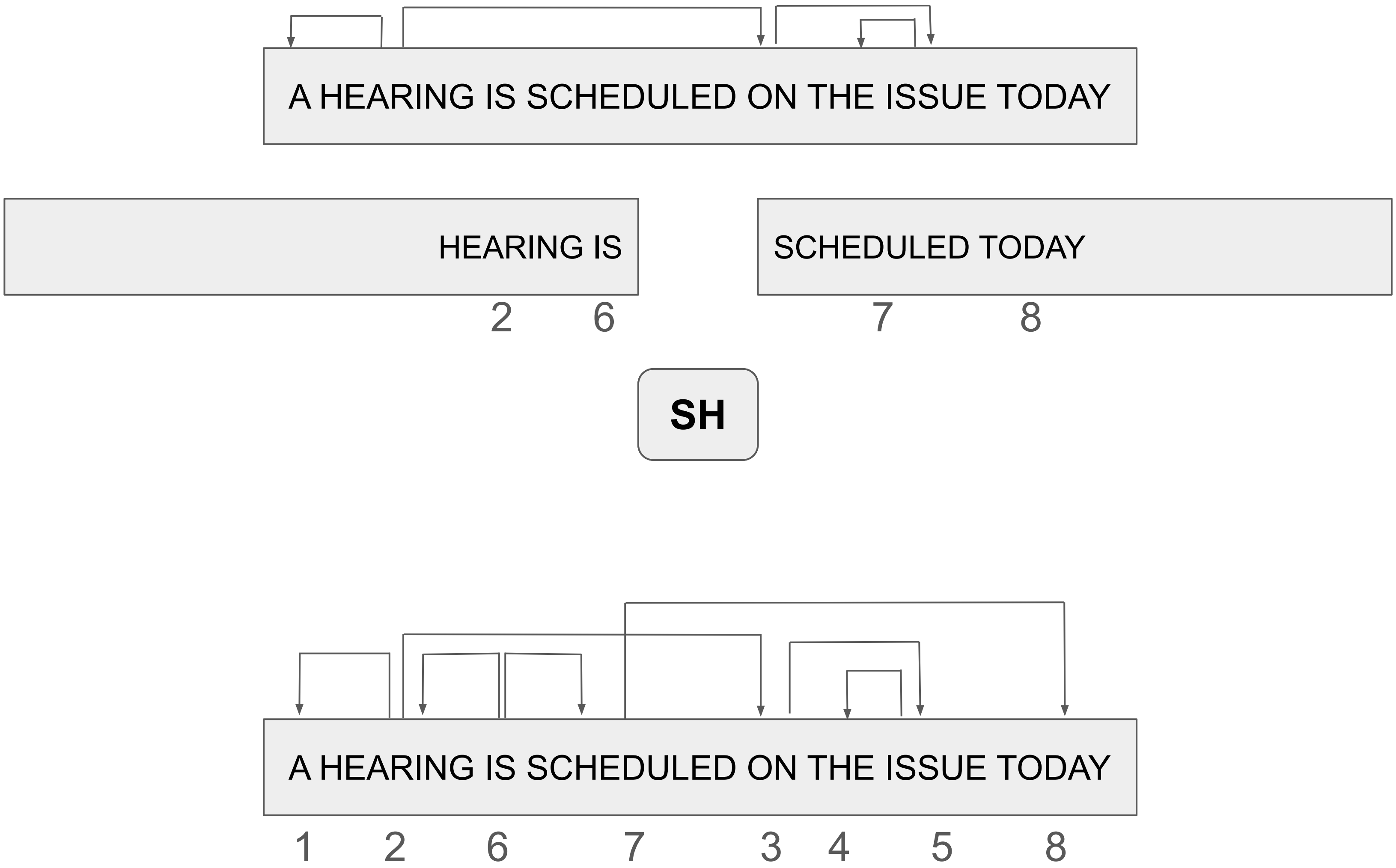
Rearranging
Recall Projective order


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING ON ISSUE
2 3 5
IS SCHEDULED TODAY
6 7 8
LA
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING ON
2 3
IS SCHEDULED TODAY
6 7 8
RA
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8



A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING
2
IS SCHEDULED TODAY
6
7
8
RA
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1
2
6
7
3
4
5
8


A HEARING IS SCHEDULED ON THE ISSUE TODAY
HEARING IS
2 6
SCHEDULED TODAY
7 8
SH
A HEARING IS SCHEDULED ON THE ISSUE TODAY
1 2 6 7 3 4 5 8

A HEARING IS SCHEDULED ON THE ISSUE TODAY

IS	-

6

LA-SH-SH-RA-RA

A HEARING IS SCHEDULED ON THE ISSUE TODAY

1 2 6 7 3 4 5 8

Results

Could now handle data without preprocessing/projectivizing beforehand.

Tests done on the English_EWT (English Web Treebank), and the Czech_CAC (Czech Academic Corpus) treebanks. Acquired from https://universaldependencies.org/

Gave similar results, ±1%, to baseline, when comparing Unlabelled Attachment Score (UAS). Similar to Nivre's comparisons.

EN_EWT : 97.13% projective. CS_CAC : 87.15% projective.

Method \ Treebank	en_ewt	cs_cac
Baseline (UAS)	0.659	0.674
Improved (UAS)	0.651	0.667

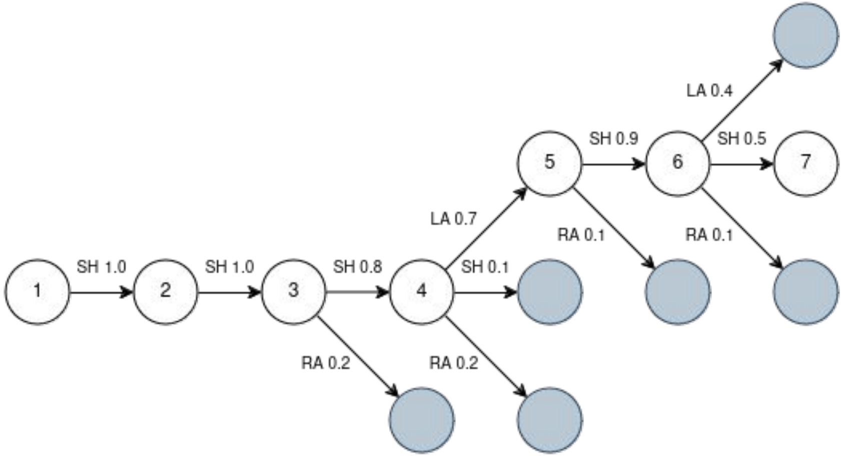
Adding beam search to the arc-standard parser

Greedy search

- Local best decision
- Simple and fast

Problem

- What if the local decision is incorrect?



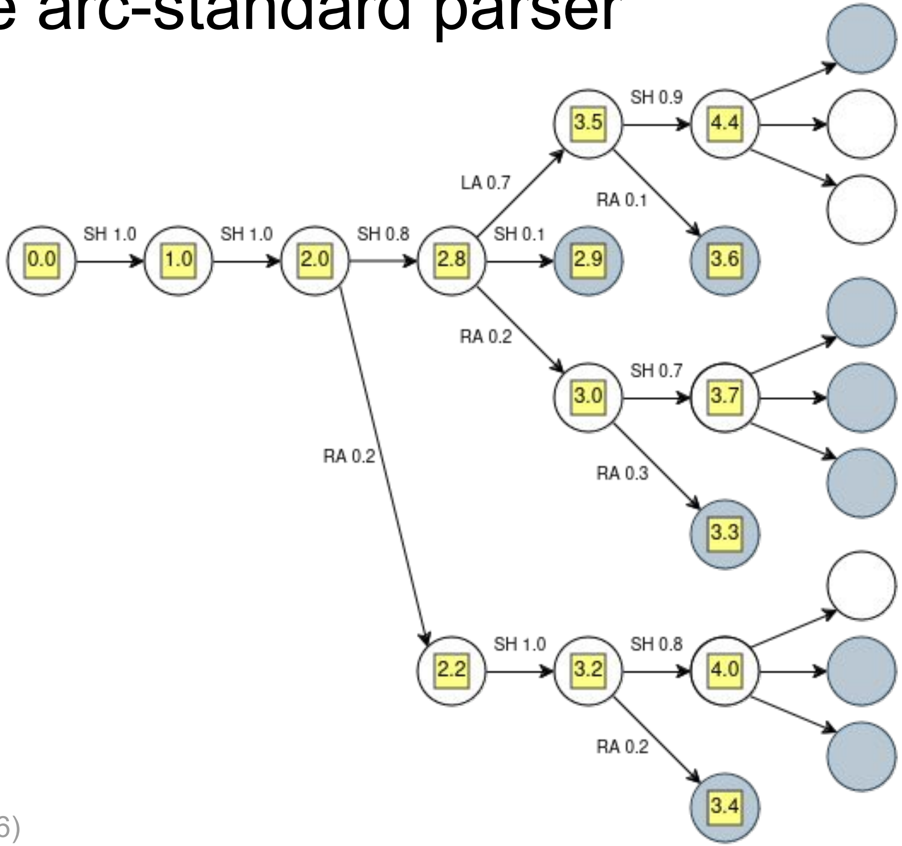
Adding beam search to the arc-standard parser

Beam search

- Keep a number of “best” states for each column
- Example: beam of width 3
- Accumulated score

Problem

- “Bad” predictions can still get high scores

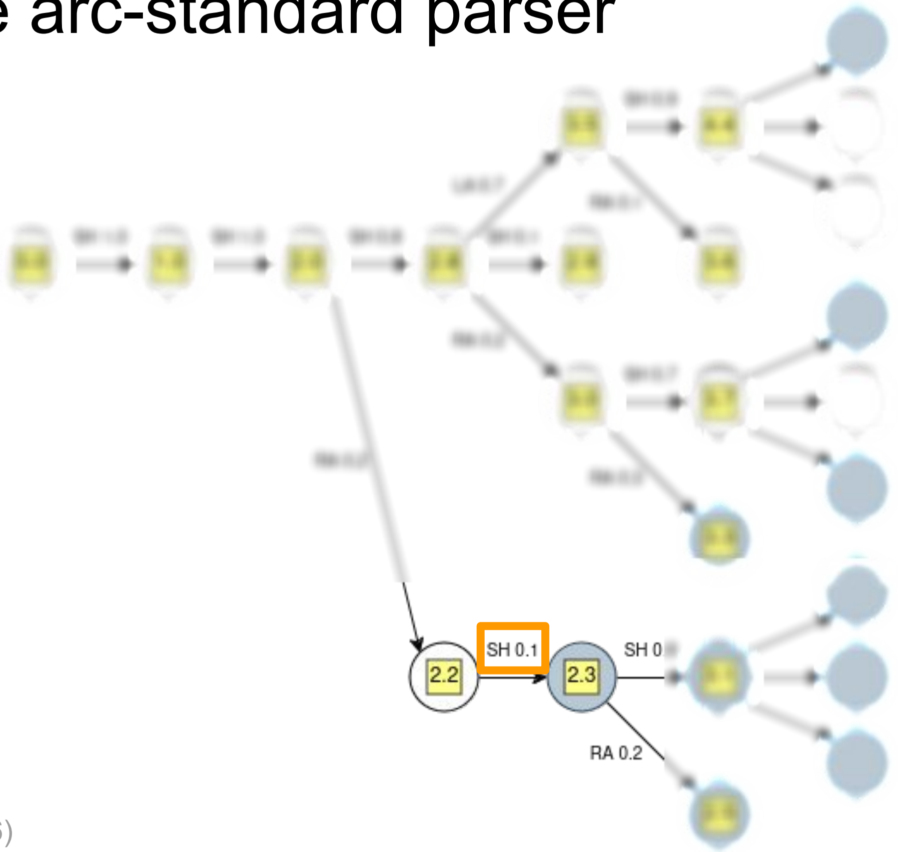


(Vaswani & Sagae 2016)

Adding beam search to the arc-standard parser

Error states

- Acts as a “sink” for bad predictions
- Error state examples used during training
- Generated from deviation from gold standard
- “Steals” some of the probability



(Vaswani & Sagae 2016)

Results

Beam search increased running time of prediction due to more processed states

Error states increased running time of training due to more training examples

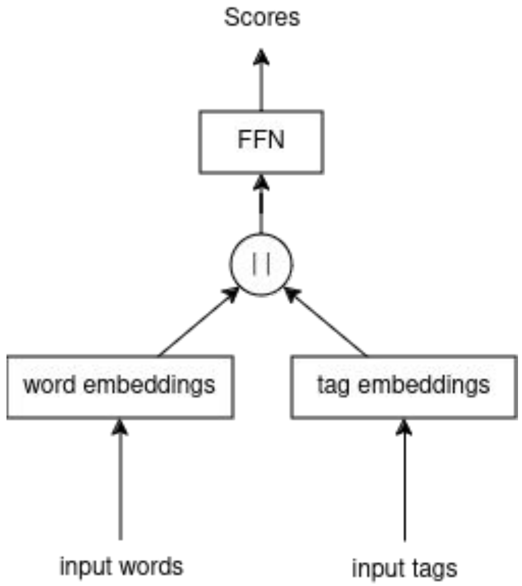
Worse performance than baseline might be due to rewarded early correct choices

Error states makes beam search less bad

Method \ Treebank	en_ewt	cs_cac
Baseline (UAS)	0.659	0.674
Beam search (UAS) β = 2	0.593	0.236
Error states (UAS) β = 2	0.652	0.650
Error states (UAS) β = 4	0.651	0.646

Adding attention to the parser

Improve underlying base model

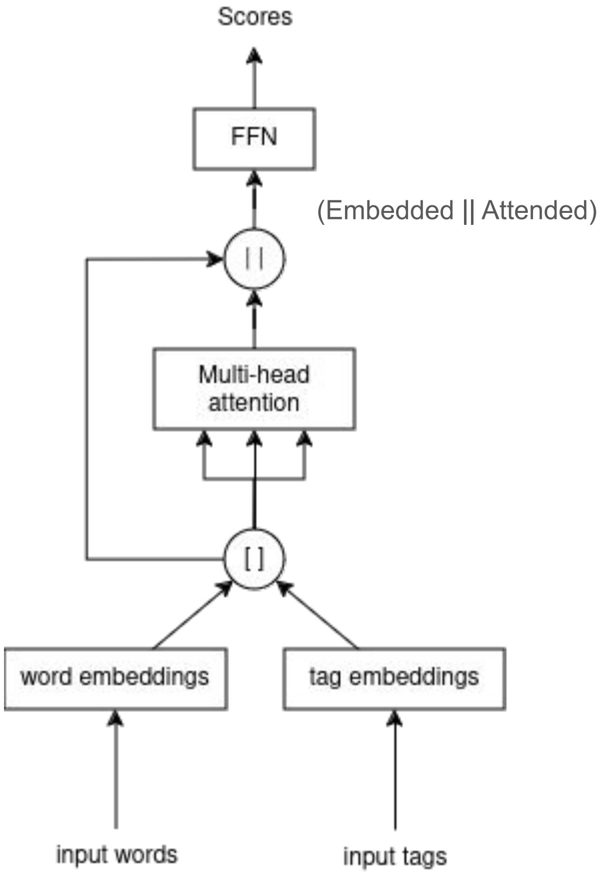


Adding attention to the parser

First attempt:

Multi-head self-attention with concatenation

Horrible results



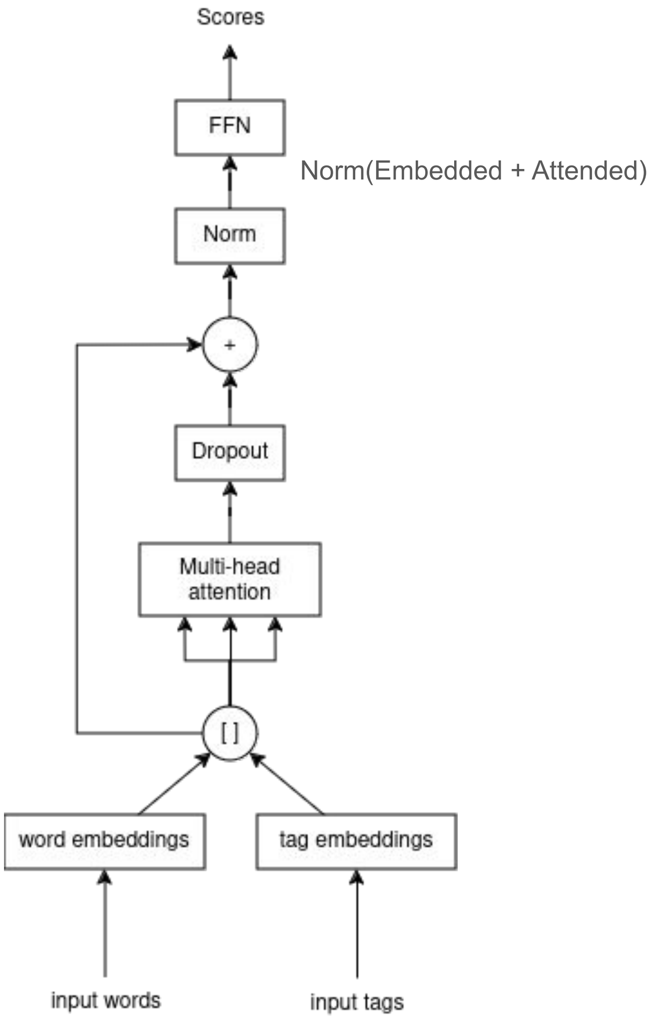
Adding attention to the parser

Second attempt:

Multi-head self-attention with addition and normalization

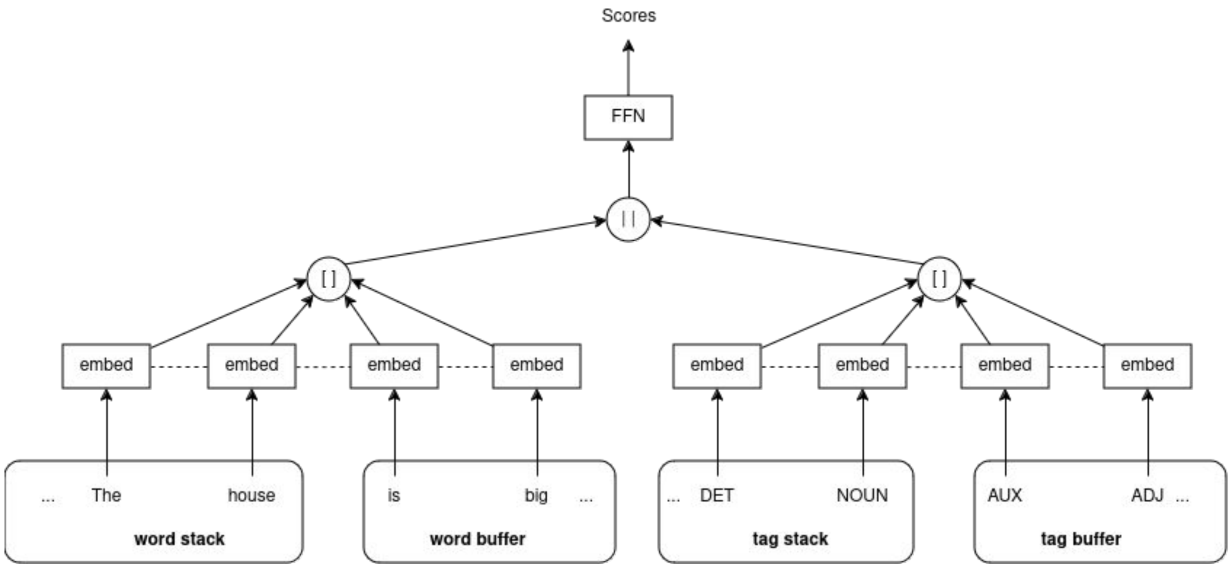
Equivalent results. Why?

Attention is all you need, Vaswani, et. al.



Adding attention to the parser

Too few features?

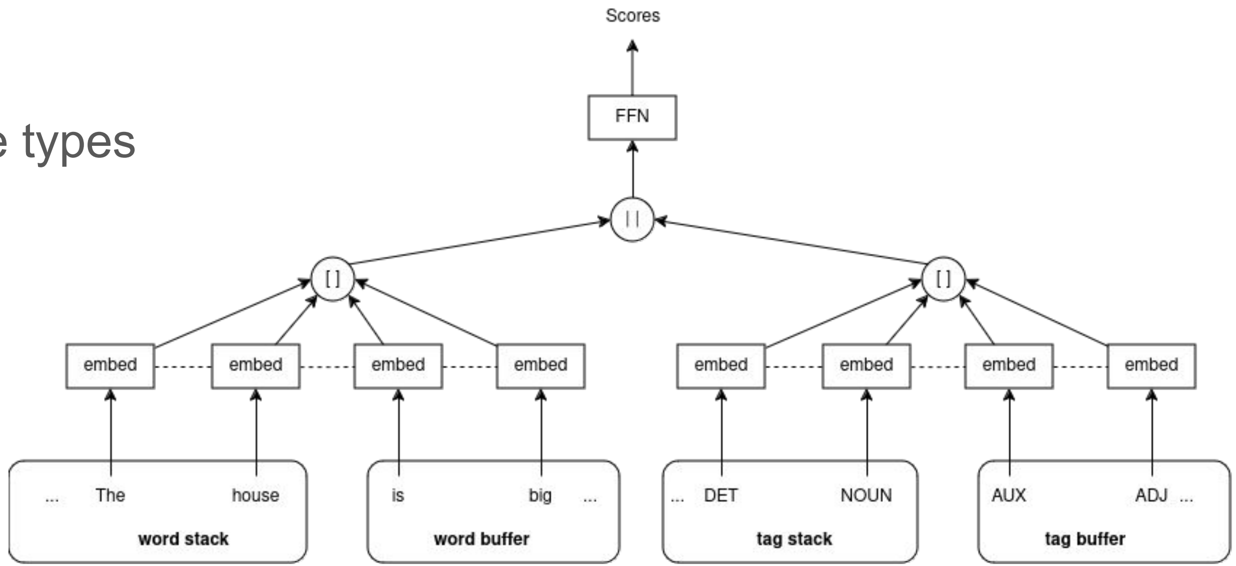


(visual representation. feature number imprecise)

Adding attention to the parser

Feature window experimentation

- 30 features best
- No bias towards feature types
- 78% (72%) UAS



(visual representation. feature number imprecise)

Conclusion

- Arc-parser with swap
  - Same performance as baseline on non-projective
  - In line with literature

- Beam search (error states)
  - Slower
  - Beam width 1-2 had about same performance
  - Beam width > 2 had worse performance

- Attention
  - No significant performance impact
  - Feature experimentation led to significant performance boost

Conclusion

- Arc-parser with swap + Attention model
  - Attention gave no performance boost with the arc-parser
  - Configuration of hyperparameters gave a boost

Model	Language	Accuracy	Total UAS	Raw UAS
Baseline	EN	0.884	0.659	0.703
Swap	EN	0.871	0.651	0.702
Swap + Hyper	EN	0.885	0.716	0.763
Swap + Hyper + Attention	EN	0.883	0.716	0.757
Baseline	CS	0.942	0.674	0.740
Swap	CS	0.938	0.667	0.739
Swap + Hyper	CS	0.941	0.692	0.762
Swap + Hyper + Attention	CS	0.939	0.694	0.764

Sources

- Non-Projective Dependency Parsing in Expected Linear Time (Nivre, ACL-IJCNLP 2009)
- Efficient Structured Inference for Transition-Based Parsing with Neural Networks and Error States (Vaswani & Sagae, TACL 2016)
- Vaswani et. al., Attention is all you need, In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc
- Bahdanau et. al., 2014, Neural Machine Translation by Jointly Learning to Align and Translate