What Do Recurrent Neural Network Grammars Learn About Syntax?

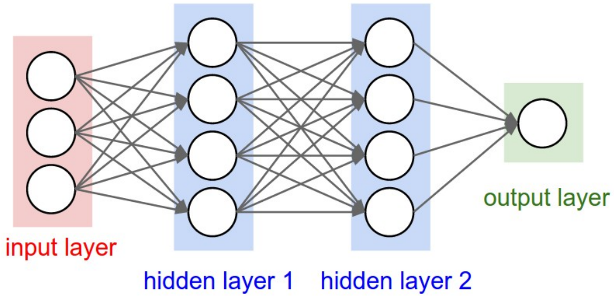
Adhiguna Kuncoro   Miguel Ballesteros   Lingpeng Kong
Chris Dyer   Graham Neubig   Noah A. Smith

Carnegie Mellon University   IBM   DeepMind   University of Washington

Language Models Are Mini-Linguists!

Syntax?

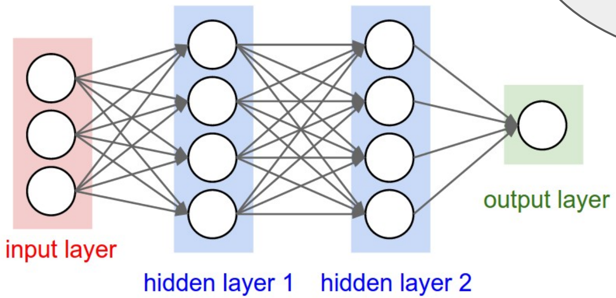
Semantics?



Language Models Are Mini-Linguists!

Syntax?

Semantics?

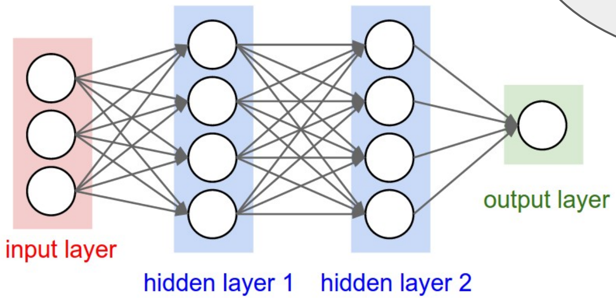


What syntactic phenomena do you learn?

Language Models Are Mini-Linguists!

Syntax?

Semantics?



What syntactic phenomena do you learn?

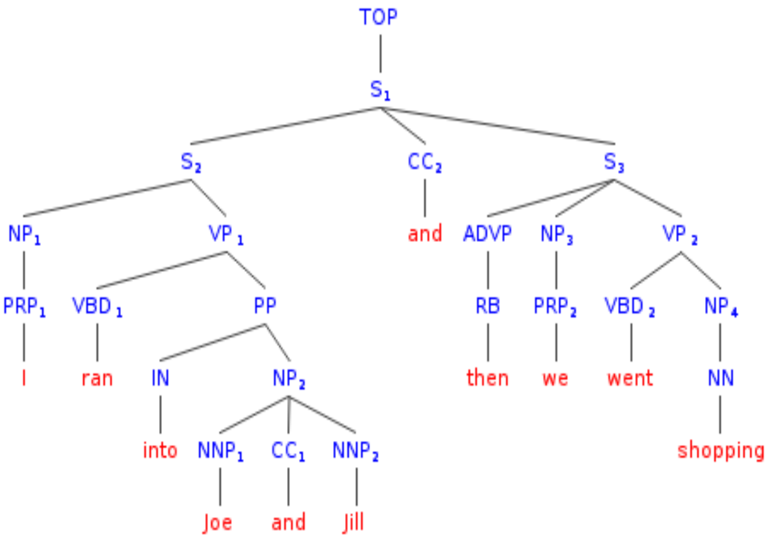
New way of testing linguistic hypothesis

Basis to further improve the model

Related to Linzen et al. (2017), but different questions

Two Ways of Generating Sentences

$P(\boldsymbol{x})$



$P(\boldsymbol{x}, \boldsymbol{y})$

Overview

- Crash course on Recurrent Neural Network Grammars (RNNG)
- Three concrete linguistic questions about what the RNNG learns

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

No. Steps	Stack	String Terminals	Action
0			NT(S)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)
2	(S | (NP		GEN(*the*)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)
2	(S | (NP		GEN(*the*)
3	(S | (NP | *the*	*the*	GEN(*hungry*)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)
2	(S | (NP		GEN(*the*)
3	(S | (NP | *the*	*the*	GEN(*hungry*)
4	(S | (NP | *the* | *hungry*	*the hungry*	GEN(*cat*)

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

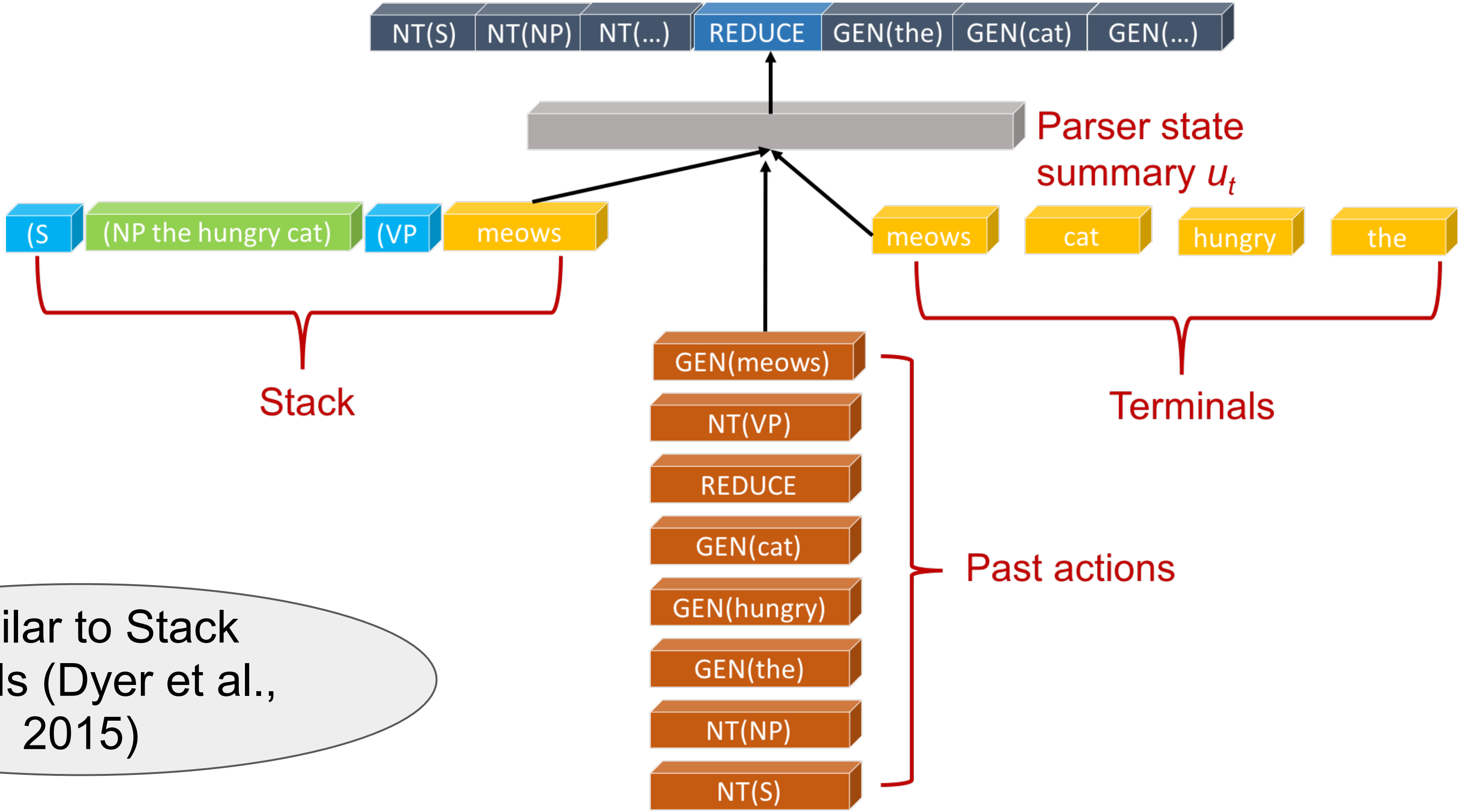
No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)
2	(S | (NP		GEN(*the*)
3	(S | (NP | *the*	*the*	GEN(*hungry*)
4	(S | (NP | *the* | *hungry*	*the hungry*	GEN(*cat*)
5	(S | (NP | *the* | *hungry* | *cat*	*the hungry cat*	REDUCE

RNNGs: Sample Action Sequences

(S (NP the hungry cat) (VP meows) .)

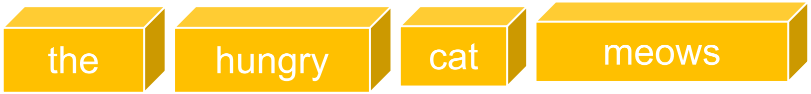
No. Steps	Stack	Terminals	Action
0			NT(S)
1	(S		NT(NP)
2	(S | (NP		GEN(*the*)
3	(S | (NP | *the*	*the*	GEN(*hungry*)
4	(S | (NP | *the* | *hungry*	*the hungry*	GEN(*cat*)
5	(S | (NP | *the* | *hungry* | *cat*	*the hungry cat*	REDUCE
6	(S | (NP *the hungry cat*)	*the hungry cat*	NT(VP)

Model Architecture



Similar to Stack LSTMs (Dyer et al., 2015)

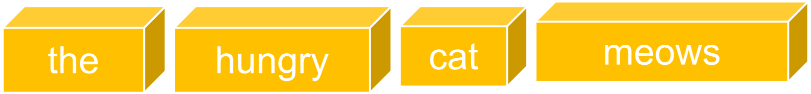
RNNG vs Sequential LSTMs



$P(\boldsymbol{x})$

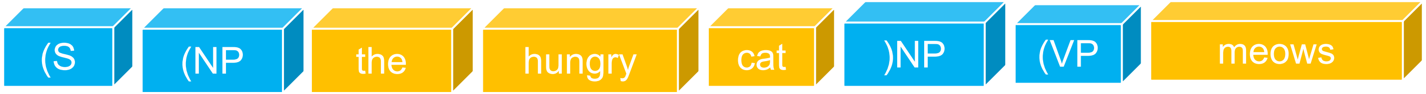
Sequential LSTMs without Syntax

RNNG vs Sequential LSTMs



$P(\boldsymbol{x})$

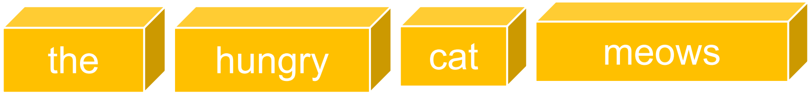
Sequential LSTMs without Syntax



$P(\boldsymbol{x}, \boldsymbol{y})$

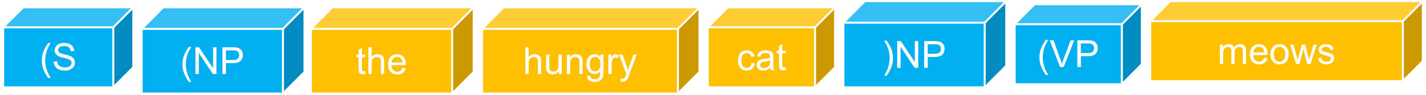
Sequential LSTMs with Syntax (Choe and Charniak, 2016)

RNNG vs Sequential LSTMs



$P(\boldsymbol{x})$

Sequential LSTMs without Syntax



$P(\boldsymbol{x}, \boldsymbol{y})$

Sequential LSTMs with Syntax (Choe and Charniak, 2016)

$P(\boldsymbol{x}, \boldsymbol{y})$

RNNG
(Dyer et al., 2016; this work)

PTB Test Experimental Results

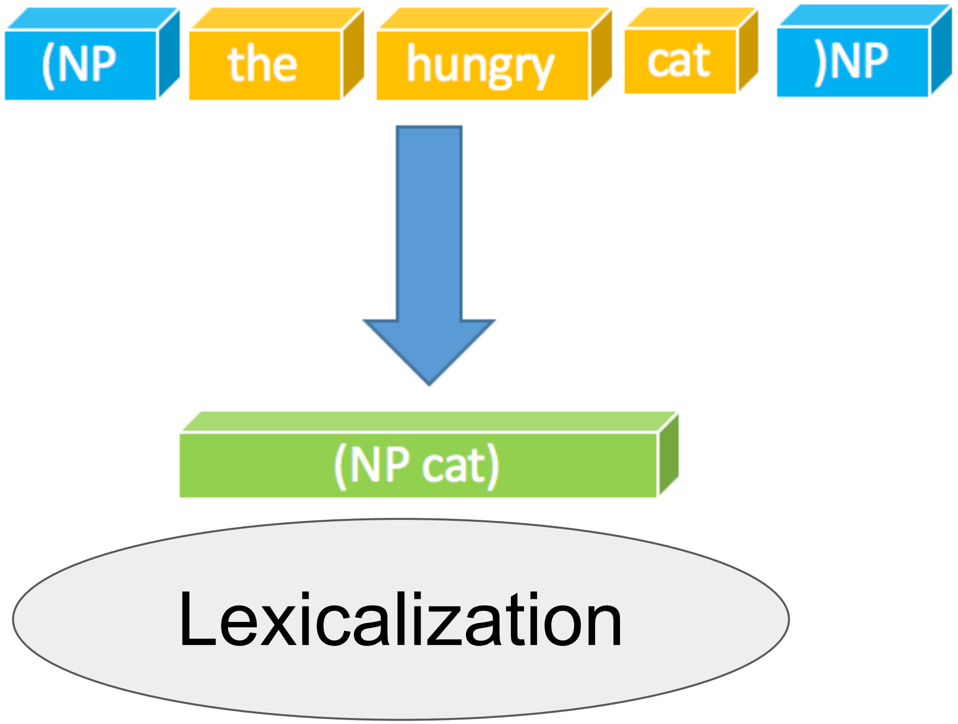
Parsing F1

Model	Parsing F1
Collins (1999)	88.2
Petrov and Klein (2007)	90.1
RNNG	**93.3**
Choe and Charniak (2016) - Supervised	92.6

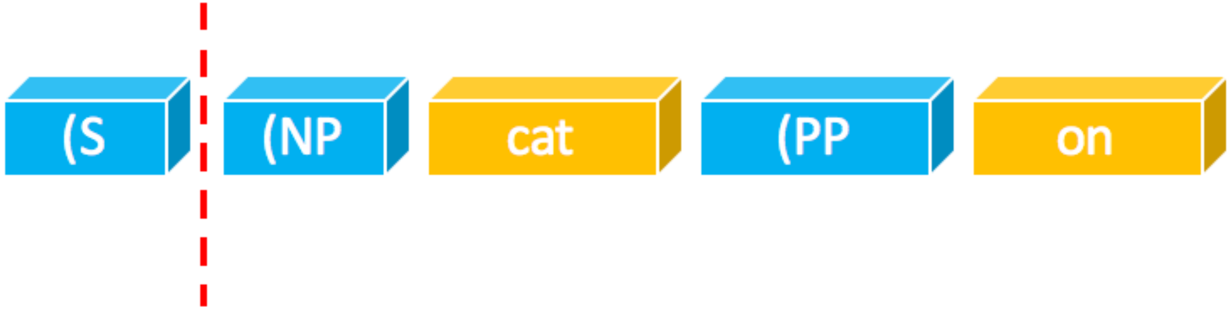
LM Ppl.

Model	LM ppl.
IKN 5-gram	169.3
Sequential LSTM LM	113.4
RNNG	**105.2**

What Can RNNGs Learn?



What Can RNNGs Learn?



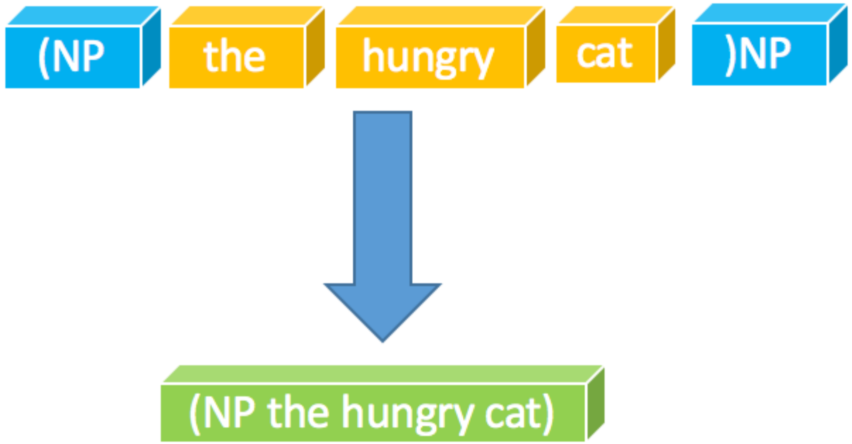
Parent annotations

Question 1

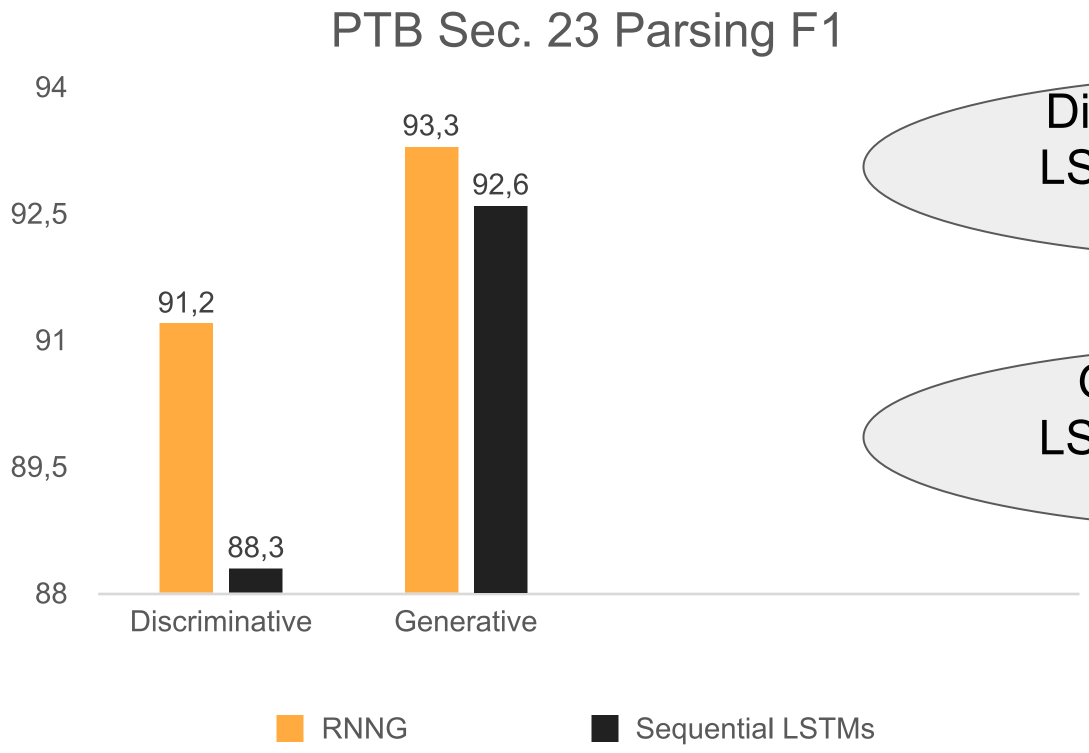
How important is explicit modeling of composition?

Method: Contrast to models that lack composition function

Result: Composition and syntactic recency are key



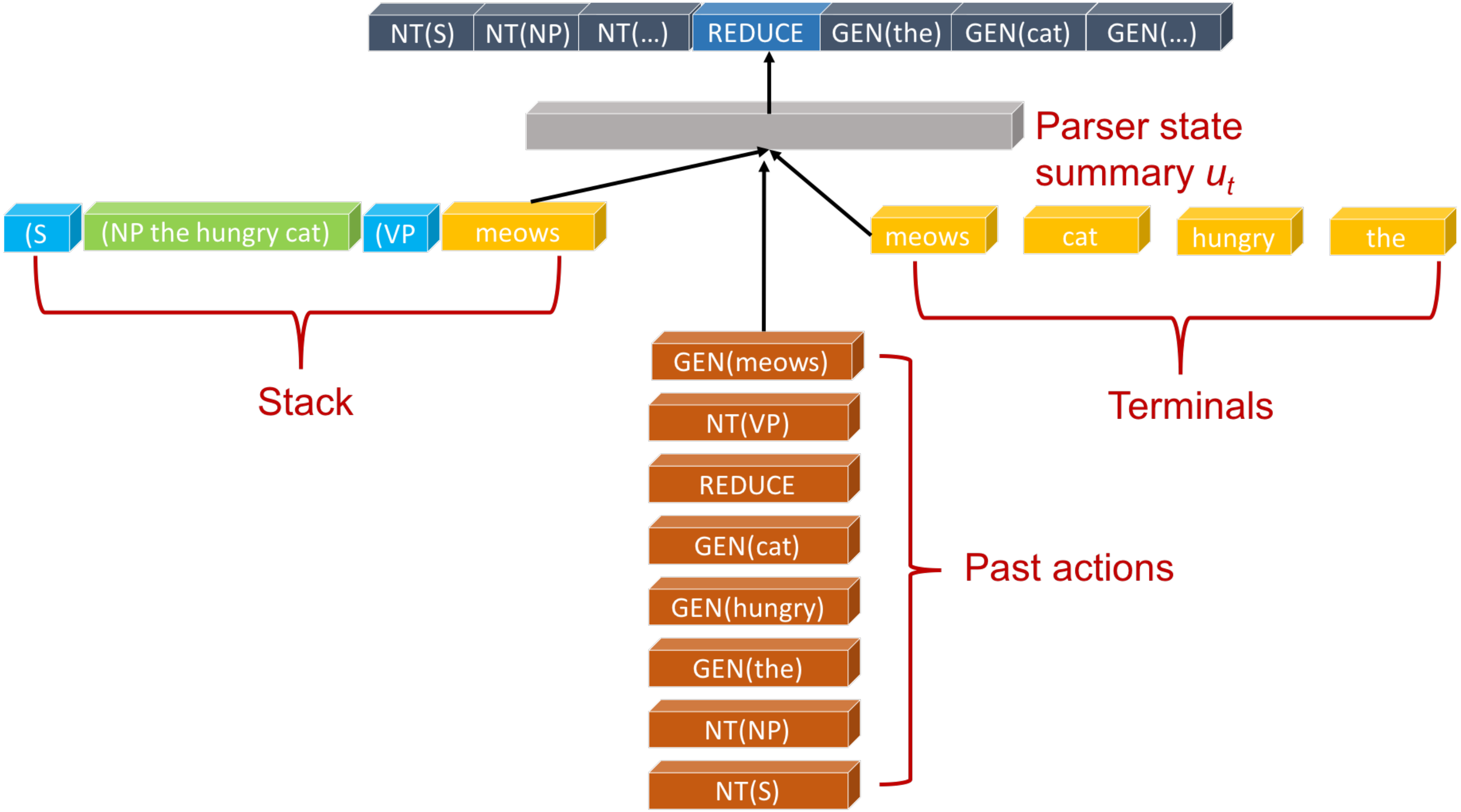
How Important Is Composition?



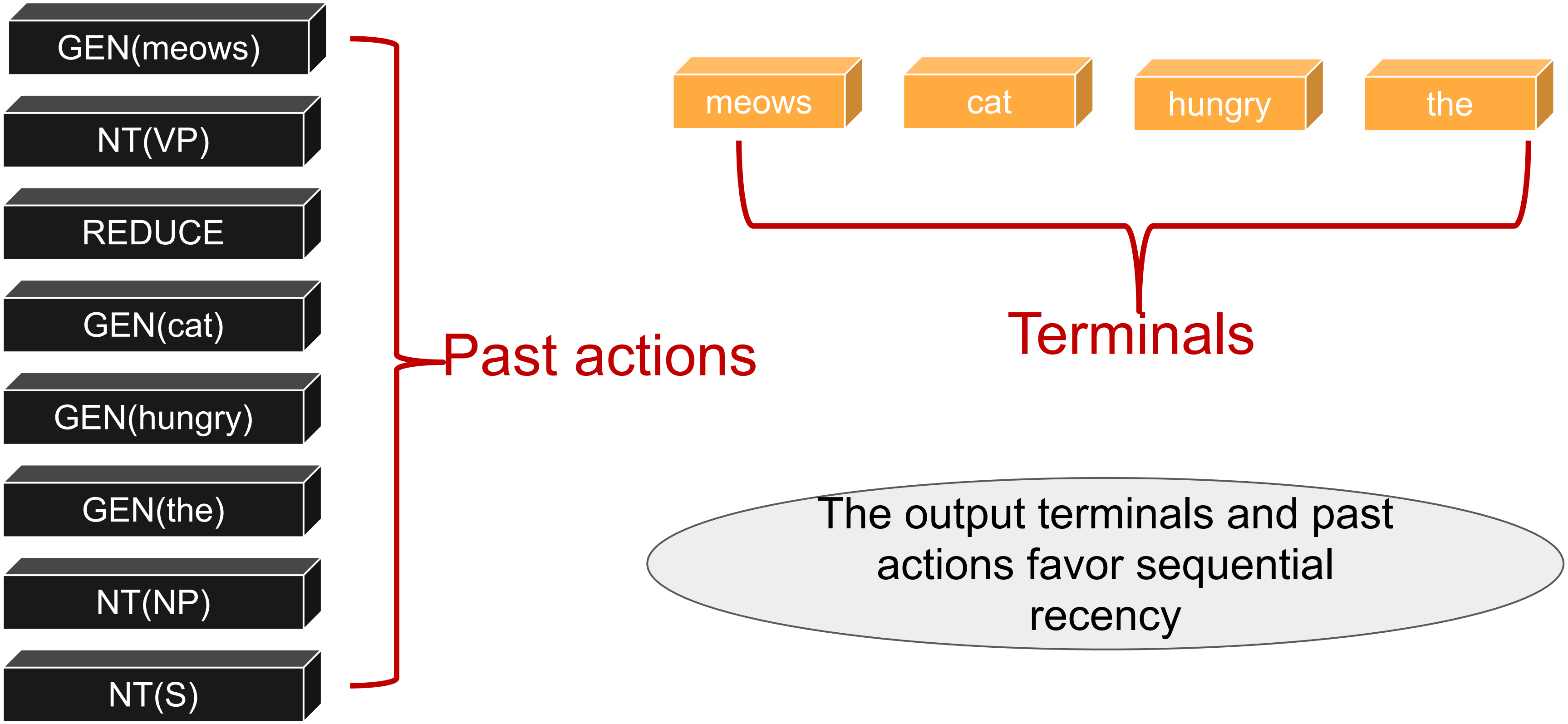
Discriminative sequential LSTM is due to Vinyals et al. (2015)

Generative sequential LSTM is due to Choe and Charniak (2016)

More Evidence that Composition is Key

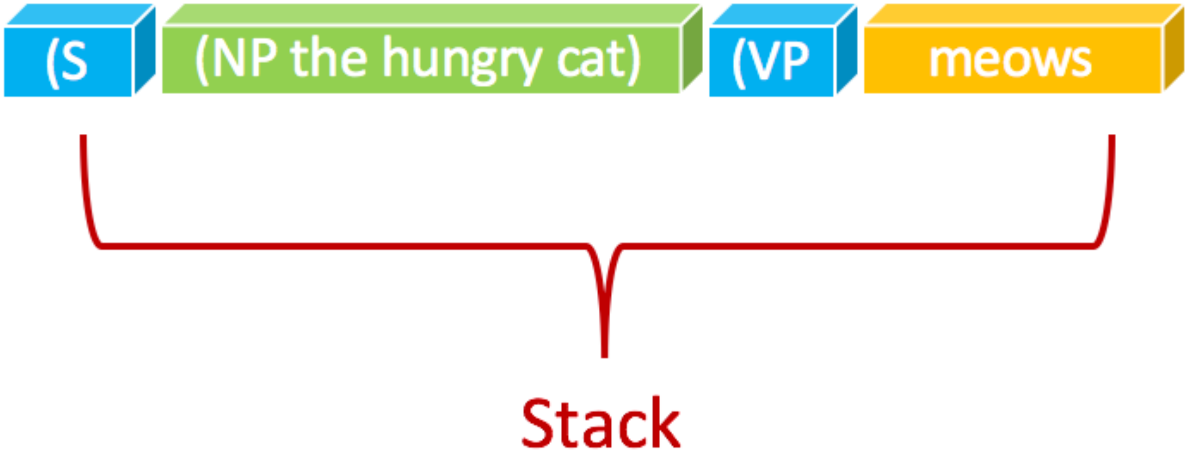


the Output Terminals and Past Actions: **Sequential Recency**



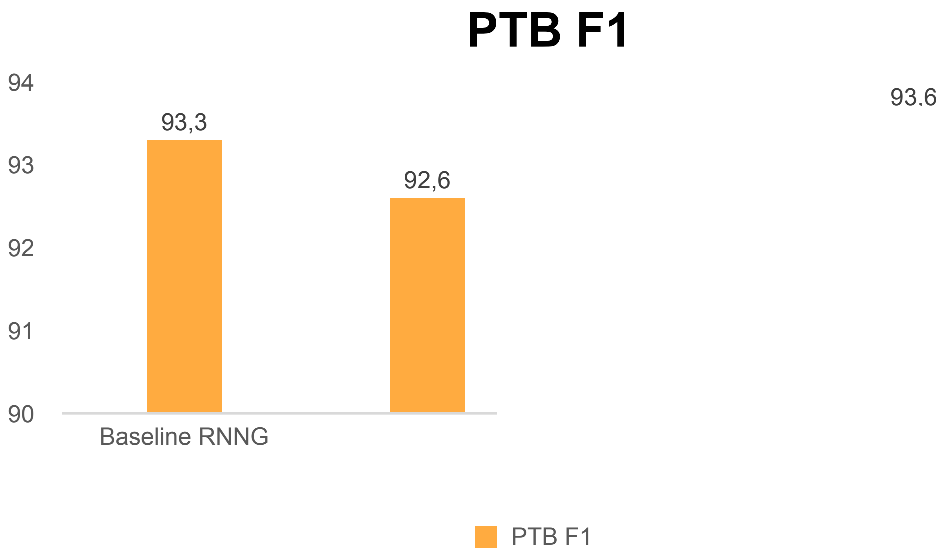
The output terminals and past actions favor sequential recency

Composition and Syntactic Recency

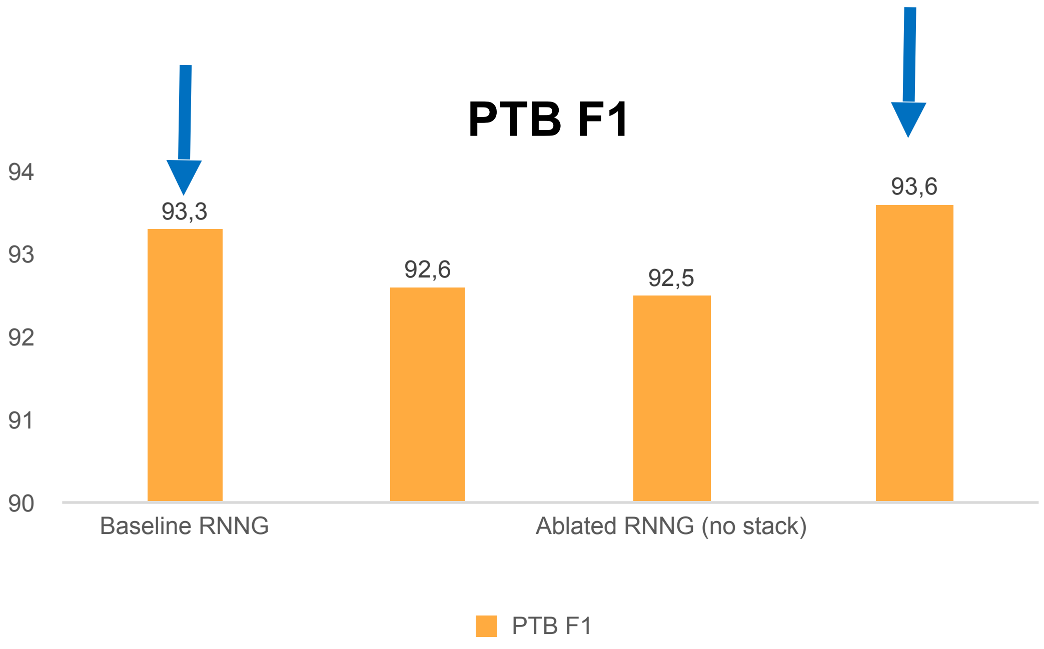


The stack favors syntactic recency

Ablation Results: Parsing F1



Ablation Results: Parsing F1



By conversion, Stack-only RNNG achieves the best results for dependency parsing (**95.8** UAS)

How Important Is Composition?

- The stack (the only element with explicit composition) is most important
- Ablating the stack provides little or no gain over sequential LSTMs in both parsing and language modeling
- RNNG with only a stack outperforms variant configurations

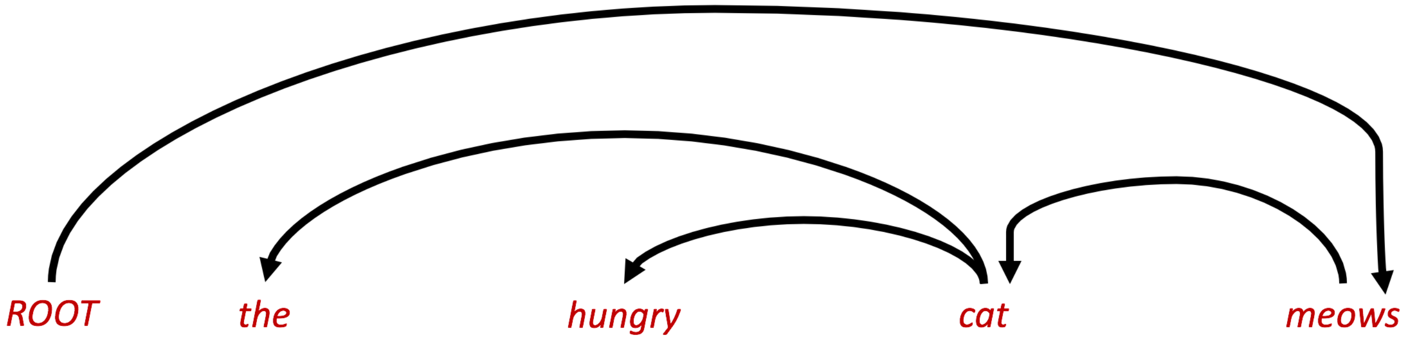
An inductive bias favoring syntactic recency is more important than sequential recency for parsing and language modeling!

Question 2

Does the model discover headedness?

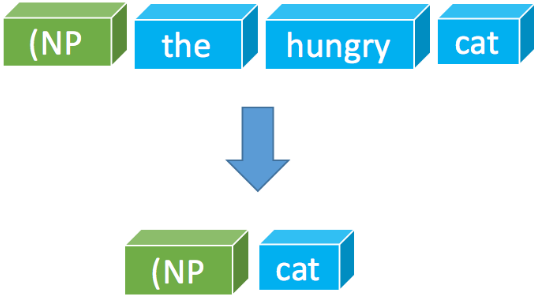
Method: New interpretable attention-based composition function

Result: sort of

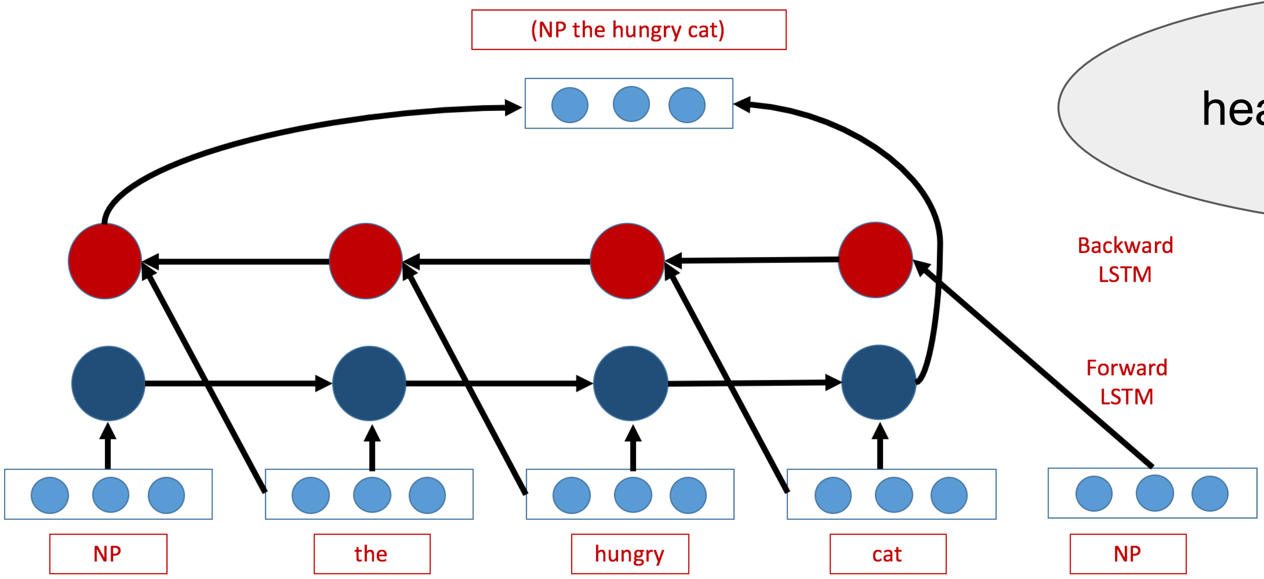


Headedness

- Linguistic theories of phrasal representation involve a strongly privileged lexical head that determines the whole representation
- Hypothesis for single lexical heads (Chomsky, 1993) and multiple ones for tricky cases (Jackendoff 1977; Keenan 1987)
- Heads are crucial as features in non-neural parsers, starting with Collins (1997)



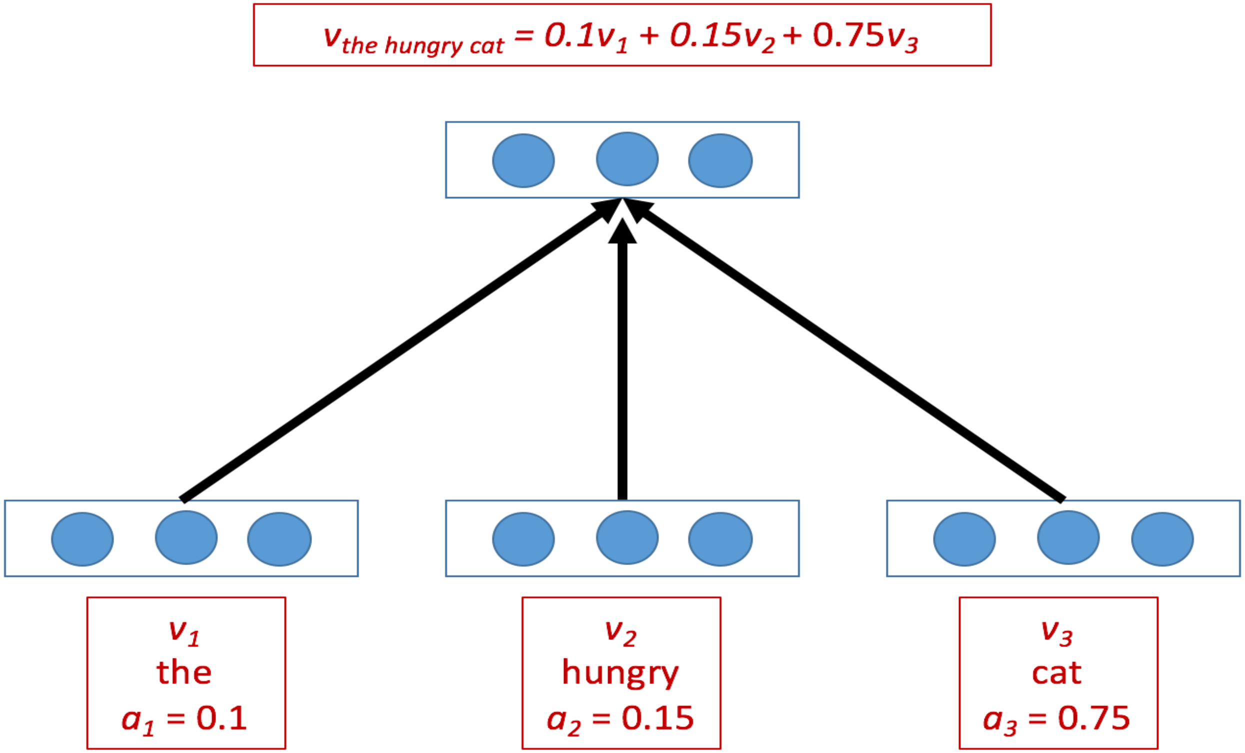
RNNG Composition Function



Hard to detect headedness in sequential LSTMs

Use “attention” in sequence-to-sequence model (Bahdanau et al., 2014)

Key Idea of Attention



Experimental Results: PTB Test Section

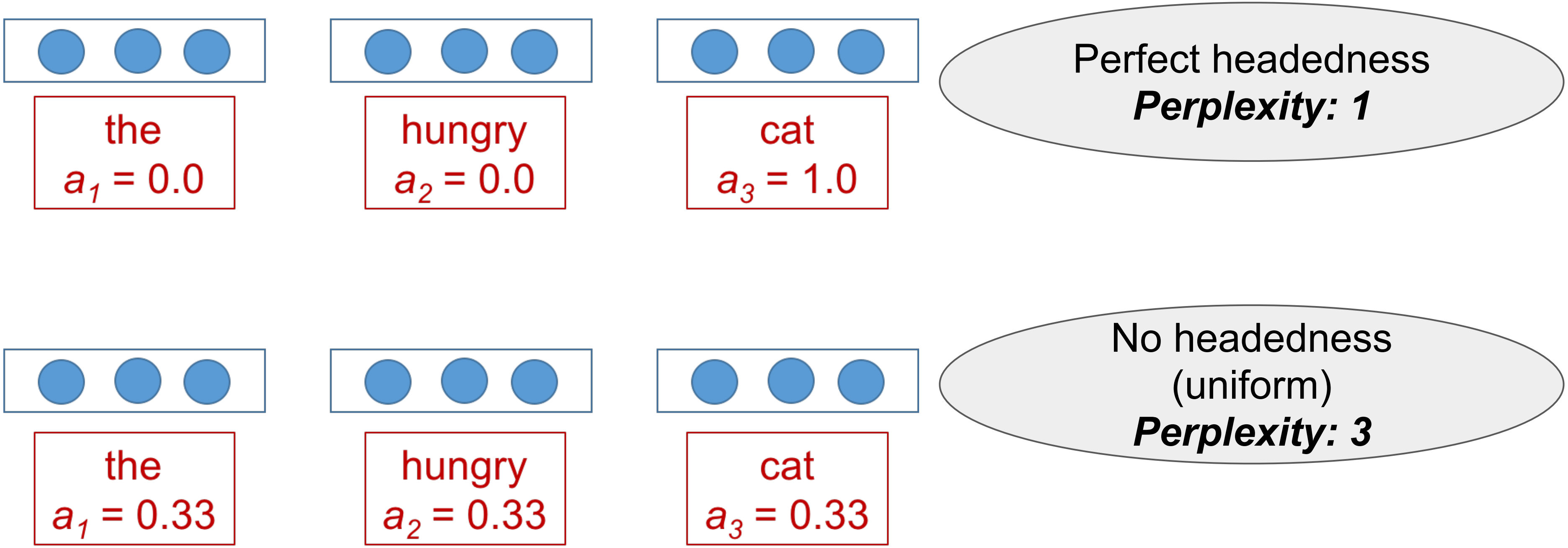
Parsing F1

Model	Parsing F1
Baseline RNNG	93.3
Stack-only RNNG	**93.6**
Gated-Attention RNNG (stack-only)	93.5

LM Ppl.

Model	LM Ppl.
Sequential LSTM	113.4
Baseline RNNG	105.2
Stack-only RNNG	101.2
Gated-Attention RNNG (stack-only)	**100.9**

Two Extreme Cases of Attention



Learned Attention Vectors

Noun Phrases
the (0.0) final (0.18) **hour (0.81)**
their (0.0) first (0.23) **test (0.77)**
Apple (0.62) , (0.02) Compaq (0.1) and (0.01) IBM (0.25)
NP (0.01) , (0.0) **and (0.98)** NP (0.01)

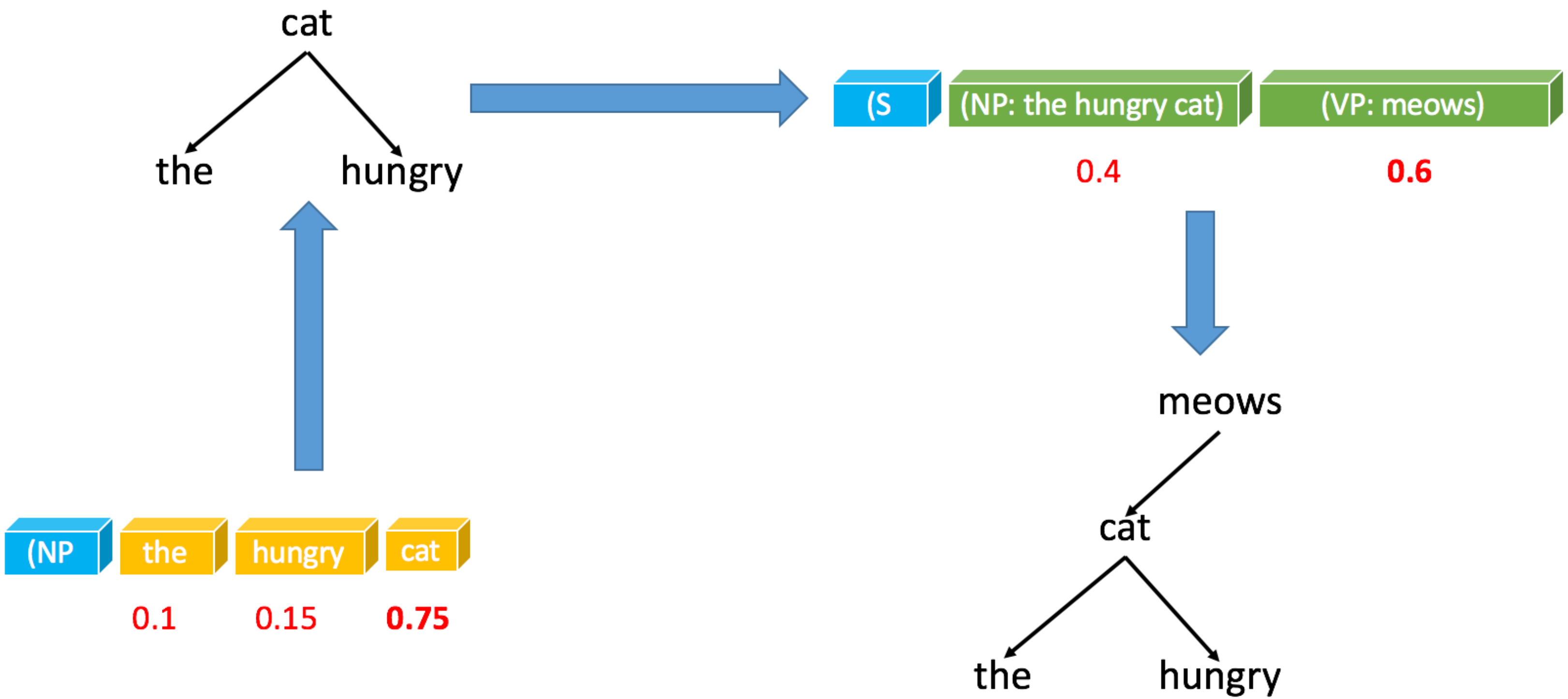
Learned Attention Vectors

Verb Phrases
to (0.99) VP (0.01)
did (0.39) **n't (0.60)** VP (0.01)
handle (0.09) **NP (0.91)**
VP (0.15) **and (0.83)** VP (0.02)

Learned Attention Vectors

Prepositional Phrases
of (0.97) NP (0.03)
in (0.93) NP (0.07)
by (0.96) S (0.04)
NP (0.1) **after** (0.83) NP (0.06)

Quantifying the Overlap with Head Rules



Quantifying the Overlap with Head Rules

Reference	UAS
Random baseline	~28.6
Collins head rules	49.8
Stanford head rules	40.4

Question 3

What is the role of nonterminal labels?

Method: Ablate the nonterminal label categories from the data

Result: Nonterminal labels add very little

Nonterminal Ablation

(S (NP the hungry cat) (VP meows) .)

(X (X the hungry cat) (X meows) .)

Quantitative Results

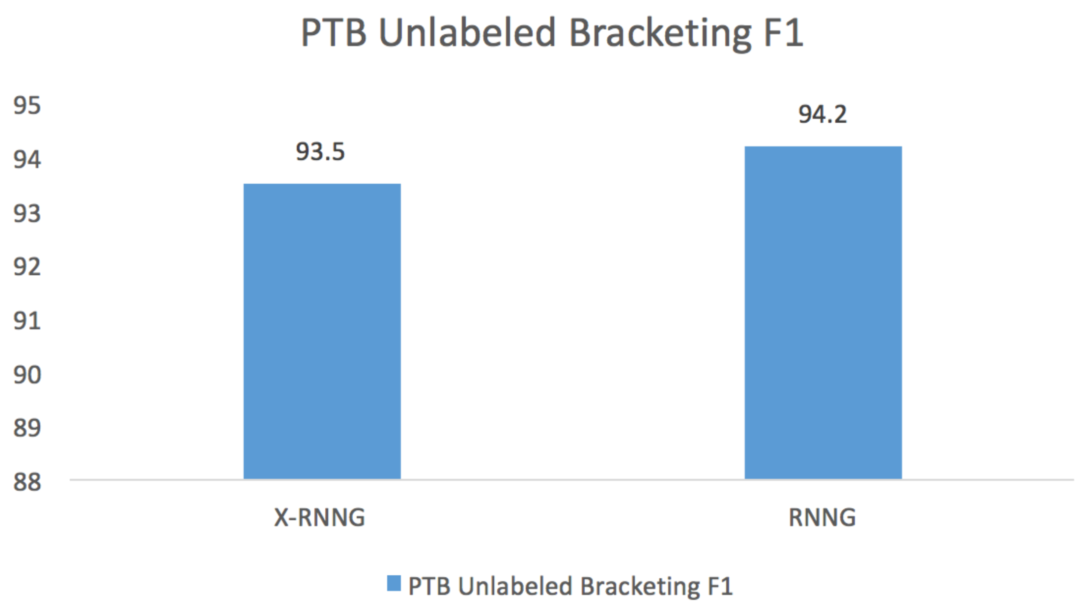
Gold: (X (X the hungry cat) (X meows) .)

Predicted: (X (X the hungry) (X cat meows) .)

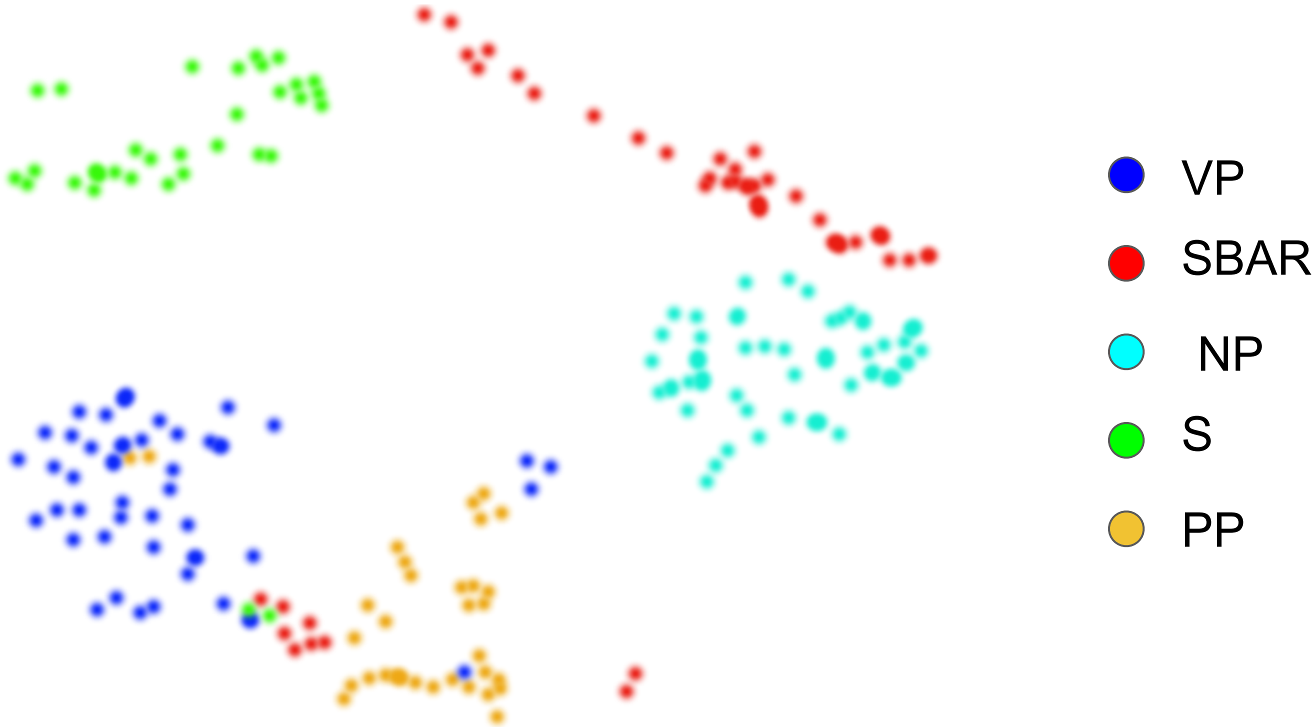
Quantitative Results

Gold: (X (X the hungry cat) (X meows) .)

Predicted: (X (X the hungry) (X cat meows) .)



Visualization



Conclusion

- Composition is important (the inductive bias of syntactic recency is beneficial for parsing and language modeling)

    It helps the model do better quantitatively

    It helps us analyze the model to the extent that we did

- RNNG learns (imperfect) headedness, which is both similar and distinct to linguistic theories
- RNNG is able to rediscover nonterminal information given weak bracketing structures, and also make nontrivial semantic distinctions

Why Are RNNGs Better than RNNs?

- Composition is key
- Composition is picking out heads
- Syntactic recency is a good bias for modeling language