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
Language exhibits **hierarchical** structure

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

...... but LSTMs work so well without explicit notions of structure.
Number agreement is a cognitively-motivated probe to distinguish **hierarchical** theories from purely sequential ones.

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

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>141,948</td>
<td>1,211,080</td>
</tr>
<tr>
<td>Types</td>
<td>10,025</td>
<td>10,025</td>
</tr>
<tr>
<td>Tokens</td>
<td>3,159,622</td>
<td>26,512,851</td>
</tr>
</tbody>
</table>

Number agreement dataset is derived from dependency-parsed Wikipedia corpus

**All** intervening nouns must be of the same number.

Number Agreement Dataset Overview

All intervening nouns must be of the same number

The vast majority of number agreement dependencies are sequential

<table>
<thead>
<tr>
<th># Attractors</th>
<th># Instances</th>
<th>% Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=0</td>
<td>1,146,330</td>
<td>94.7%</td>
</tr>
<tr>
<td>n=1</td>
<td>52,599</td>
<td>4.3%</td>
</tr>
<tr>
<td>n=2</td>
<td>9,380</td>
<td>0.77%</td>
</tr>
<tr>
<td>n=3</td>
<td>2,051</td>
<td>0.17%</td>
</tr>
<tr>
<td>n=4</td>
<td>561</td>
<td>0.05%</td>
</tr>
<tr>
<td>n=5</td>
<td>159</td>
<td>0.01%</td>
</tr>
</tbody>
</table>
First Part: Can LSTMs Learn Number Agreement Well?

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

Number of attractors

Number Agreement Error Rates

Random
Majority
Linzen LSTM (H=50)
Small LSTM Number Agreement Error Rates

- Random
- Majority
- Linzen LSTM (H=50)
- Our LSTM (H=50)

Lower is better


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

LSTM Number Agreement Error Rates

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

Domain and training settings matter too

Lower is better

Can Character LSTMs Learn Number Agreement Well?

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

Three Concrete Alternatives for Modeling Sentences

Sequential LSTMs without Syntax

Sequential LSTMs with Syntax (Choe and Charniak, 2016)

RNNG (Dyer et al., 2016)
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

Performance differences are significant ($p < 0.05$)

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

Lower is better
<table>
<thead>
<tr>
<th>Model Type</th>
<th>Dev ppl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM LM</td>
<td>72.6</td>
</tr>
<tr>
<td>Seq. Syntactic LSTM</td>
<td>79.2</td>
</tr>
<tr>
<td>RNNGs</td>
<td>77.9</td>
</tr>
</tbody>
</table>

**Perplexity**

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

- 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

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

- **Control condition**

**DISAGREE**

- Confounded by **first nouns**
  - Much less likely to affect human experiments

**AGREE**

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

Control Condition LSTM LM Error Rates

Lower is **better**
Control Condition Experiments for RNNG

Control Condition RNNG Error Rates

- 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.

Lower is better

Same y-axis scale as LSTM LM
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).
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.

\[ x: \text{the flowers in the vase are/is [blooming]} \]

**Top-down**

```
(S)
(NP (NP the flowers) (PP in (NP the vase)))
```

**Bottom-up**

```

(NP (NP the flowers) (PP in (NP the vase)))
```

**Left-corner**

```
(S)
(NP (NP the flowers) (PP in (NP the vase)))
```
Quick Illustration of the Differences: Top-Down
Quick Illustration of the Differences: Top-Down

START

NP

S

TOP-DOWN
Quick Illustration of the Differences: Top-Down
Quick Illustration of the Differences: Top-Down

Quick Illustration of the Differences: Left-Corner
Quick Illustration of the Differences: Left-Corner

The LEFT-CORNER
Quick Illustration of the Differences: Left-Corner

The hungry cat

START
Quick Illustration of the Differences: Left-Corner

Quick Illustration of the Differences: Bottom-Up

START

BOTTOM-UP
Quick Illustration of the Differences: Bottom-Up

The hungry cat

START

BOTTOM-UP
Quick Illustration of the Differences: Bottom-Up

Bottom-Up

START
Quick Illustration of the Differences: Bottom-Up

START

NP

The

hungry

meows

cat

BOTTOM-UP
Quick Illustration of the Differences: Bottom-Up

START

The

hungry

The

NP

cat

VP

meows
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

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

**Action:** GEN(\textit{The})
Bottom-Up Traversal

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

**Action:** GEN(*hungry*), GEN(*cat*)
Bottom-Up Traversal

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

\( (NP \text{ The hungry cat}) \)

Topmost stack element

**Action:** REDUCE-3-NP

**Bottom-Up Traversal**

\[ x, y: (S \text{ (NP the hungry cat) (VP meows)}) \]

**Action:** REDUCE-1-VP
Bottom-Up Traversal: After REDUCE-1-VP

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

![Diagram showing NP and VP with words The, hungry, cat, and meows, and the action REDUCE-1-VP]
Bottom-Up Parameterisation of Constituent Extent

Composed constituent type
- PP
- NP
- VP
- ...

Reduce further?
- No
- Yes

Action Softmax
- Reduce GEN ...

Stack LSTM

NP
- meows

The
- hungry
- cat

Stick-breaking construction

## Summary Statistics

<table>
<thead>
<tr>
<th></th>
<th>Avg. Stack Depth</th>
<th>Dev ppl. $p(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-Down</td>
<td>12.29</td>
<td>94.9</td>
</tr>
<tr>
<td>Left-Corner</td>
<td>11.45</td>
<td>95.9</td>
</tr>
<tr>
<td>Bottom-Up</td>
<td>7.41</td>
<td>96.5</td>
</tr>
</tbody>
</table>

Near-identical perplexity for each variant

Bottom-up has the shortest stack depth
### Different Traversal Number Agreement Error Rates

<table>
<thead>
<tr>
<th>Method</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our LSTM (H=350)</td>
<td>5.8</td>
<td>9.6</td>
<td>14.1</td>
</tr>
<tr>
<td>Top-Down</td>
<td>5.5</td>
<td>7.8</td>
<td>8.9</td>
</tr>
<tr>
<td>Left-Corner</td>
<td>5.4</td>
<td>8.2</td>
<td>9.9</td>
</tr>
<tr>
<td>Bottom-Up</td>
<td>5.7</td>
<td>8.5</td>
<td>9.7</td>
</tr>
</tbody>
</table>

**Lower is better**

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

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