Constituency Parsing

李宏毅 Hung-yi Lee

	One Sequence	Multiple Sequences
One Class	Sentiment Classification Stance Detection Veracity Prediction Intent Classification Dialogue Policy	NLI Search Engine Relation Extraction
Class for each Token	POS tagging Word segmentation Extractive Summarization Slotting Filling NER	
Copy from Input		Extractive QA
General Sequence	Abstractive Summarization Translation Grammar Correction NLG	General QA Chatbot State Tracker Task Oriented Dialogue
Other?	Parsing, Coreference Resolution	

Constituency Parsing

- Some text spans are constituents ("units")
- Each constituent has a label.

not constituent constituent

deep learning is very powerful

constituent constituent

NP ADJP

Constituency Parsing - Labels

Table 1.2. The Penn Treebank syntactic tagset

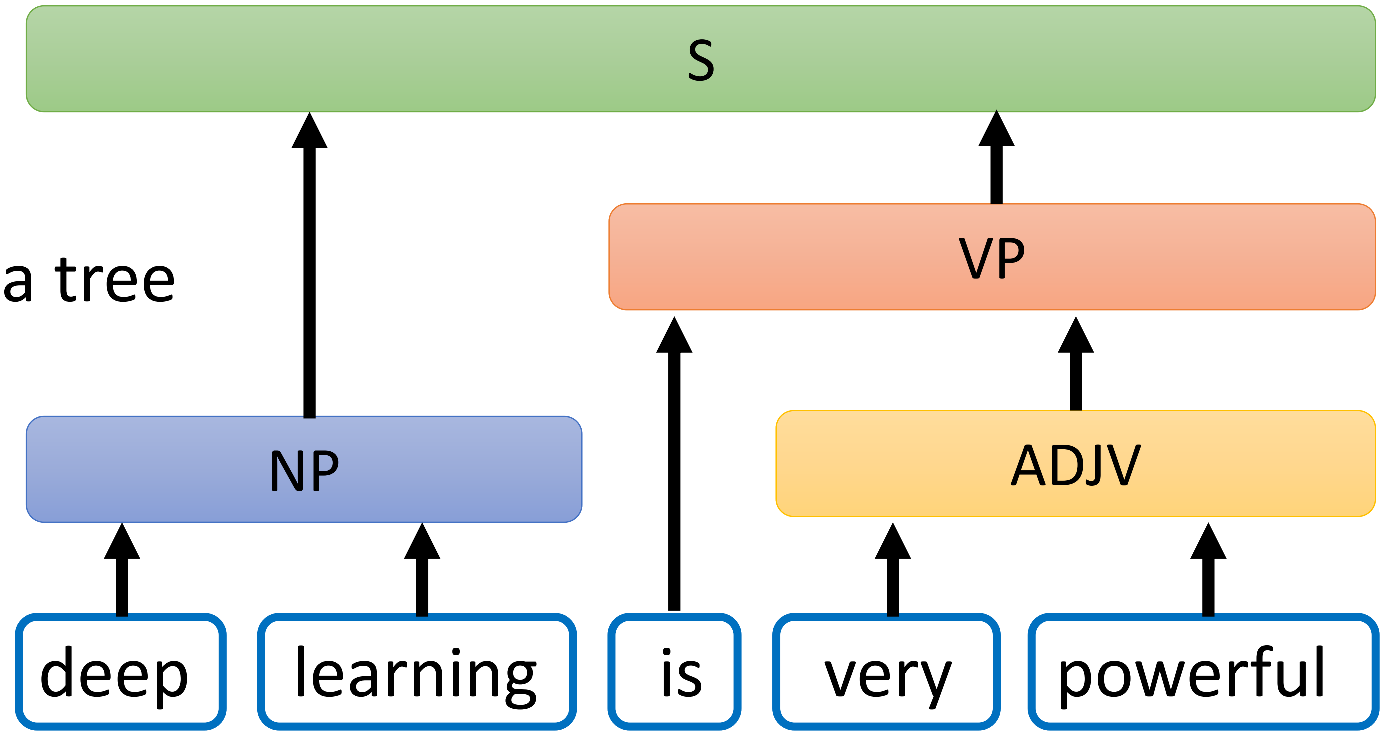
+ All POS tags

ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding *wh*-element
VP	Verb phrase
WHADVP	Wh-adverb phrase
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category
*	"Understood" subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses
T	Trace of wh-Constituent

(Only considering binary tree in this course for simplicity)

Constituency Parsing

Form a tree

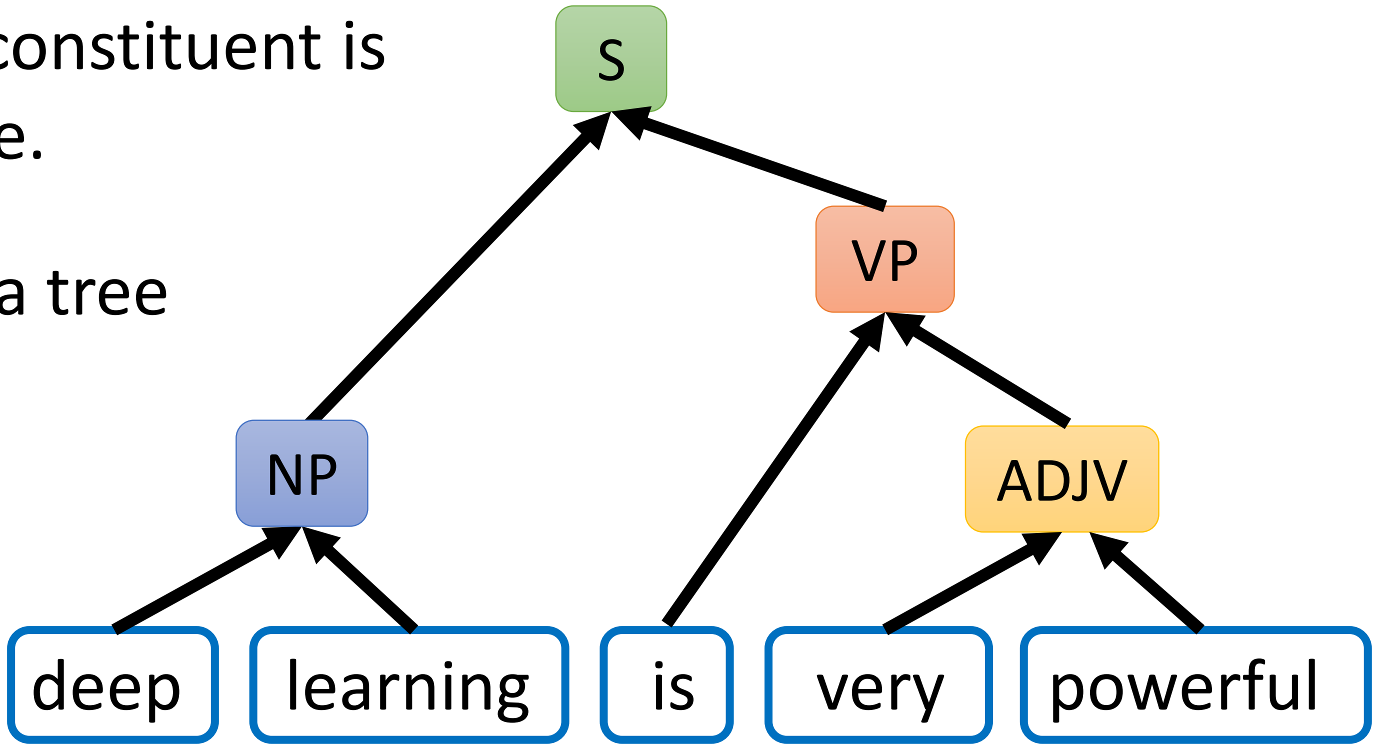


- Each word is a constituent (their labels are POS tags)
- Some consecutive constituents can form a larger one.

Constituency Parsing

Each constituent is a node.

Form a tree



- Each word is a constituent (their labels are POS tags)
- Some consecutive constituents can form a larger one.

Chart-based Approach

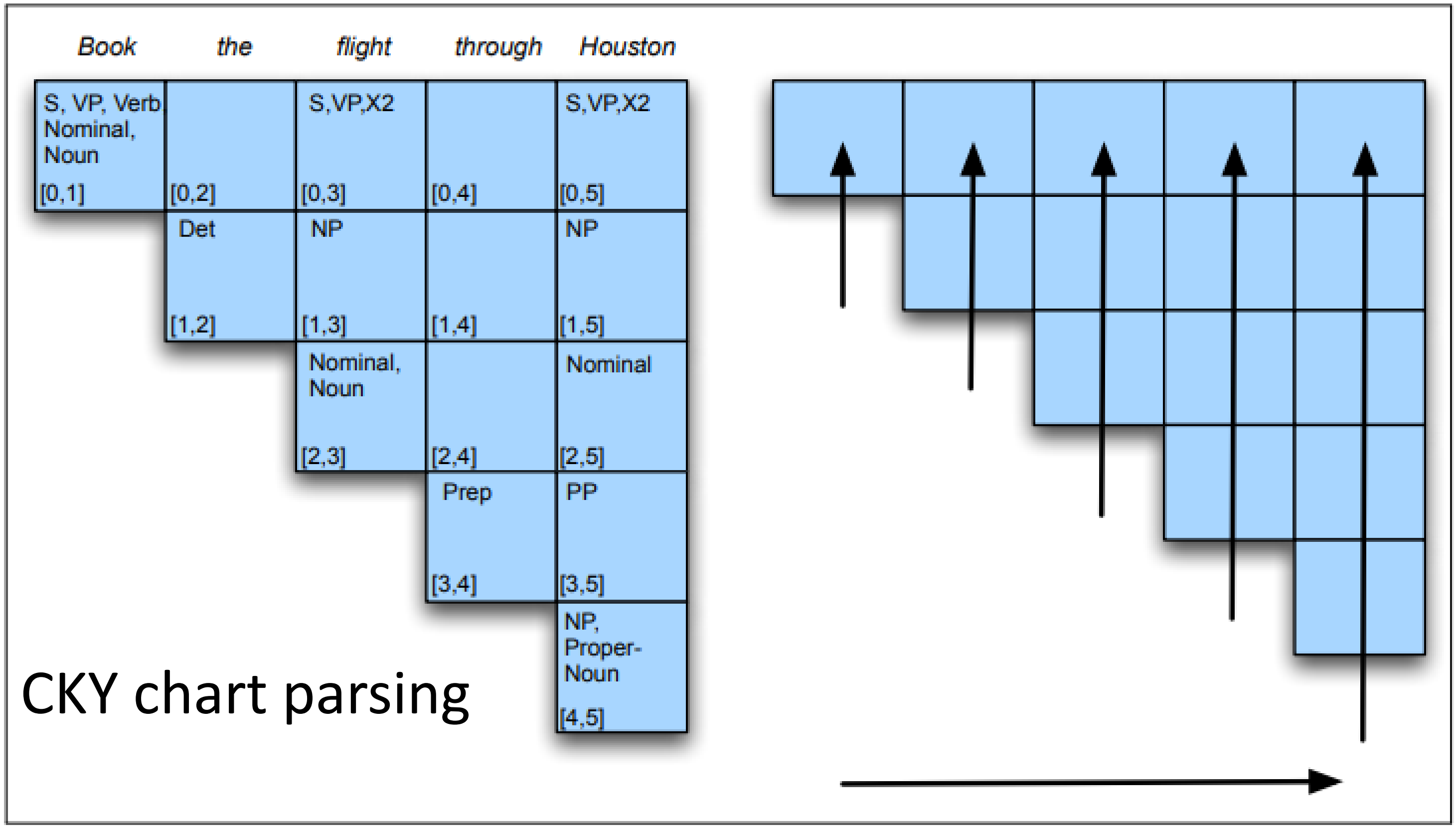
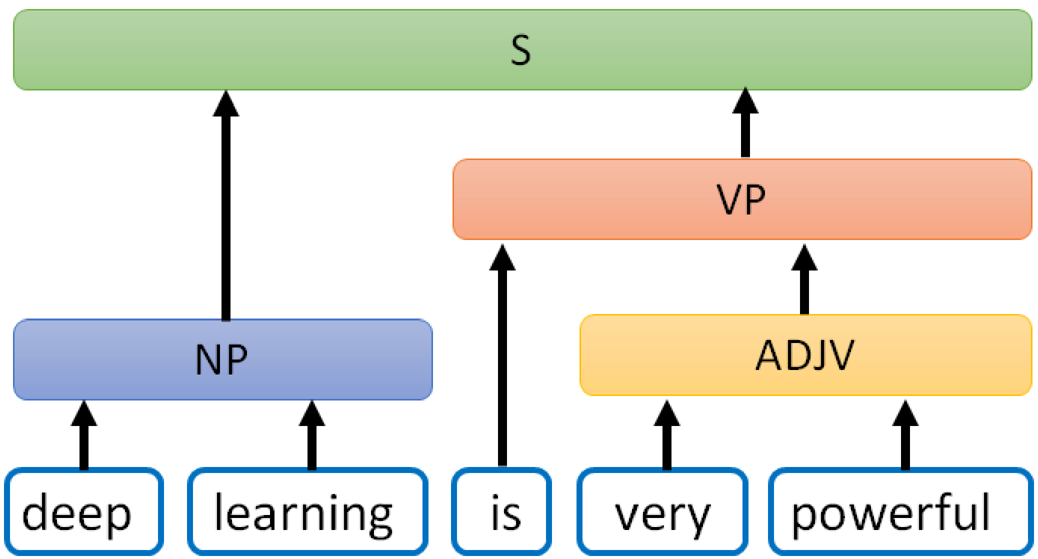


Figure 13.4 Completed parse table for *Book the flight through Houston.*

Source of image: https://web.stanford.edu/~jurafsky/slp3/13.pdf

Chart-based


S
VP
NP
ADJV
deep
learning
is
very
powerful

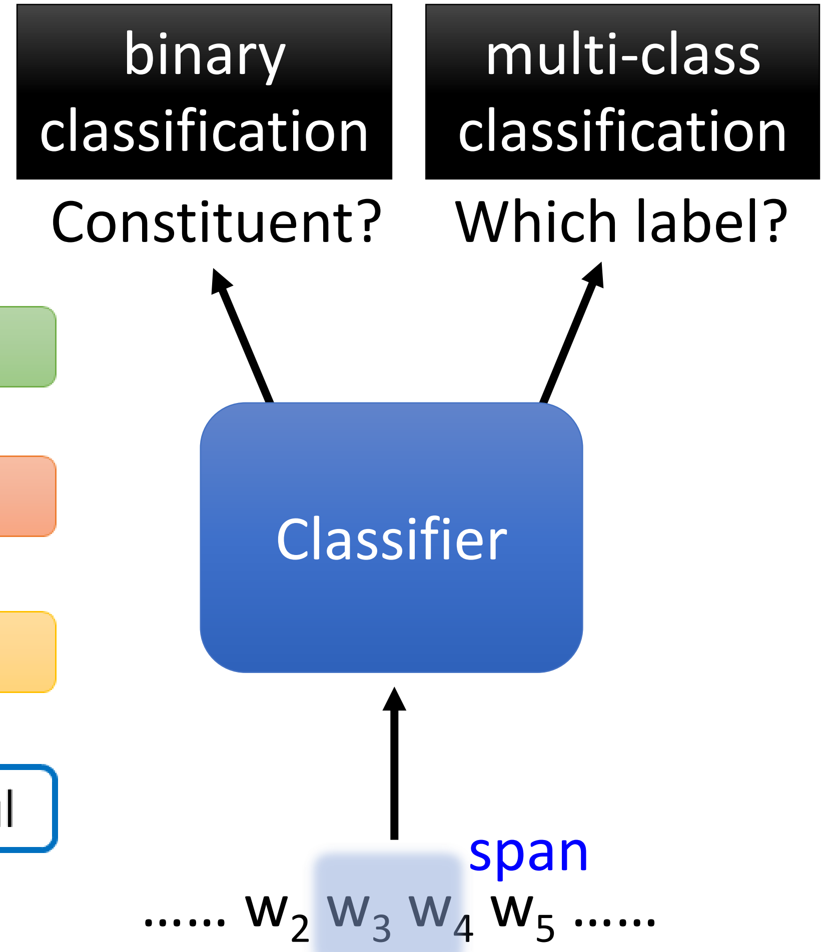
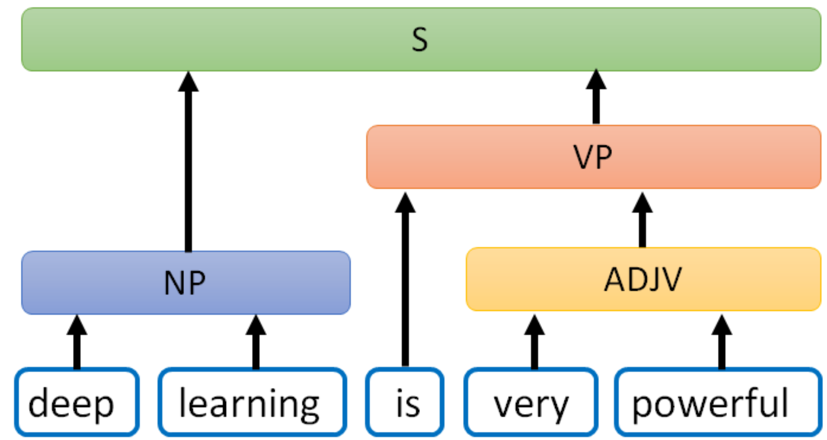

binary classification
multi-class classification
Constituent?
Which label?
Classifier
span
...... w2 w3 w4 w5 ......

Chart-based


S
VP
NP
ADJV
deep
learning
is
very
powerful

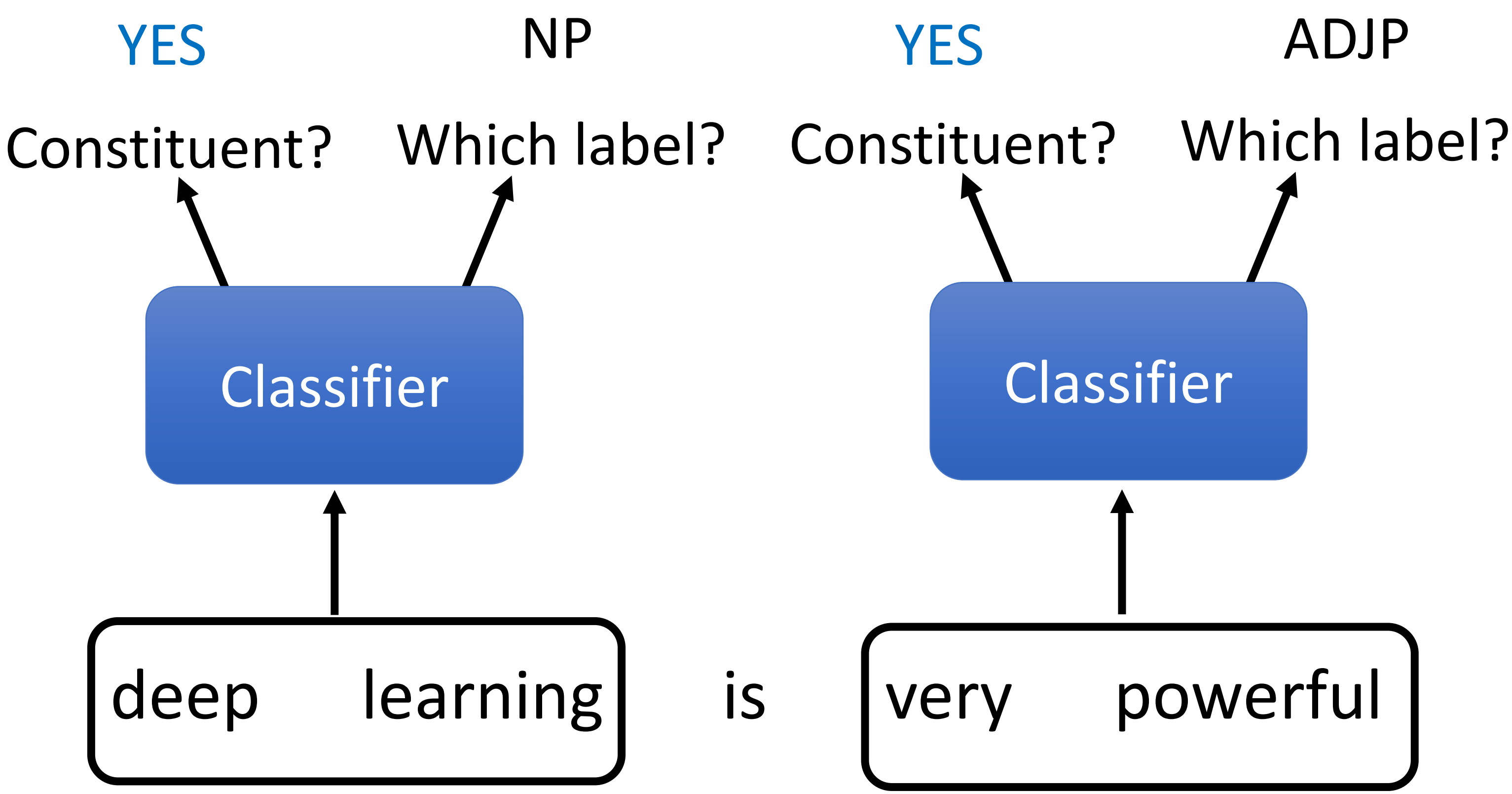
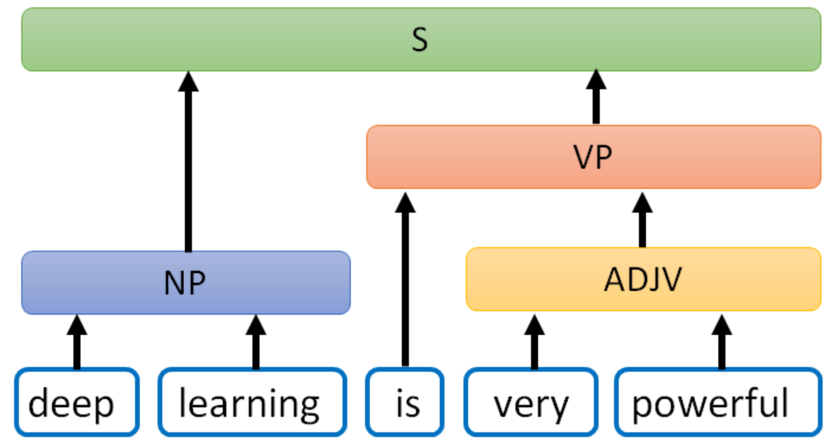

YES
NP
YES
ADJP
Constituent?
Which label?
Constituent?
Which label?
Classifier
Classifier
deep learning
is
very powerful

Chart-based


S
VP
NP
ADJV
deep
learning
is
very
powerful

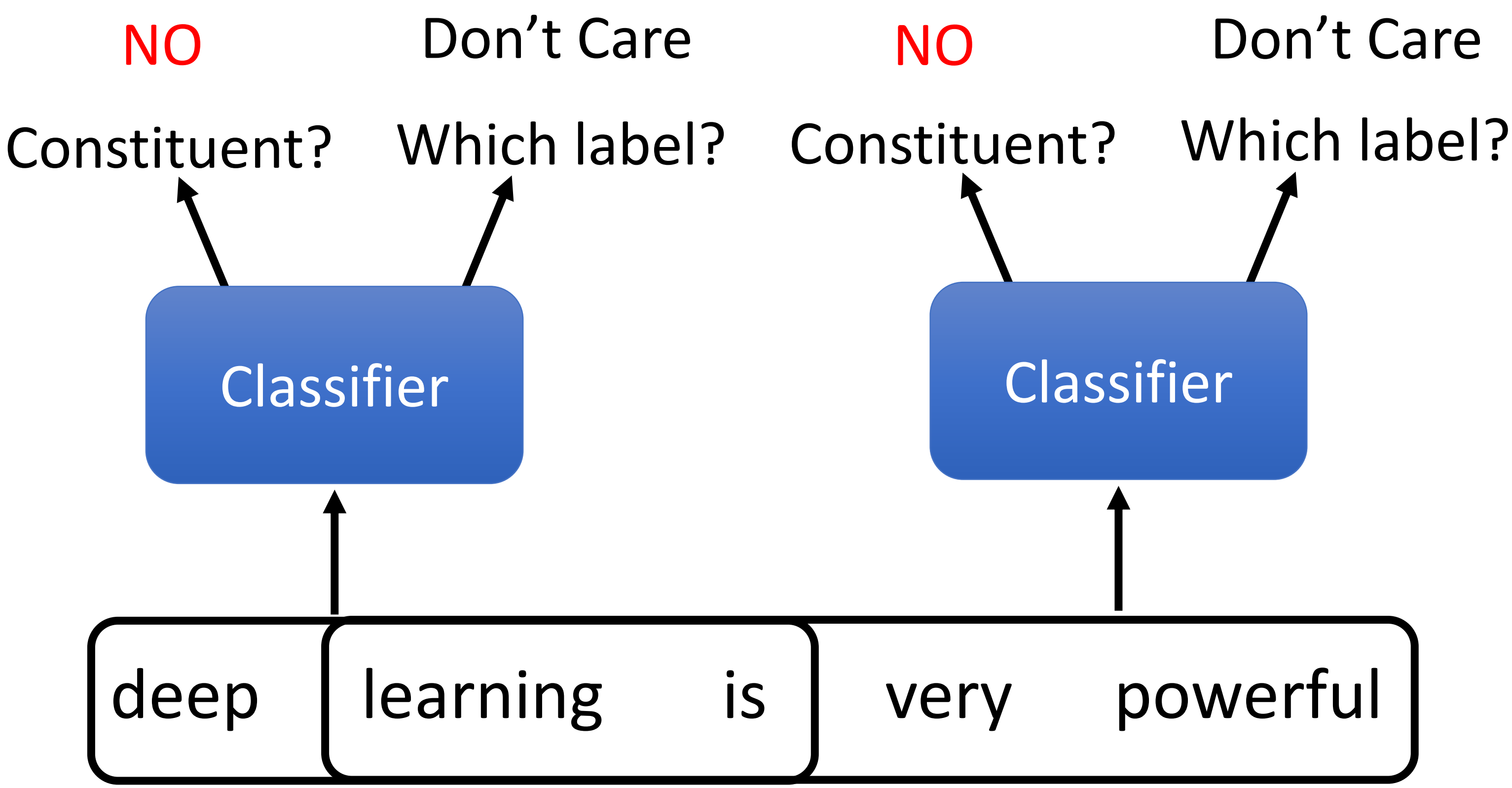

NO
Don’t Care
NO
Don’t Care
Constituent?
Which label?
Constituent?
Which label?
Classifier
Classifier
deep
learning
is
very
powerful

Chart-based – Classifier

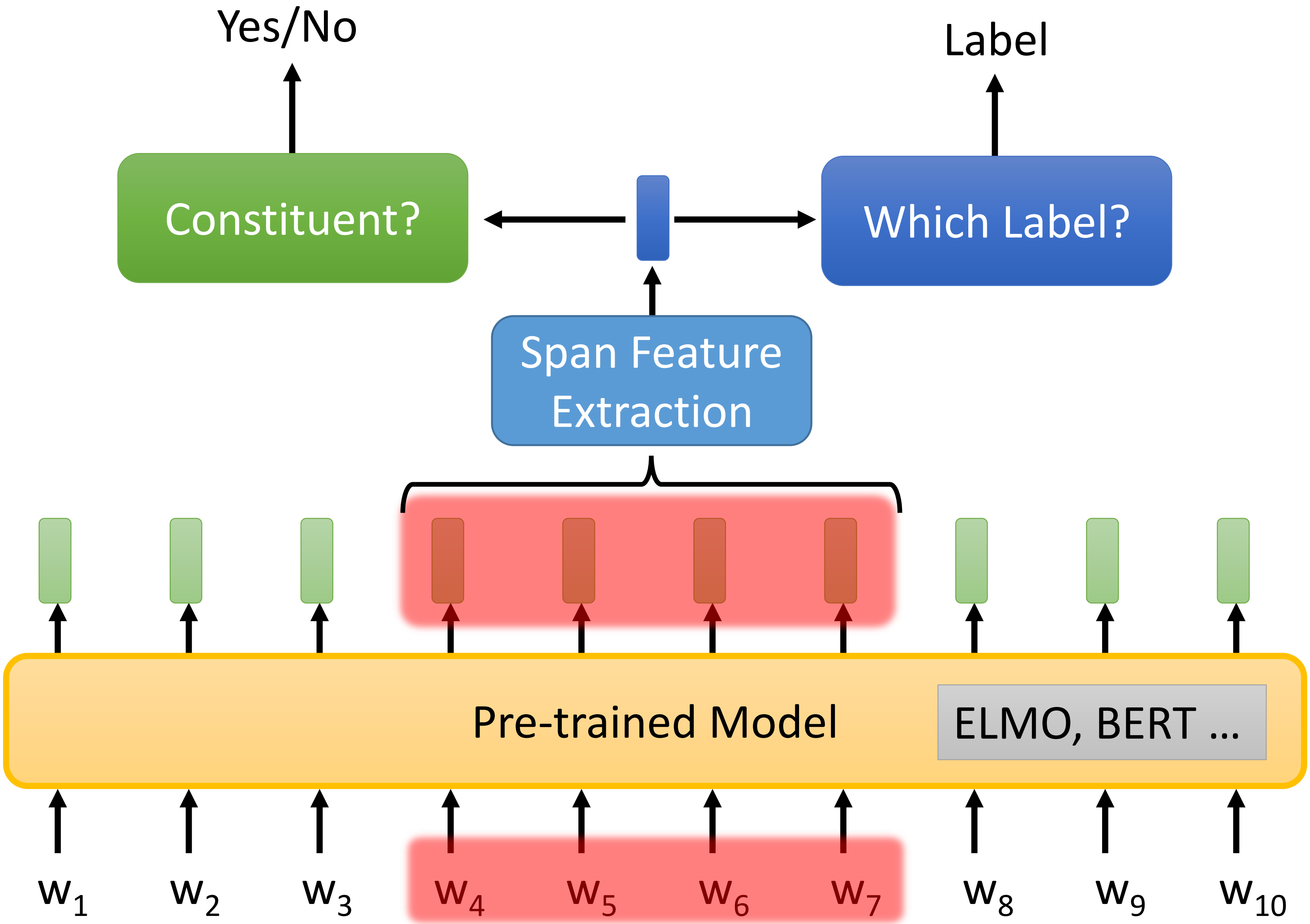
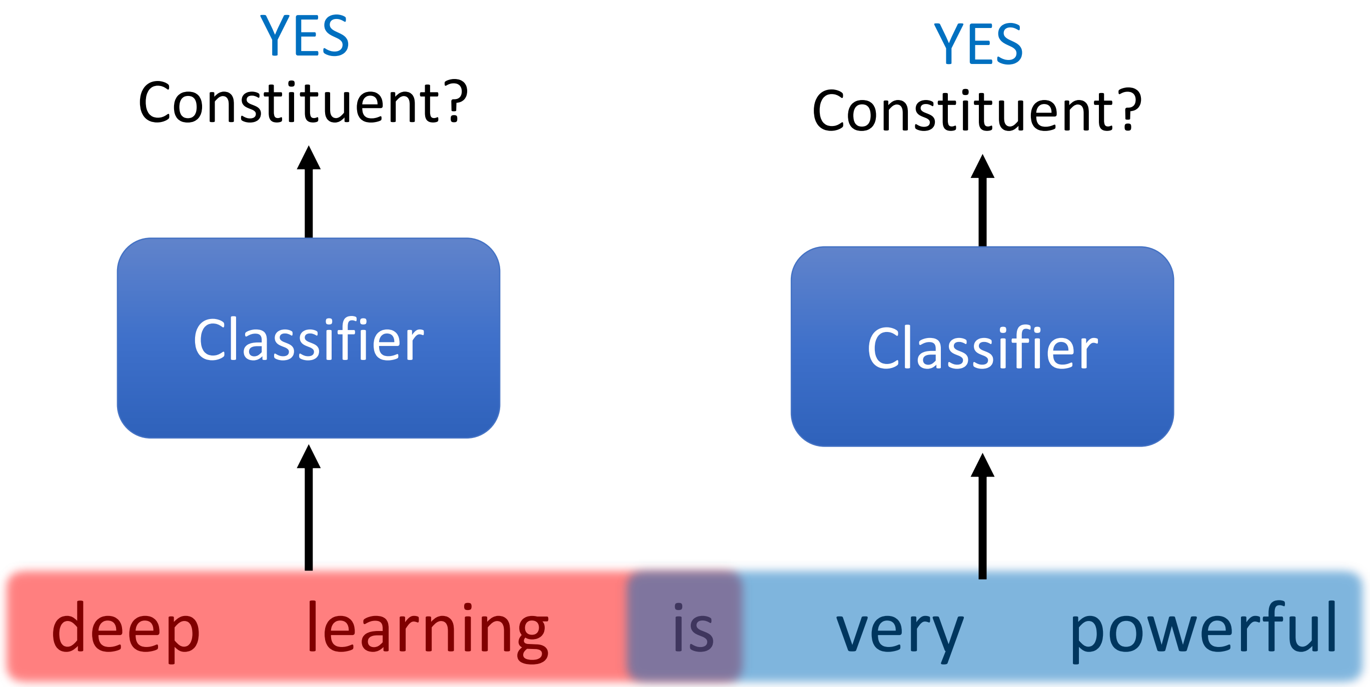


Chart-based

- Given a sequence with N tokens, then run the classifier N(N-1)/2 times ……

Contradiction!



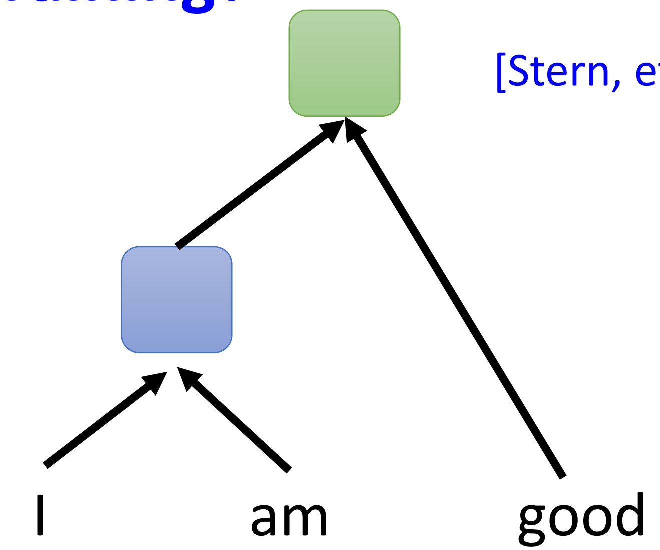
Inference

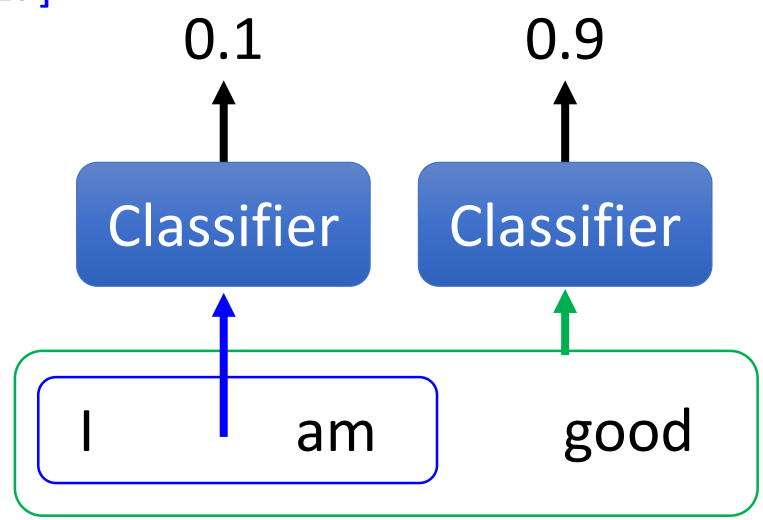
Training?

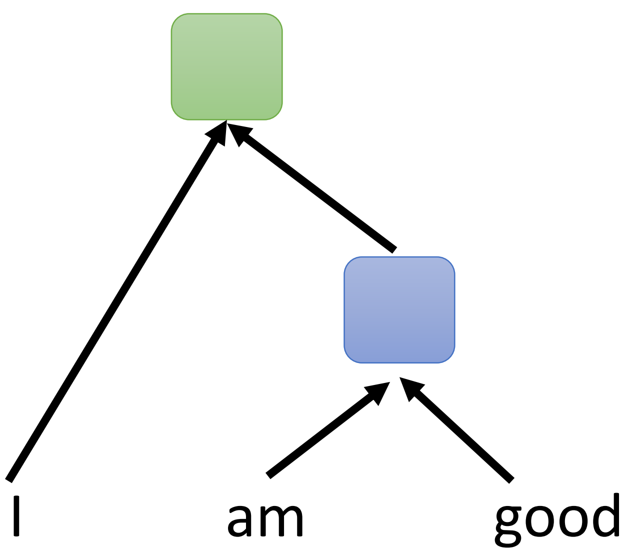
Enumerate all possible trees, and use the classifier to give scores.

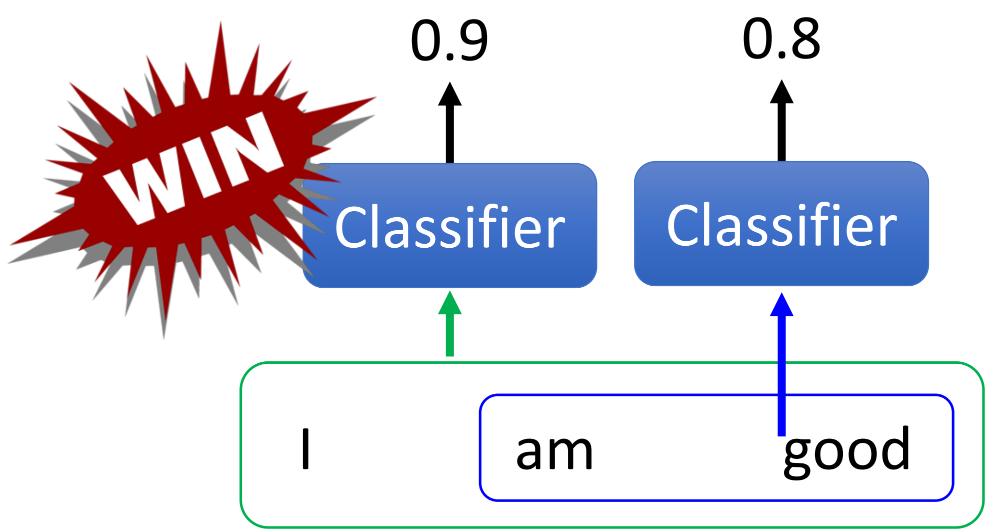
where you need CKY algorithm

[Stern, et al., ACL'17]









Transition-based

Input: *The hungry cat meows .*

<table>
<tr><th></th><th>Stack</th><th>Buffer</th><th>Action</th></tr>
<tr><td>0</td><td></td><td>The | hungry | cat | meows | .</td><td>NT(S)</td></tr>
<tr><td>1</td><td>(S</td><td>The | hungry | cat | meows | .</td><td>NT(NP)</td></tr>
<tr><td>2</td><td>(S | (NP</td><td>The | hungry | cat | meows | .</td><td>SHIFT</td></tr>
<tr><td>3</td><td>(S | (NP | The</td><td>hungry | cat | meows | .</td><td>SHIFT</td></tr>
<tr><td>4</td><td>(S | (NP | The | hungry</td><td>cat | meows | .</td><td>SHIFT</td></tr>
<tr><td>5</td><td>(S | (NP | The | hungry | cat</td><td>meows | .</td><td>REDUCE</td></tr>
<tr><td>6</td><td>(S | (NP The hungry cat)</td><td>meows | .</td><td>NT(VP)</td></tr>
<tr><td>7</td><td>(S | (NP The hungry cat) | (VP</td><td>meows | .</td><td>SHIFT</td></tr>
<tr><td>8</td><td>(S | (NP The hungry cat) | (VP meows</td><td>.</td><td>REDUCE</td></tr>
<tr><td>9</td><td>(S | (NP The hungry cat) | (VP meows)</td><td>.</td><td>SHIFT</td></tr>
<tr><td>10</td><td>(S | (NP The hungry cat) | (VP meows) | .</td><td></td><td>REDUCE</td></tr>
<tr><td>11</td><td>(S (NP The hungry cat) (VP meows) .)</td><td></td><td></td></tr>
</table>

Source of image: https://arxiv.org/pdf/1602.07776.pdf

Transition-based

[Dyer, et al., NAACL'16]

Stack (empty at the beginning)

Buffer deep learning is very powerful

Actions

CREATE(X)	SHIFT	REDUCE
Create a constitute X	More a token from buffer to stack	Close a constitute
(X = NP, VP …)		

Transition-based

(empty at the beginning)

CREATE(S)

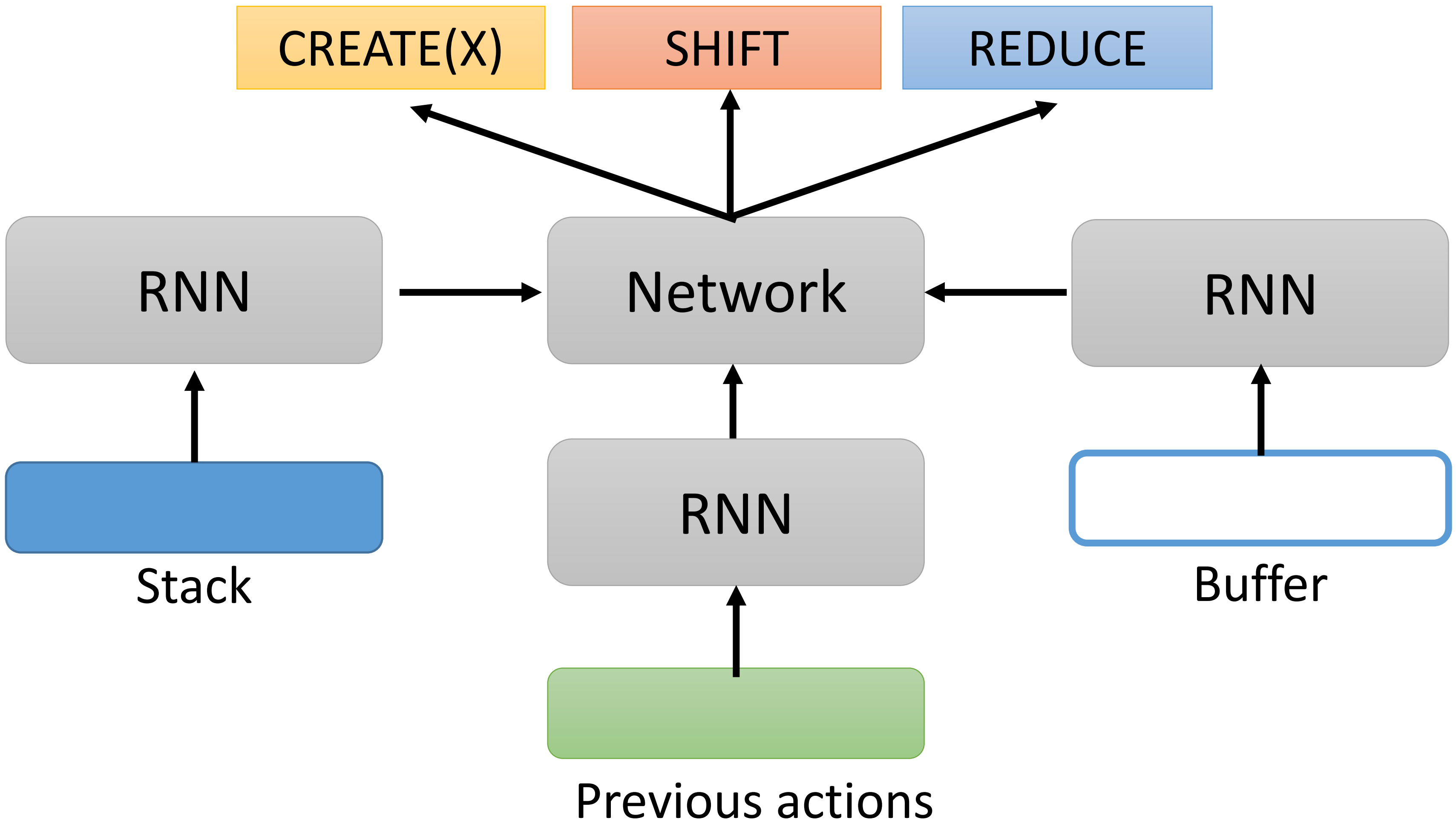
deep learning is very powerful

Transition-based

(S (NP deep learning) (VP is
(ADJV very powerful)))

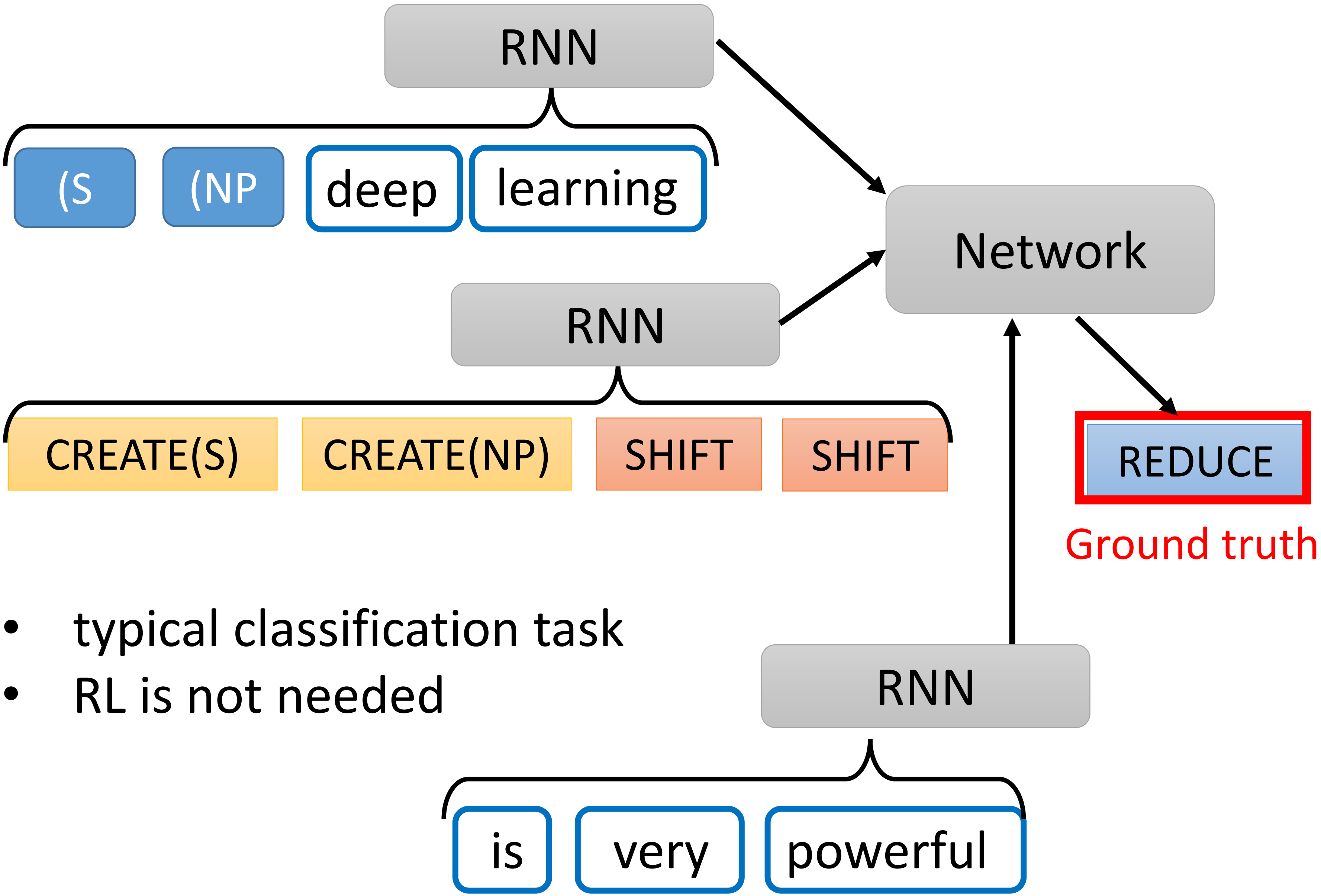
CREATE(S) CREATE(NP) SHIFT SHIFT REDUCE
CREATE(VP) SHIFT CREATE(ADJV) SHIFT SHIFT
REDUCE REDUCE REDUCE

RNN Grammar



CREATE(X)
SHIFT
REDUCE
RNN
Network
RNN
RNN
Stack
Buffer
Previous actions

RNN Grammar – Training



- typical classification task
- RL is not needed

Grammar as a Foreign Language

Oriol Vinyals*
Google
vinyals@google.com

Lukasz Kaiser*
Google
lukaszkaiser@google.com

Terry Koo
Google
terrykoo@google.com

Slav Petrov
Google
slav@google.com

Ilya Sutskever
Google
ilyasu@google.com

Geoffrey Hinton
Google
geoffhinton@google.com

[Vinyals, et al., NIPS'15]

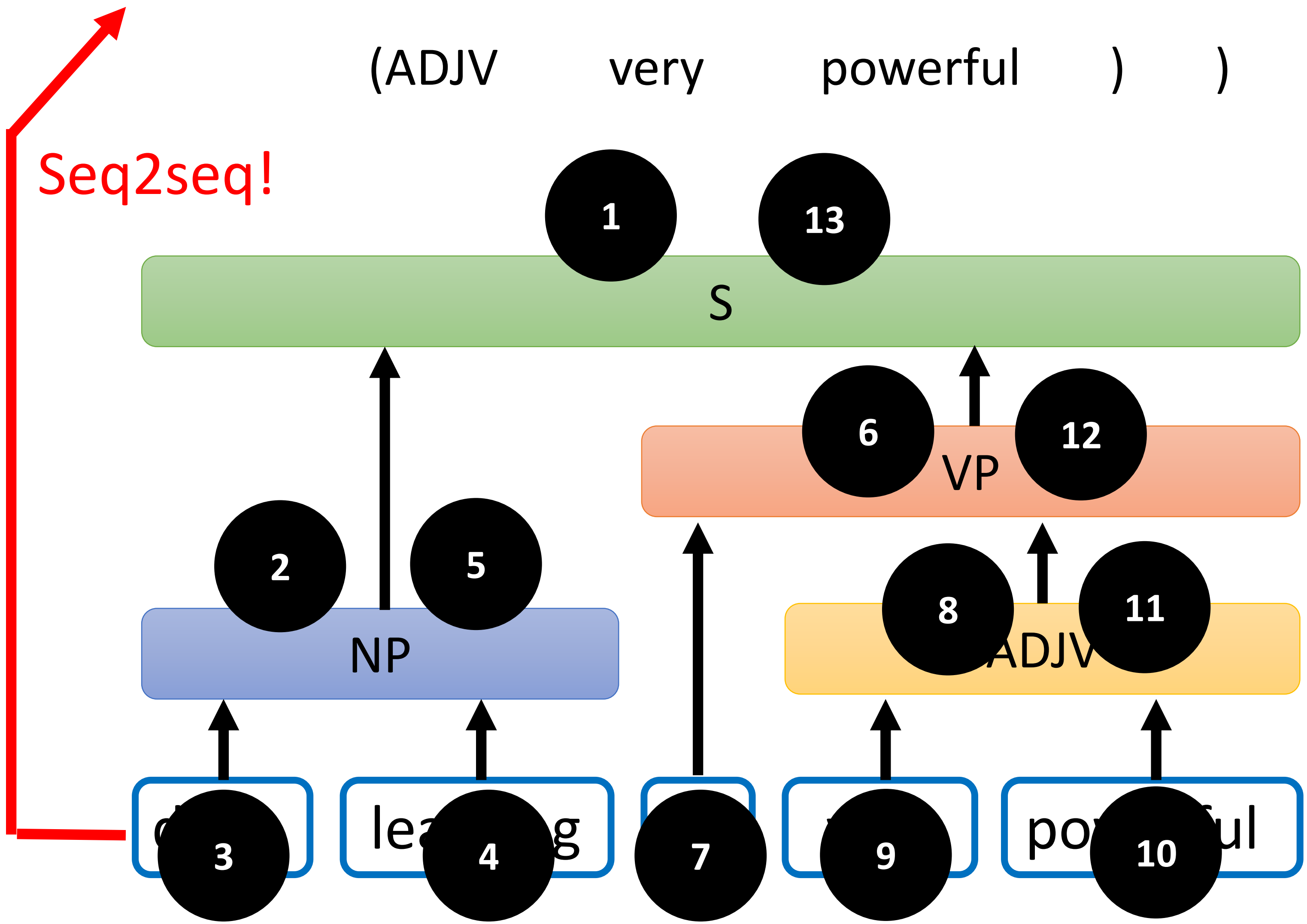
Source of image: https://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf

Tree to Sequence

Of course, you can try other tree traversal approaches

[Liu, et al., TACL'17]

(S (NP deep learning) (VP is (ADJV very powerful)))



Seq2seq v.s. RNN grammar

(S (NP deep learning) (VP is (ADJV very powerful)))

[Vinyals, et al., NIPS'15]

CREATE(S) CREATE(NP) SHIFT SHIFT REDUCE

CREATE(VP) SHIFT CREATE(ADJV) SHIFT SHIFT

REDUCE REDUCE REDUCE

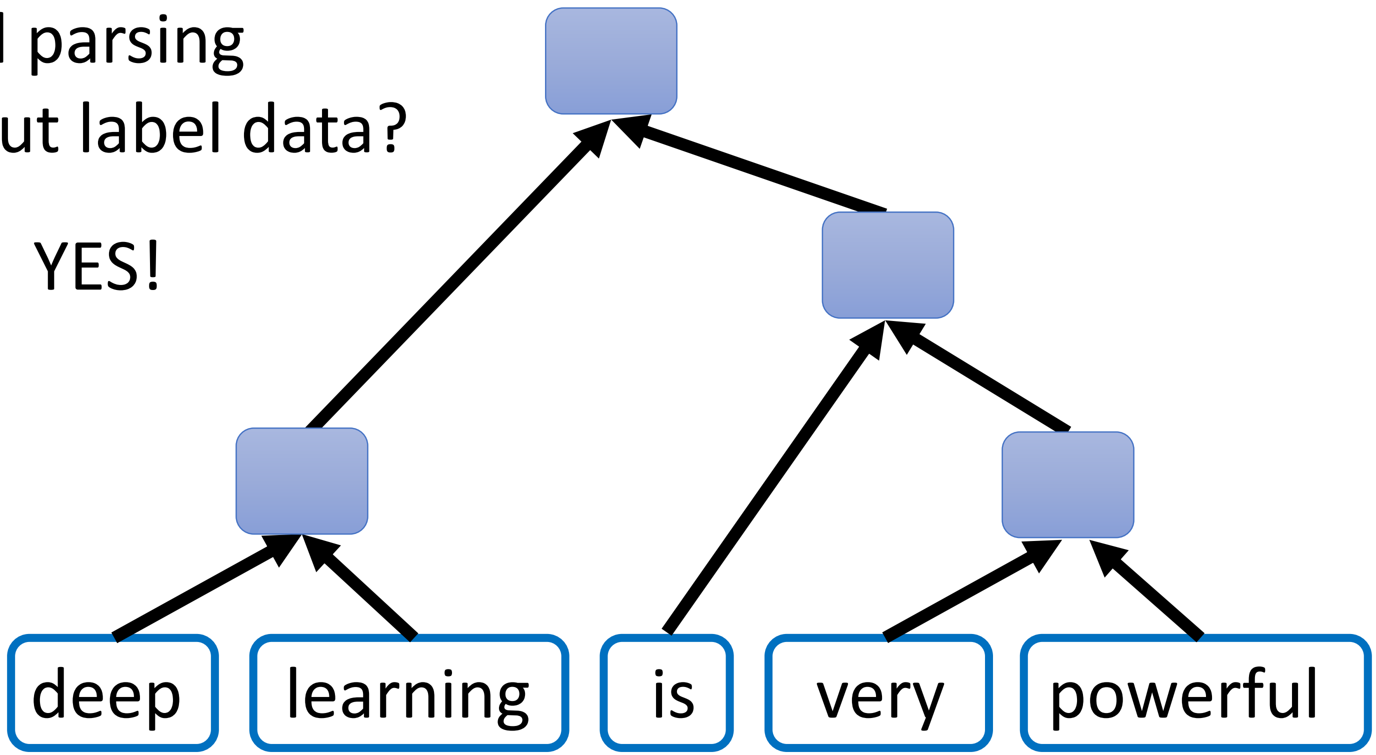
[Dyer, et al., NAACL'16]

deep learning is very powerful

Unsupervised Parsing?

Can we find parsing trees without label data?

YES!



Reference: https://youtu.be/YIuBHB9Ejok

Reference

- [Vinyals, et al., NIPS'15] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, Grammar as a foreign language, NIPS, 2015
- [Dyer, et al., NAACL'16] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, Noah A. Smith, Recurrent Neural Network Grammars, NAACL, 2016
- [Stern, et al., ACL'17] Mitchell Stern, Jacob Andreas, Dan Klein, A Minimal Span-Based Neural Constituency Parser, ACL,2017
- [Liu, et al., TACL'17] Jiangming Liu, Yue Zhang, In-Order Transition-based Constituent Parsing, TACL, 2017