LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom

DeepMind

Motivation

Language exhibits **hierarchical** structure

[[The cat [that he adopted]] [sleeps]]

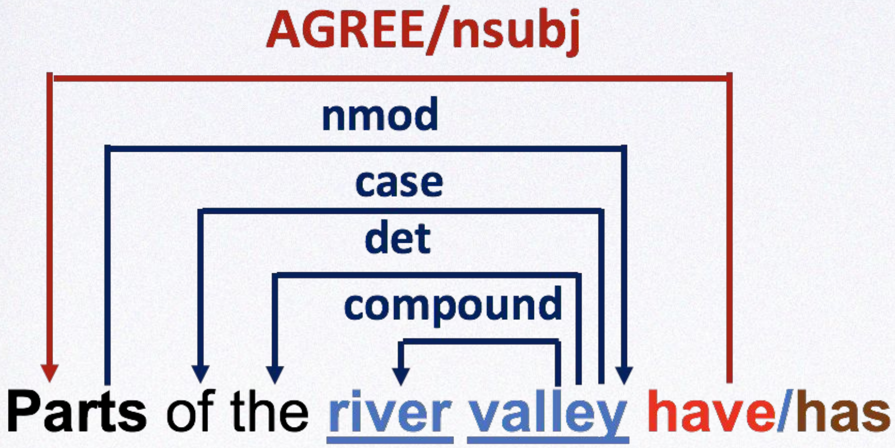
...... but LSTMs work so well without explicit notions of structure.

Number Agreement

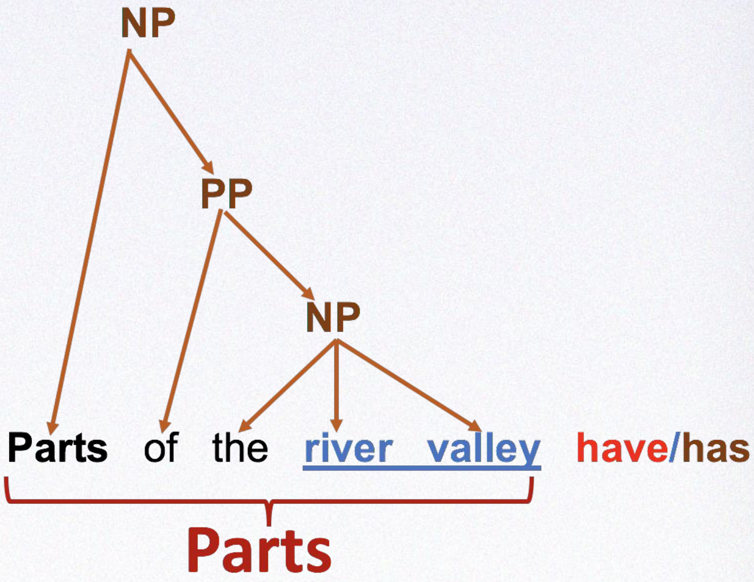
Number agreement example with **two** attractors (Linzen et al., 2016)

Number agreement is a cognitively-motivated probe to distinguish **hierarchical** theories from purely sequential ones.

Number Agreement is Sensitive to Syntactic Structure



Number agreement reflects the **dependency** relation between subjects and verbs



Models that can capture **headedness** should do better at number agreement

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Overview

- Revisit the prior work of Linzen et al. (2016) that argues LSTMs trained on language modelling objectives fail to learn such dependencies.
- Investigate whether models that explicitly incorporate syntactic structure can do better, and *how* syntactic information should be encoded.
- Demonstrate that *how* the structure is built affects number agreement generalisation.

Number Agreement Dataset Overview

	Train	Test
Sentences	141,948	1,211,080
Types	10,025	10,025
Tokens	3,159,622	26,512,851

Number agreement dataset is derived from dependency-parsed Wikipedia corpus

All intervening nouns must be of the same number

Number Agreement Dataset Overview

AGREE

Parts of the river valley **have/has**

All intervening nouns must be of the same number

The vast majority of number agreement dependencies are **sequential**

# Attractors	# Instances	% Instances
n=0	1,146,330	94.7%
n=1	52,599	4.3%
n=2	9,380	0.77%
n=3	2,051	0.17%
n=4	561	0.05%
n=5	159	0.01%

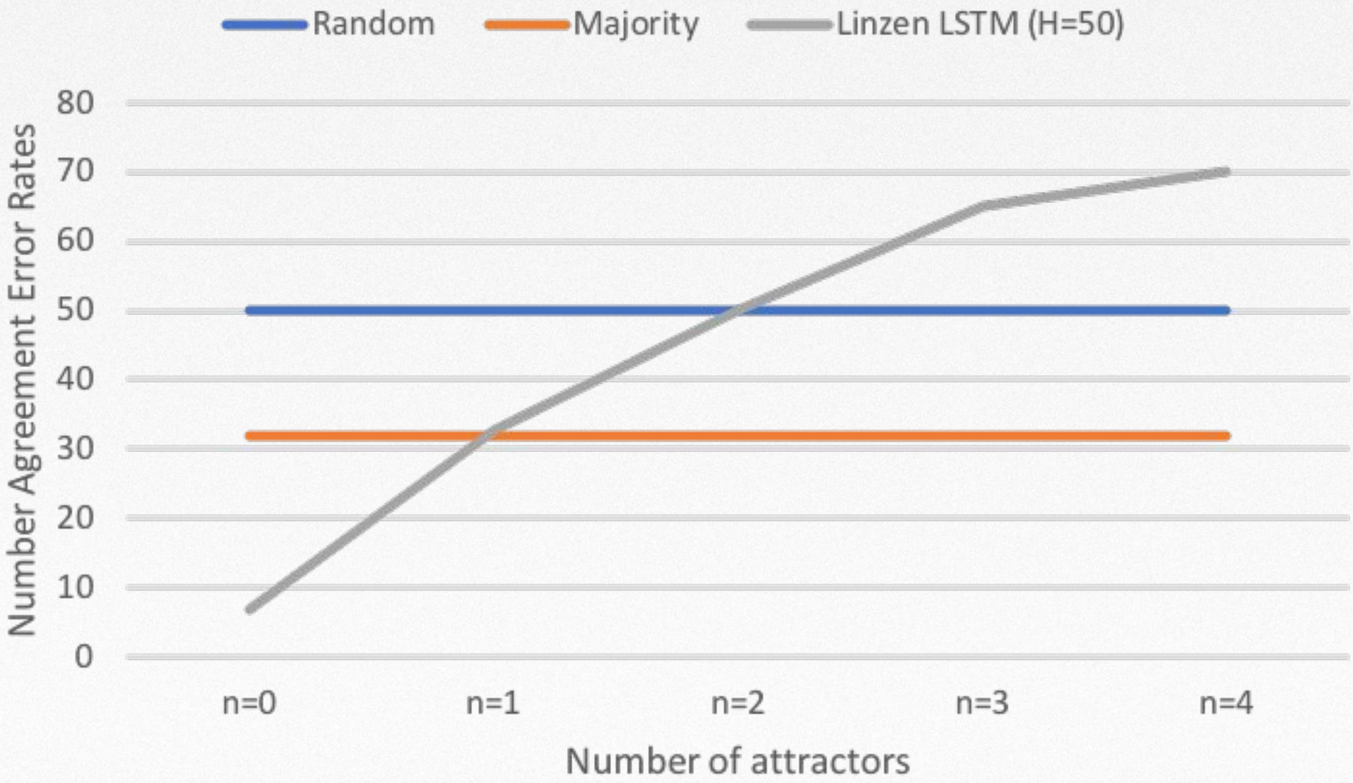
First Part: Can LSTMs Learn Number Agreement Well?

The model is trained with **language modelling** objectives

Revisit the same question as Linzen et al. (2016):

To what extent are LSTMs able to learn **non-local syntax-sensitive dependencies** in natural language?

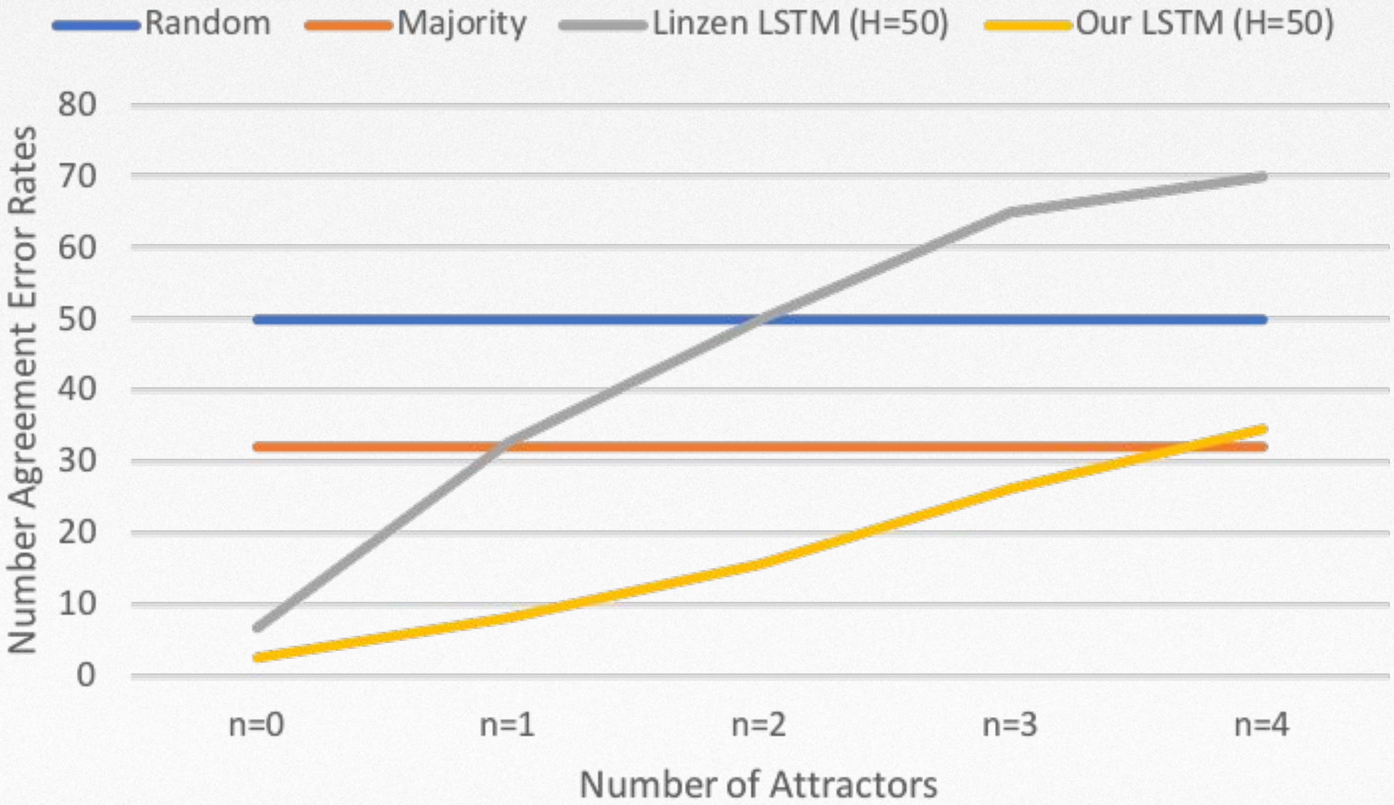
Linzen et al. LSTM Number Agreement Error Rates



Lower is better

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Small LSTM Number Agreement Error Rates

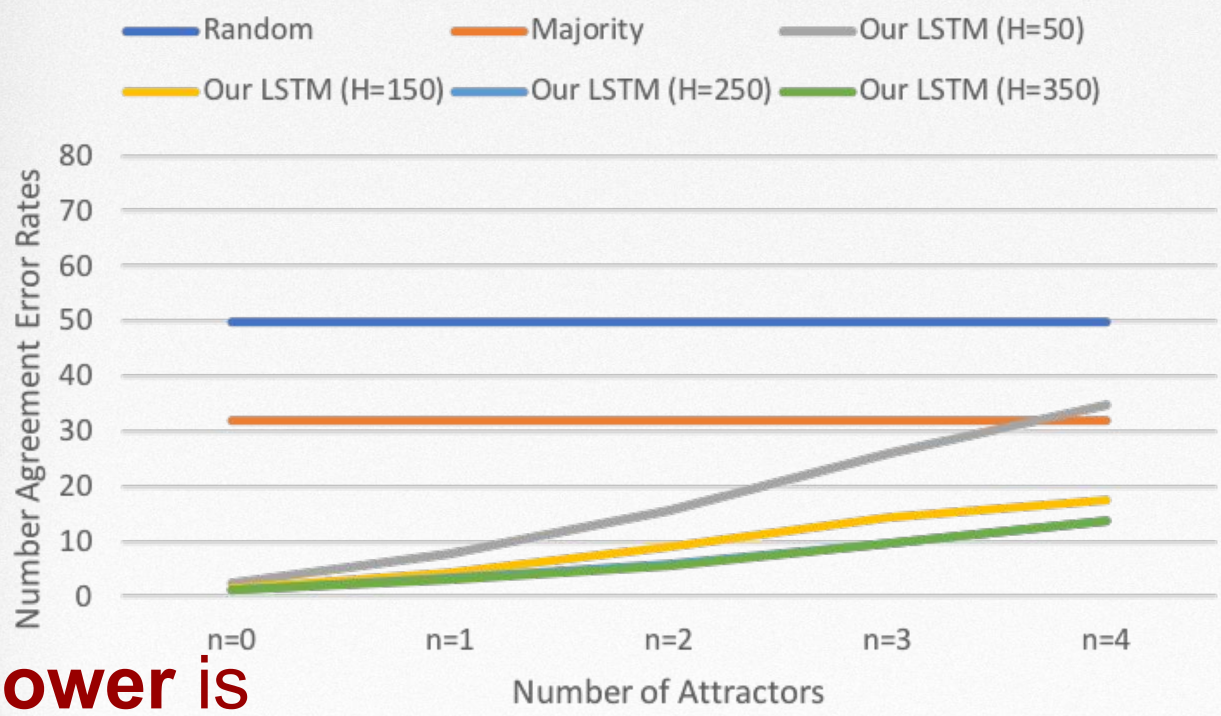


Lower is better

DeepMind

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Larger LSTM Number Agreement Error Rates



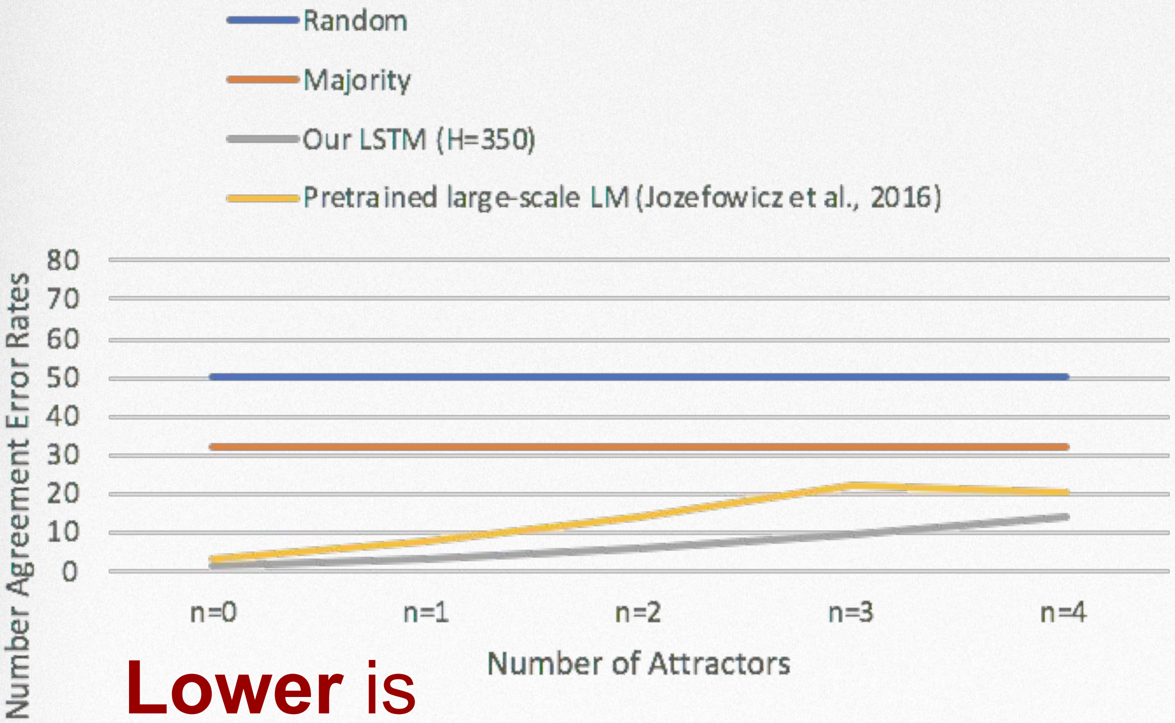
Capacity matters for capturing non-local structural dependencies

Despite this, relatively **minor** perplexity difference (~10%) between H=50 and H=150

Lower is better

DeepMind

LSTM Number Agreement Error Rates



Lower is better

Capacity and size of training corpus are **not** the full story

Domain and training settings matter too

Can Character LSTMs Learn Number Agreement Well?

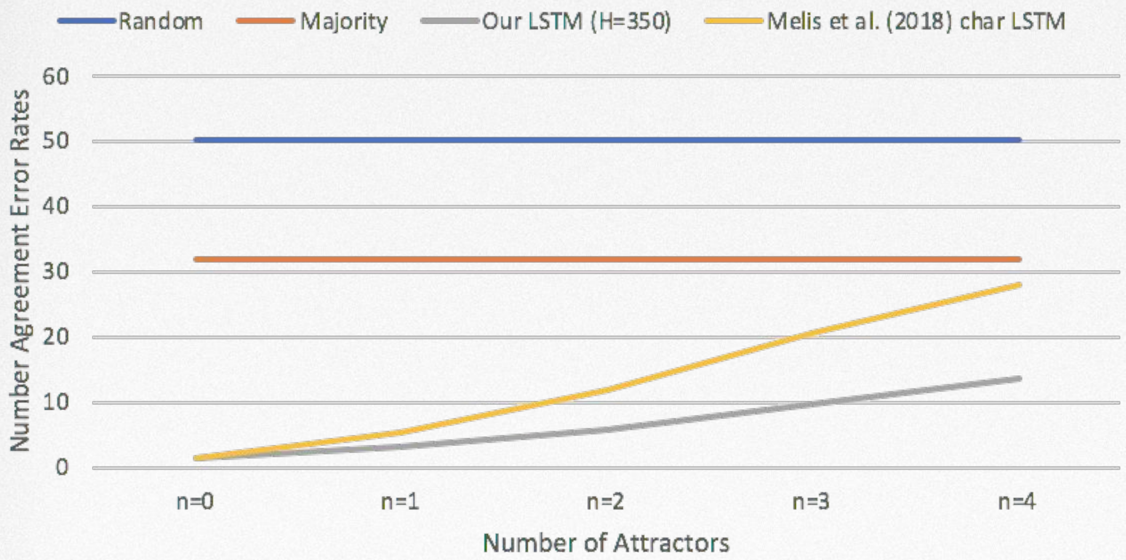
AGREE

P a r t s <space> … r i v e r <space> v a l l e y <space> h a v e / h a s

Character LSTMs have been used in various tasks, including machine translation, language modelling, and many others.

+ It is easier to exploit **morphological cues.**
- Model has to resolve dependencies between **sequences** of tokens.
- The sequential dependencies are **much longer**.

Character LSTM Agreement Error Rates



State-of-the-art character LSTM (Melis et al., 2018) model on Hutter Prize, with **27M** parameters.

Trained, validated, and tested on the same data.

Lower is better

Strong character LSTM model performs **much worse** for multiple attractor cases

Consistent with earlier work (Sennrich, 2017) and potential avenue for improvement

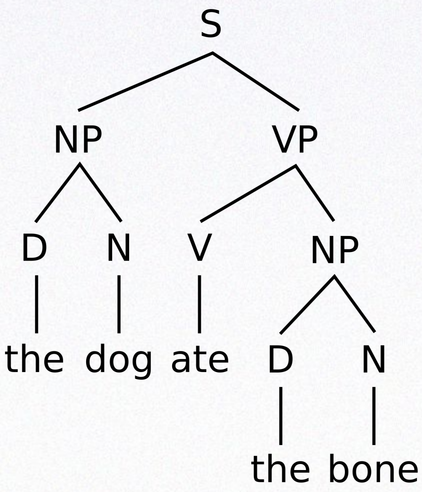
First Part Quick Recap

- LSTM language models are able to learn number agreement to a much larger extent than suggested by earlier work.
  - Independently confirmed by Gulordava et al. (2018).
  - We further identify model capacity as one of the reasons for the discrepancy.
  - Model tuning is important.

- A strong character LSTM language model performs much worse for number agreement with multiple attractors.

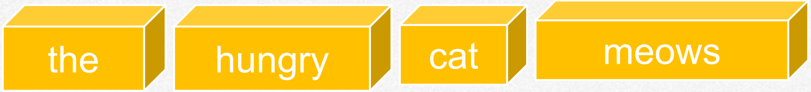
Two Ways of Modelling Sentences

$P(\boldsymbol{x})$



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

Three Concrete Alternatives for Modeling Sentences

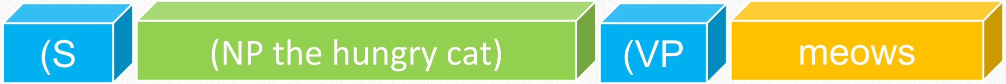


$P(\boldsymbol{x})$

Sequential LSTMs without Syntax

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

Sequential LSTMs with Syntax (Choe and Charniak, 2016)



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

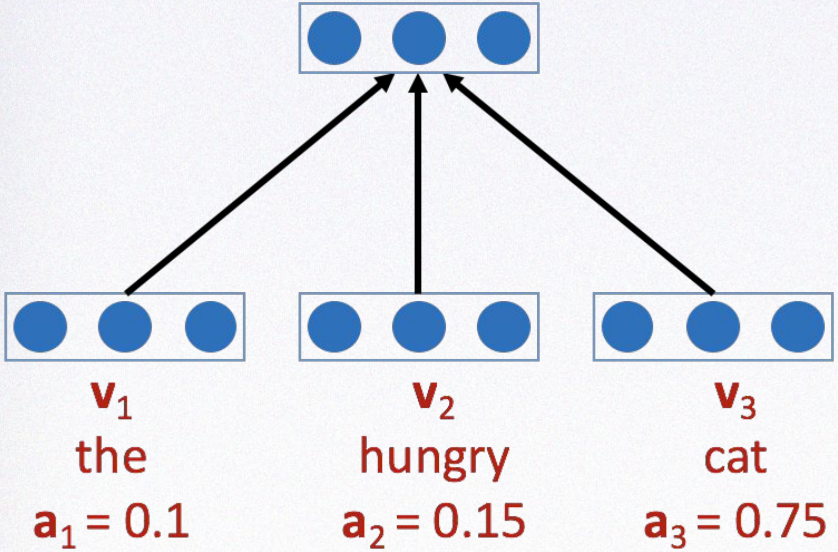
RNNG (Dyer et al., 2016)

Hierarchical inductive bias

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Evidence of Headedness in the Composition Function

$$\mathbf{v}_{\text{the hungry cat}} = 0.1\mathbf{v}_1 + 0.15\mathbf{v}_2 + 0.75\mathbf{v}_3$$



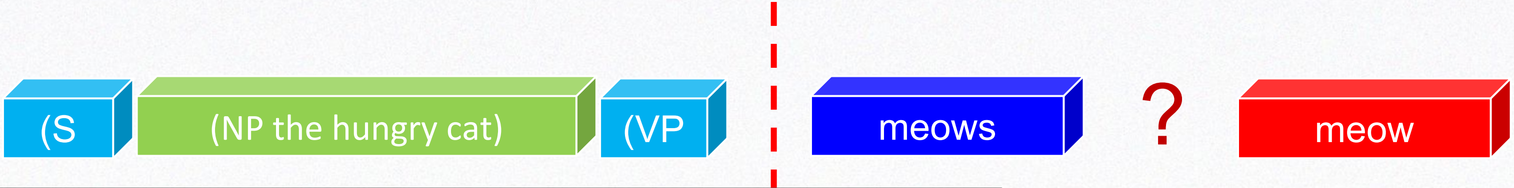
Kuncoro et al. (2017) found evidence of **syntactic headedness** in RNNGs (Dyer et al., 2016)

The discovery of syntactic heads would be useful for number agreement

Inspection of composed representation through the **attention weights**

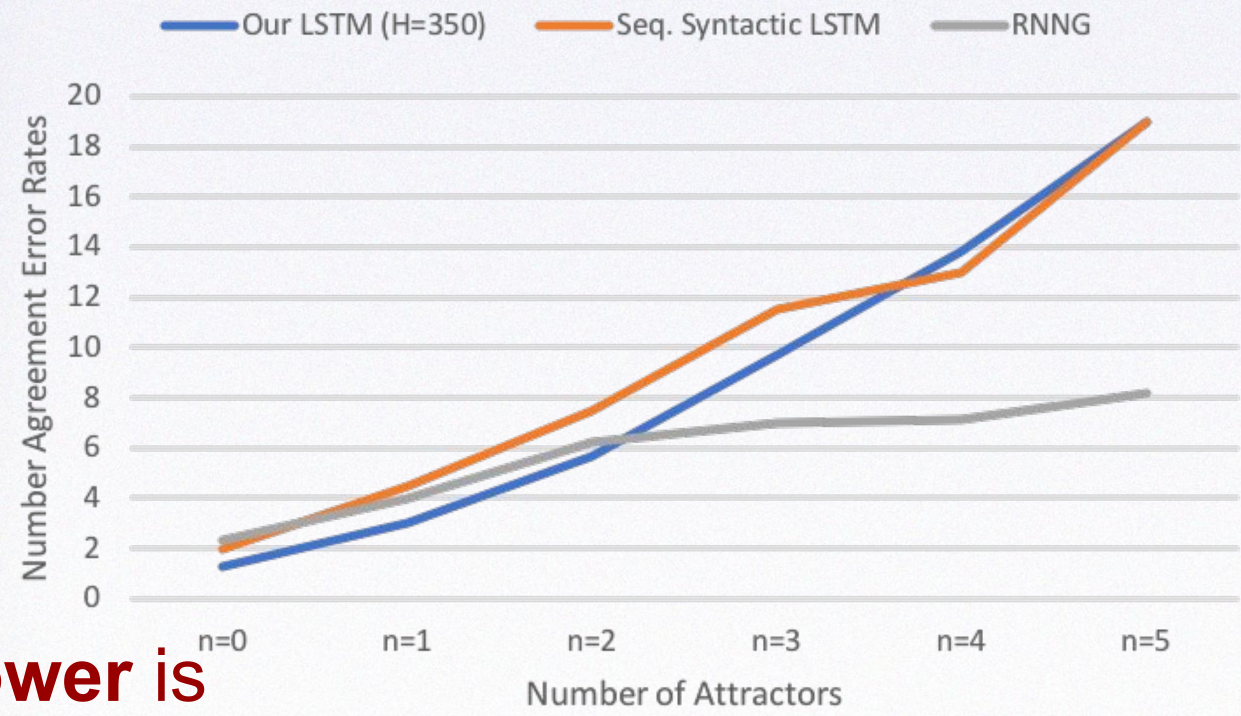
Experimental Settings

- All models are trained, validated, and tested on the same dataset.
- On the training split, the syntactic models are trained using predicted phrase-structure trees from the Stanford parser.
- At test time, we run the incremental beam search (Stern et al., 2017) procedure **up to the main verb** for both verb forms, and take the **highest-scoring tree**.



The most probable tree might potentially be **different** for the correct/incorrect verbs

Experimental Findings



50% error rate reductions for *n=4* and *n=5*

Performance differences are **significant** ($p < 0.05$)

Lower is better

DeepMind

Perplexity

	Dev ppl.
LSTM LM	**72.6**
Seq. Syntactic LSTM	79.2
RNNGs	77.9

Perplexity for syntactic models are obtained with importance sampling (Dyer et al., 2016)

LSTM LM has the best perplexity despite **worse** number agreement performance

Further Remarks: Confound in the Dataset

LSTM language models largely succeed in number agreement

- In around **80%** of cases with multiple attractors, the agreement controller coincides with the **first noun**.

Key question: How do LSTMs succeed in this task?

Identifying the syntactic structure	Memorising the first noun

Kuncoro et al., L2HM 2018

Control Condition Experiments for LSTM LM

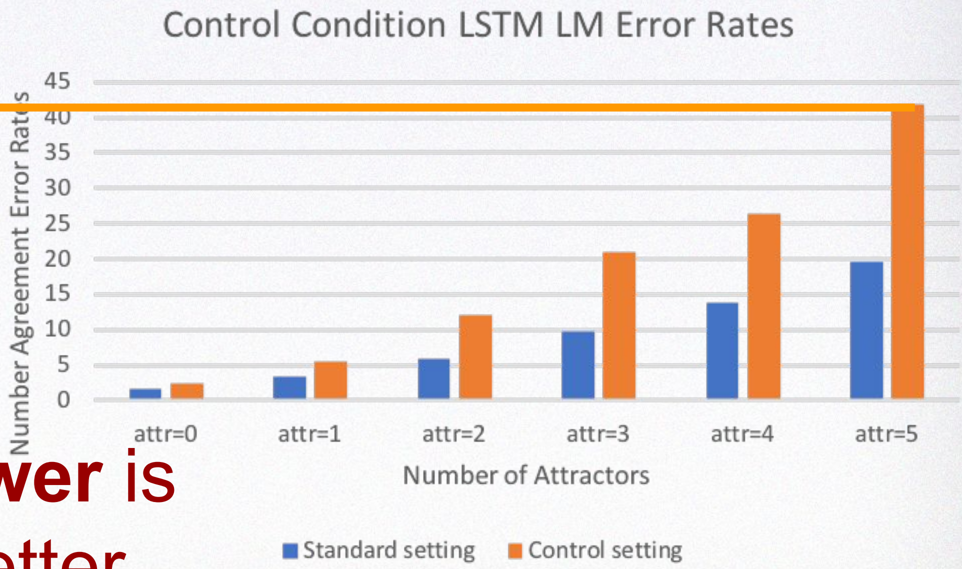
DISAGREE

AGREE

The **scientist** thinks that **[parts** of the river valley **have**/**has]**

Control condition

Control condition **breaks** the correlation between the first noun and agreement controller



Confounded by **first nouns**

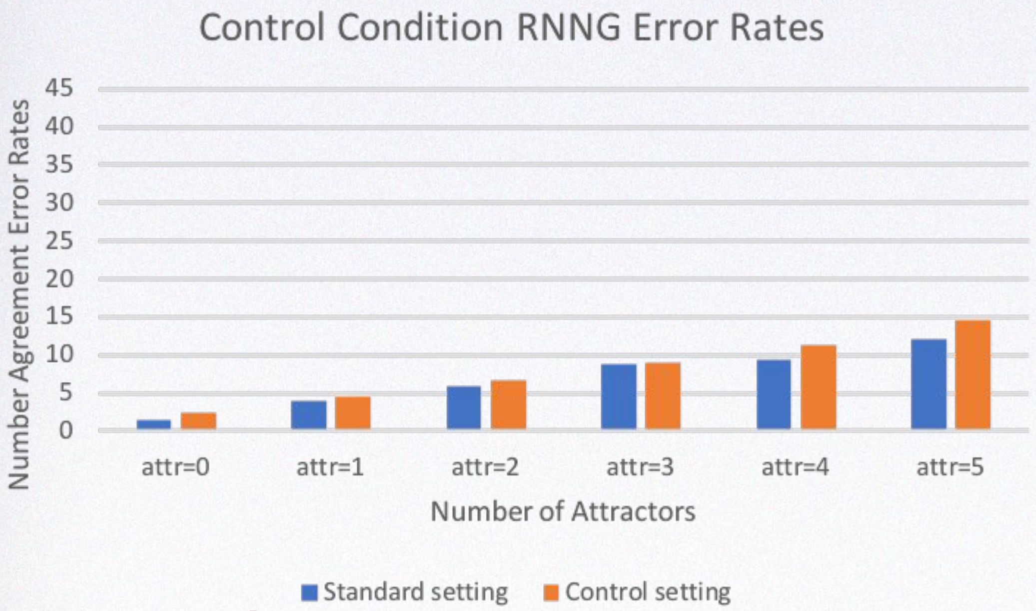
Much less likely to affect human experiments

Lower is better

DeepMind

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Control Condition Experiments for RNNG



Lower is better

Same **y-axis** scale as LSTM LM

- Control for cues that artificial learners can exploit in a cognitive task.
- Adversarial evaluation can better distinguish between models with **correct generalisation** and those that overfit to surface cues.

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Related Work

- Augmenting our models with a hierarchical inductive bias is **not** the only way to achieve better number agreement.

- Another alternative is to make relevant past information **more salient**, such as through memory architectures or attention mechanism.
    - Yogatama et al. (2018) found that **both** attention mechanism and memory architectures outperform standard LSTMs.
    - They found that a model with a **stack-structured memory** performs best, also demonstrating that a **hierarchical, nested** inductive bias is important for capturing syntactic dependencies.

Second Part Quick Recap

- RNNGs **considerably outperform** LSTM language model and sequential syntactic LSTM for number agreement with multiple attractors.
  - Syntactic annotation alone has little impact on number agreement accuracy.
  - RNNGs' success is due to the hierarchical inductive bias.
  - The RNNGs' performance is a new **state of the art** on this dataset (previous best from Yogatama et al. (2018) for n=5 is 88.0% vs **91.8%**)

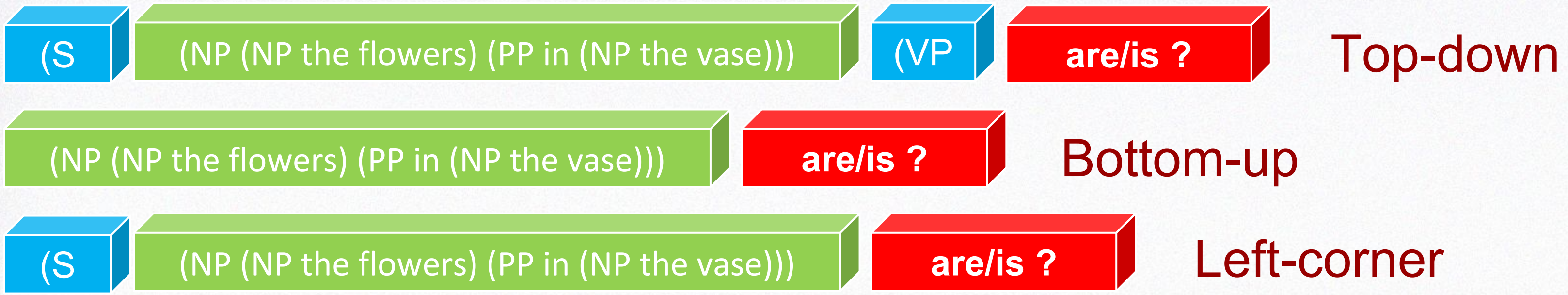
- Perplexity is only **loosely correlated** with number agreement.
  - Independently confirm the finding of Tran et al. (2018).

Different Tree Traversals

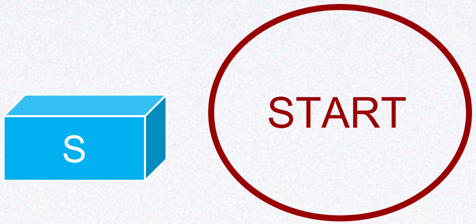
RNNGs operate according to a **top-down, left-to-right** traversal

Here we propose two alternative tree construction orders for RNNGs: **left-corner** and **bottom-up** traversals.

x: the **flowers** in the vase **are/is** [blooming]

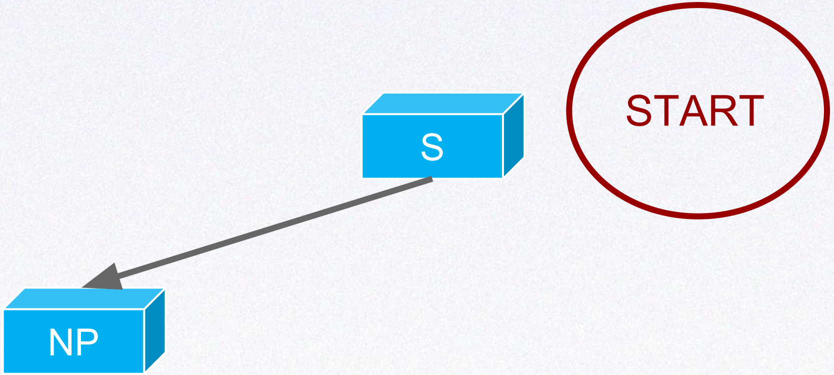


Quick Illustration of the Differences: Top-Down



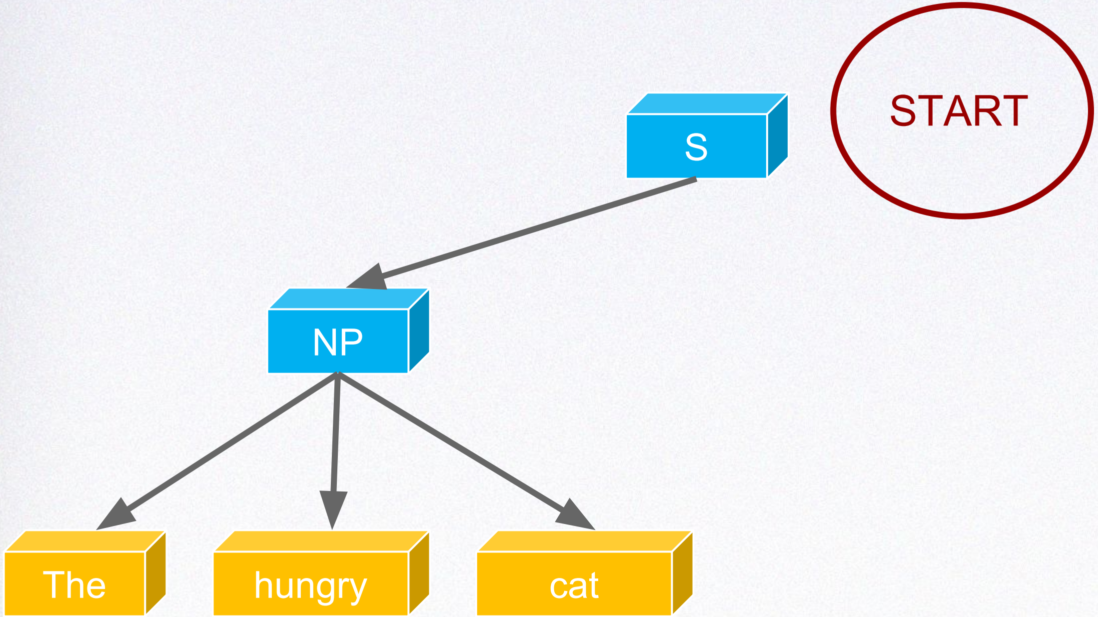
TOP-DOWN

Quick Illustration of the Differences: Top-Down



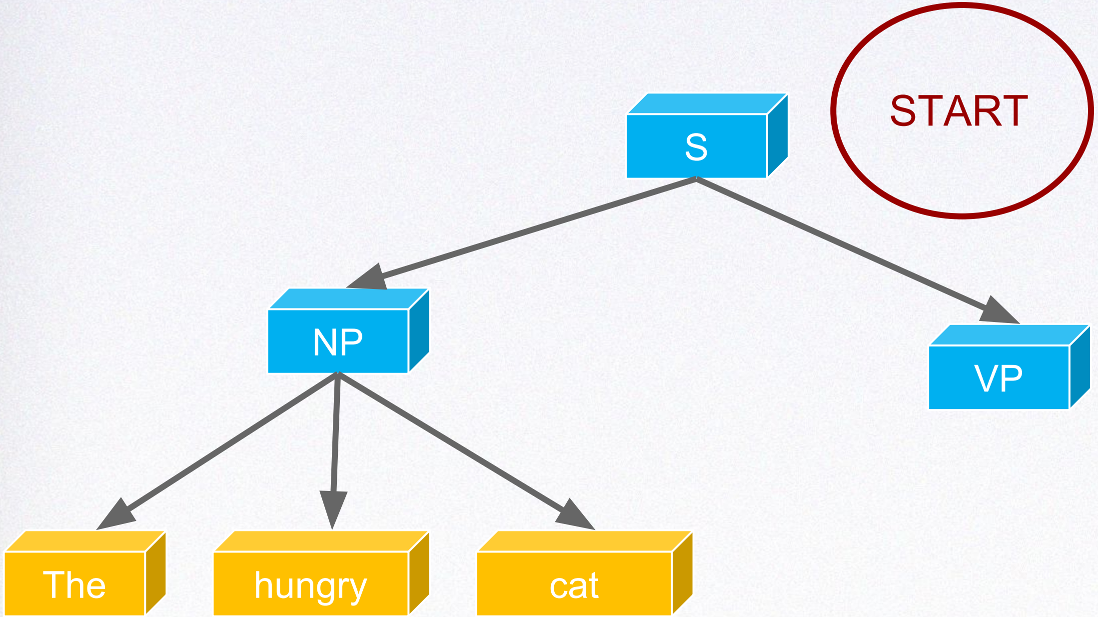
TOP-DOWN

Quick Illustration of the Differences: Top-Down



TOP-DOWN

Quick Illustration of the Differences: Top-Down



TOP-DOWN

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

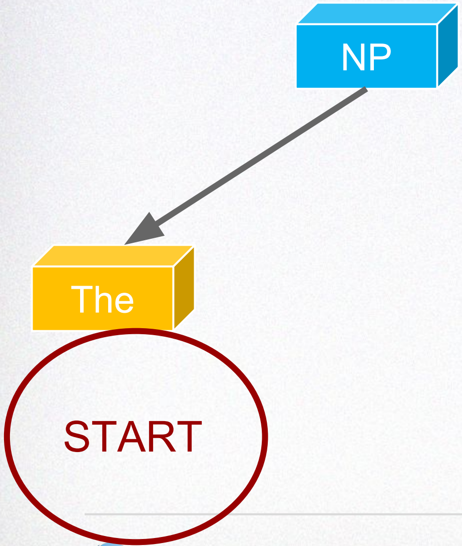
Quick Illustration of the Differences: Left-Corner

**LEFT-CORNE
R**

The

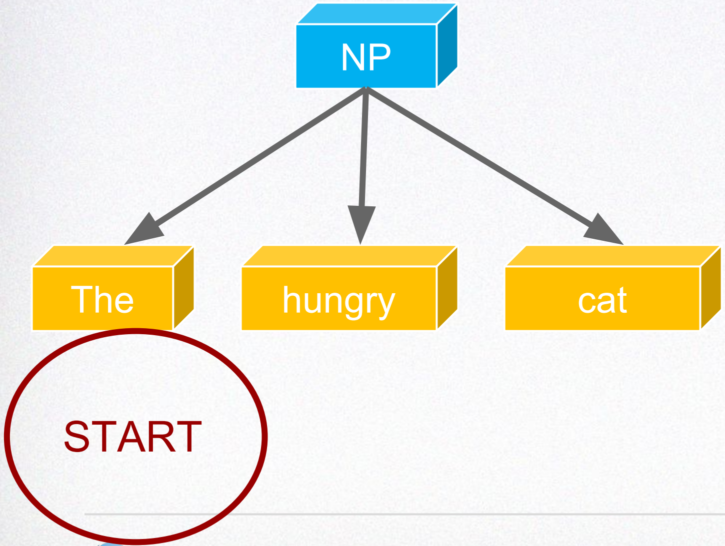
START

Quick Illustration of the Differences: Left-Corner



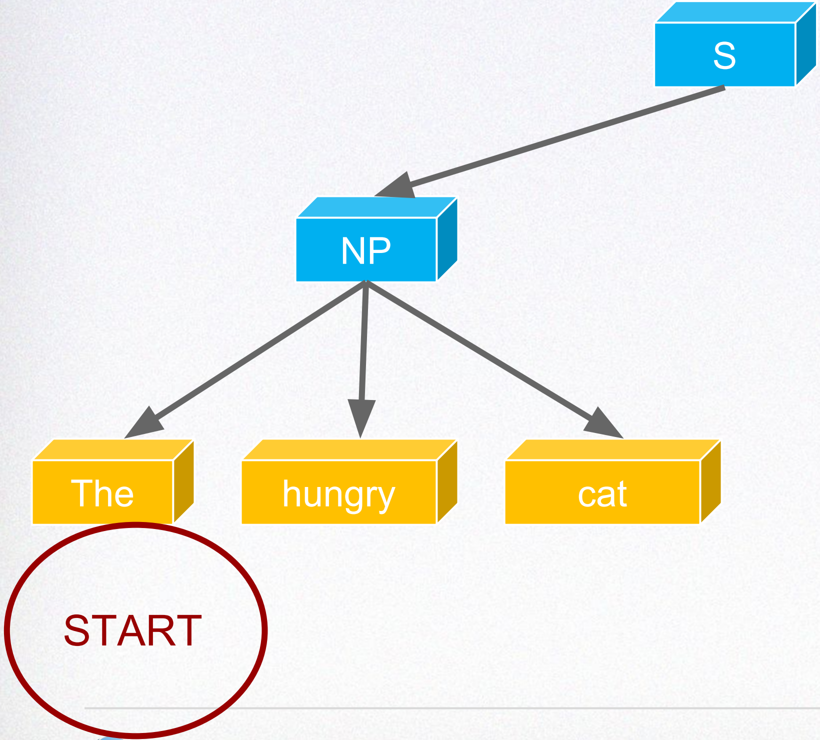
LEFT-CORNE R

Quick Illustration of the Differences: Left-Corner



LEFT-CORNE R

Quick Illustration of the Differences: Left-Corner



**LEFT-CORNE
R**

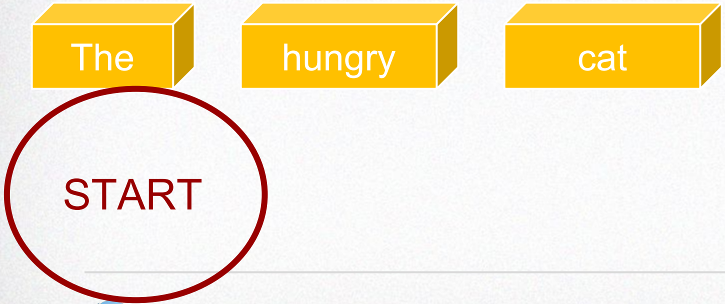
LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modelling Structure Makes Them Better - Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (ACL 2018)

Quick Illustration of the Differences: Bottom-Up

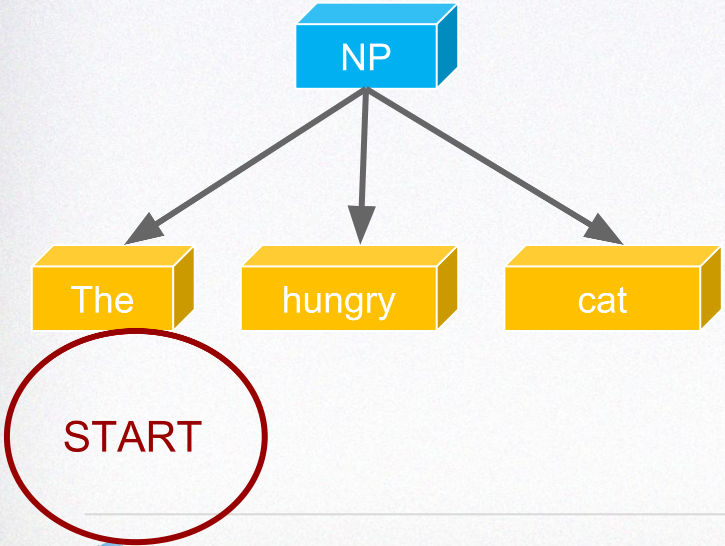
BOTTOM-UP

Quick Illustration of the Differences: Bottom-Up

BOTTOM-UP

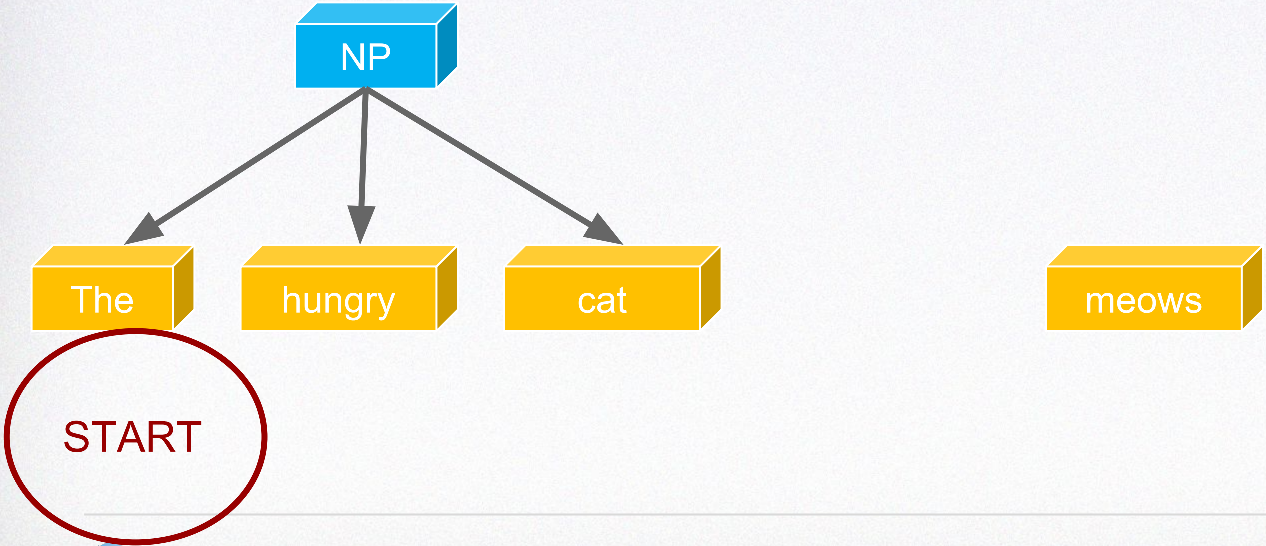


Quick Illustration of the Differences: Bottom-Up



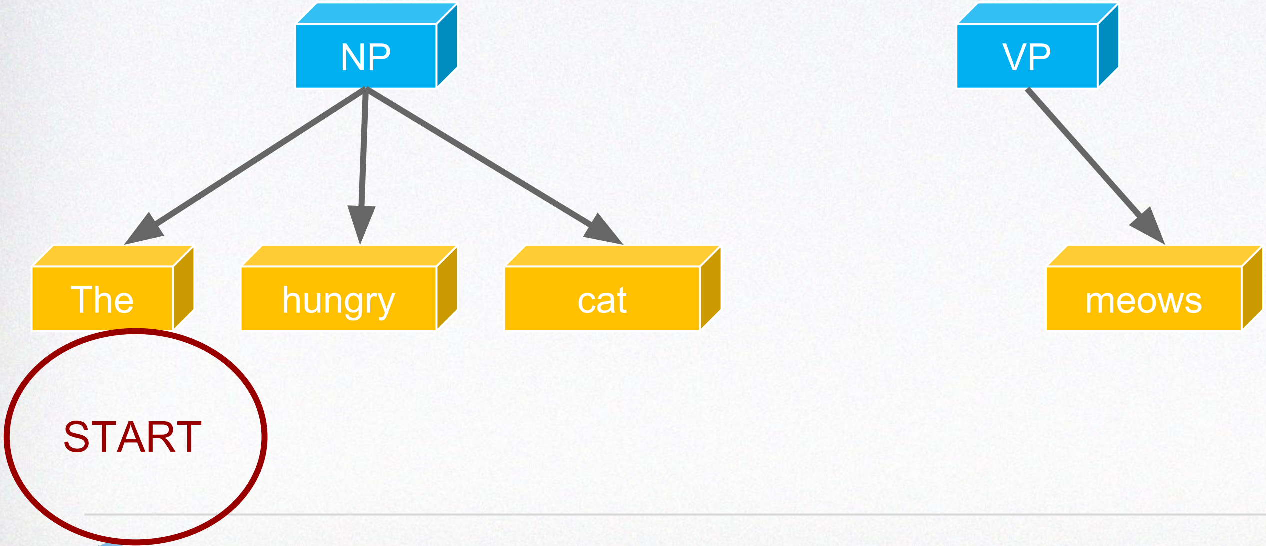
BOTTOM-UP

Quick Illustration of the Differences: Bottom-Up



BOTTOM-UP

Quick Illustration of the Differences: Bottom-Up



BOTTOM-UP

Why Does The Build Order Matter?

Machine learning

- The three different strategies yield different intermediate states during the generation process and impose **different biases** on the learner.

Cognitive

- Earlier work in **parsing** has characterised the strategies' plausibility in **human sentence processing** (Johnson-Laird, 1983; Pulman, 1986; Resnik, 1992). We evaluate these strategies as models of **generation** (Manning and Carpenter, 1997) in terms of number agreement accuracy.

Bottom-up Traversal

x, y: (S (NP the hungry cat) (VP meows))

The

Topmost stack element

The

Action: GEN(*The*)

Bottom-Up Traversal

x, y: (S (NP the hungry cat) (VP meows))

The
hungry
cat

Topmost stack element

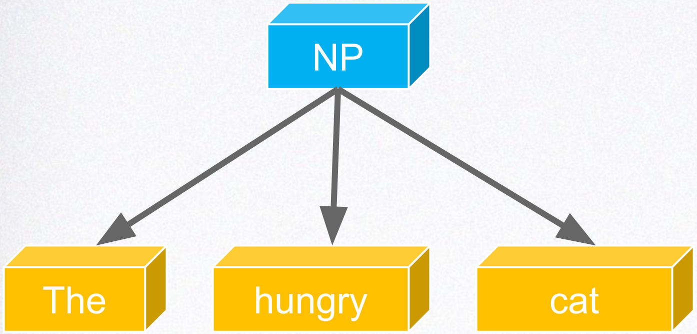
Action: GEN(*hungry*), GEN(*cat*)

Bottom-Up Traversal

x, y: (S (NP the hungry cat) (VP meows))

(**NP** *The hungry cat*)

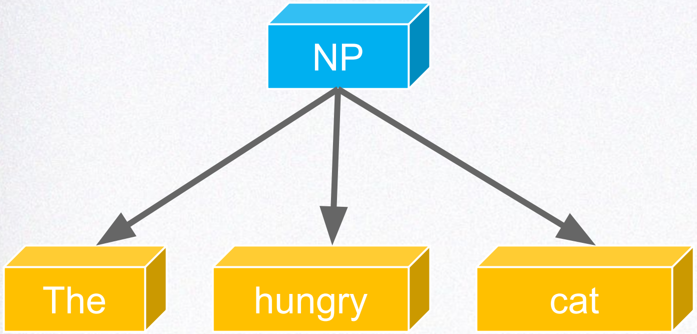
Topmost stack element



Action: REDUCE-3-NP

Bottom-Up Traversal

x, y: (S (NP the hungry cat) (VP meows))



(**NP** *the hungry cat*)
meows

Topmost stack element

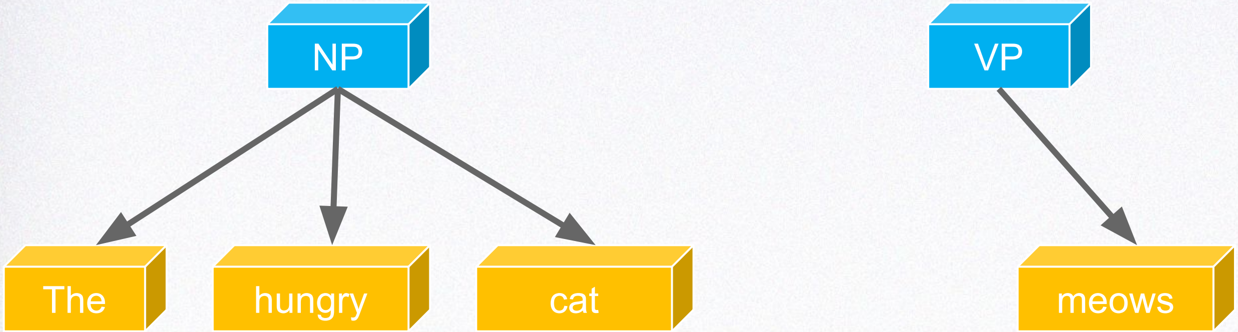
(**NP** *the hungry cat*)
(**VP** *meows*)

Topmost stack element

Action: REDUCE-1-VP

Bottom-Up Traversal: After REDUCE-1-VP

x, y: (S (NP the hungry cat) (VP meows))

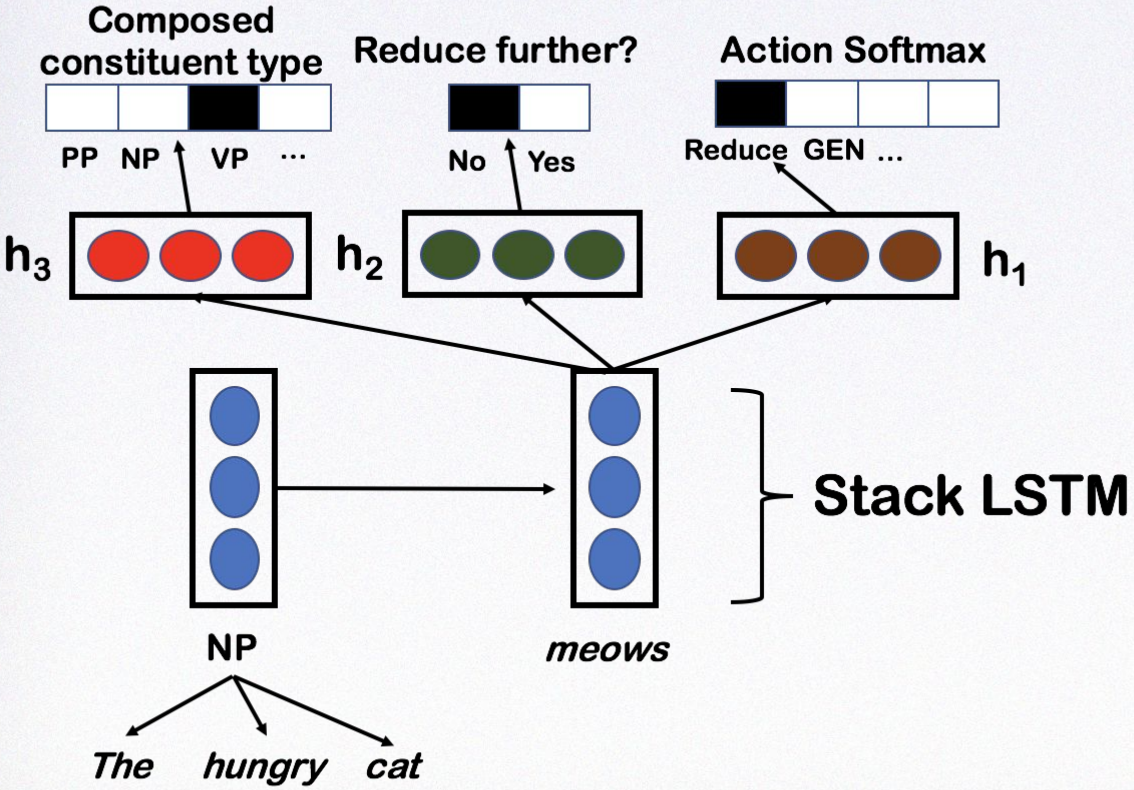


(**NP** *the hungry cat*)
(**VP** *meows*)

Topmost stack element

Action: REDUCE-1-VP

Bottom-Up Parameterisation of Constituent Extent



Stick-breaking construction

Summary Statistics

	Avg. Stack Depth	Dev ppl. $p(x, y)$
Top-Down	12.29	94.9
Left-Corner	11.45	95.9
Bottom-Up	7.41	96.5

Near-identical perplexity for each variant

Bottom-up has the shortest stack depth

Different Traversal Number Agreement Error Rates

	n=2	n=3	n=4
Our LSTM (H=350)	5.8	9.6	14.1
Top-Down	5.5	**7.8**	**8.9**
Left-Corner	**5.4**	8.2	9.9
Bottom-Up	5.7	8.5	9.7

Top-down performs best for n=3 and n=4

For n=4 this is **significant** (p < 0.05)

Lower is better

Part Three Recap and Outlook

- We proposed two new RNNG variants with different tree construction orders: **left-corner** and **bottom-up** RNNGs.

- Top-down construction still performs best in number agreement.
  - It is the most **anticipatory** (Marslen-Wilson, 1973; Tanenhaus et al., 1995).

- We can apply the three strategies to parsing and as linking hypothesis to human brain signal during comprehension (Hale et al., 2018).

Conclusion

- LSTM language models with **enough capacity** **can** learn number agreement well, while a strong character LSTM performs much **worse**.
- Explicitly modelling the syntactic structure with RNNGs that have a hierarchical inductive bias leads to **much better** number agreement.
  - Syntactic annotation alone does not help if the model is still sequential.
- **Top-down** construction order **outperforms** left-corner and bottom-up variants in difficult number agreement cases.
- Perplexity does **not** completely correlate with number agreement.

The end & thank you