An analytic study of how LSTM language models use prior linguistic context. We measure changes in LSTM performance, as a result of ablations applied to contextual features of the input, during evaluation.

### Setup
- Perturbations applied only during evaluation.
- Datasets: Penn Treebank (PTB) and Wikitext-2 (Wiki).
- Standard LSTM LM architecture (Merity et al., 2018).
- All results are reported on the development set (to protect the test set).
- Measuring changes in negative log likelihood:

\[
\text{NLL} = -\frac{1}{T} \sum_{i=1}^{T} \log P(w_t | w_{t-1}, \ldots, w_1)
\]

### How much context is used?
- Perturbation: guess a context size, delete all prior tokens

LSTM language models can use at least about 200 tokens of context, on average.

### Does word order matter?
- Perturbation: shuffle/reverse spans in prior context

Local word order only matters within the most recent sentence, ~20 tokens.

Global word order only matters for the most recent 50 tokens.

### Can LSTMs copy words?

#### Three Categories of Target Words
1. Appear in their own nearby context (within 50 tokens).
2. Appear only in their own long-range context (beyond 50 tokens).

LSTMs can regenerate words seen in nearby context.

Neural Caches (Grave et al., 2017b) help words that can be copied from long-range context, the most.

### Implications
- Improve existing models!
- Compare model classes on more than just test set perplexities!
- Can we decouple the data from the models?
  Experiment with different model classes and different languages
- Theoretical justifications???

### Code:
https://github.com/urvashik/lm-context-analysis

### References

### Acknowledgements
We gratefully acknowledge support of the DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF15-1-0462 and the NSF via grant IIS-1514268.