LET'S TEST COMPOSITIONALITY

Michael Neely & Leila Talha

Natural Language Processing 2, University of Amsterdam

UNIVERSITY OF AMSTERDAM

Problem Statement

The principle of compositionality states that "the meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined" (Kamp and Partee 1995).

Are data-driven deep learning models compositional? We explore the behavior of recurrent and attention-based neural models in terms of two aspects of the compostionality tests proposed by (Hupkes et al. 2019).

- **Localism**: Whether models' composition operations are local or global
- **Systematicity**: Whether models systematically recombine known parts and rules

We evaluate model performance in the context of a sequence-to-sequence translation task with a new artificial compositional language.

PCFG List-Edit-Task

The vocabulary of PCFG-LET consists of the **integers 1 to 520 (exclusive)**, the **empty token** `E` (a special symbol that represents an empty list), **unary and binary interpretation functions**, and **commas** to separate binary function arguments.
Examples of commands and their corresponding output are:

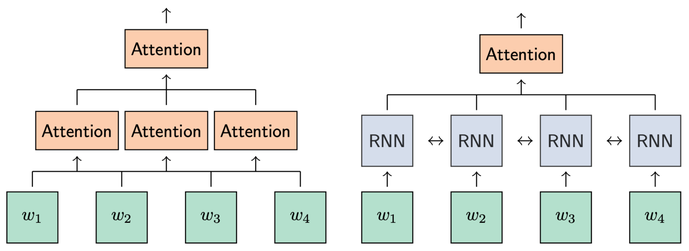
- `mirror 1 2 3 4 5` → `2 1 3 5 4`
- `intersection remove_repeated 1 1 2 3 4 4, 4 5 6` → `E`

As the examples indicate, PCFG-LET commands can be **nested**.

Models

Models are implemented and tested using the OpenNMT-py (Klein et al. 2017) library. We train models for 25 epochs or until convergence, which we define as five successive epochs with no improvement in word accuracy and perplexity on the validation set. An epoch consists of 1328 steps of 64 sequence batches.

1. **LSTMS2s**: Fully recurrent, bidirectional model with scaled dot-product attention and 512-dimensional hidden state and word vector size.
2. **Transformer** (Vaswani et al. 2017): Six stacked layers per encoder and decoder with 8 self-attention heads, a feed-forward network with a hidden size of 2048 and embedding and sub-layers of dimensionality 256.

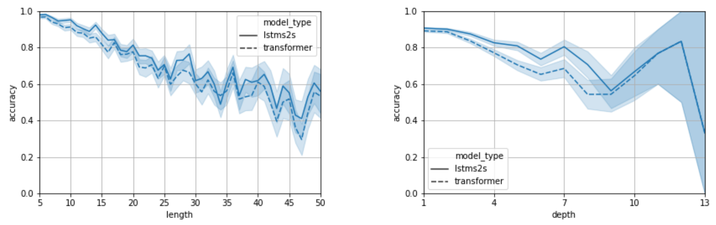


LSTMS2S (left) and **Transformer** (right) model architectures taken from (Hupkes et al. 2019)

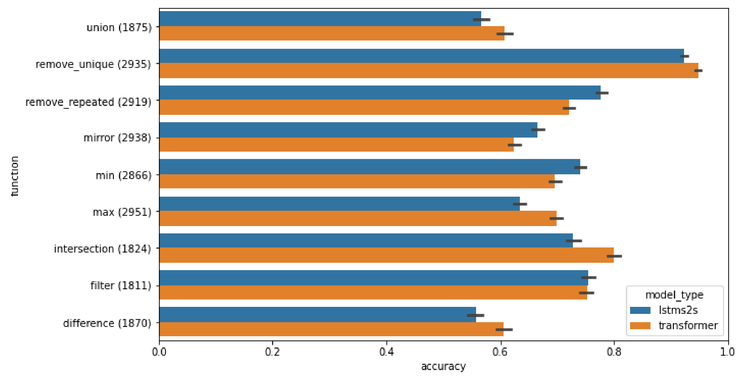
Task Performance

Experiment	LSTMS2S	Transformer
Task Accuracy*	0.8839 ± 0.0051	0.8569 ± 0.0010
Localism*	0.6704 ± 0.0079	0.6371 ± 0.0097
Localism†	0.7232 ± 0.0066	0.6925 ± 0.0096
Systematicity*, empty token	0.4456 ± 0.0028	0.4483 ± 0.0035
Systematicity*, numbers	0.5232 ± 0.0118	0.5340 ± 0.0136
Systematicity*, functions	0.5037 ± 0.0076	0.4980 ± 0.0190

Results are average over three runs with standard deviation included. Sequence accuracy is denoted with *, and consistency score with †.



Sequence accuracy per function (number of function occurrences shown in parentheses):



Systematicity

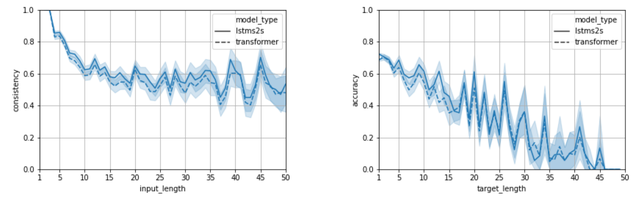
Similarly to (Hupkes et al. 2019, the **Systematicity** of both models was tested by focusing on words w_1 and w_2 that never occur together in the training corpus. The systematic treatment to input commands has been decomposed into three parts: the **empty token**, **numbers**, and **functions**.

The three subtests were conducted by retraining both models on a new dataset manipulated to prevent `mirror` from co-occurring with (1) the empty token (2) numbers 7 and 13 and (3) `union`.

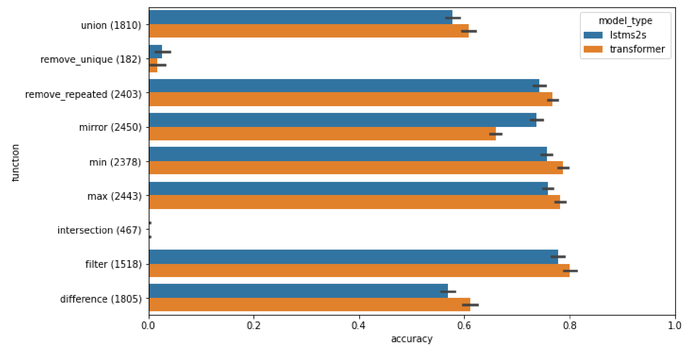
Subsequently, the models were tested on three different sets such that every occurrence of the `mirror` function either: (1) contains an empty token in its argument list, (2) has an argument list containing the numbers 7 or 13, or (3) is applied to a nested `union` function. Test performance was measured in terms of sequence accuracy.

Localism

We unroll computations and compare each model's successive local predictions to its global ones. The final prediction is **consistent** if it matches the global prediction, and **accurate** if it matches the target.



Sequence accuracy per function where the correct target is not an empty token (number of function occurrences shown in parentheses):



Discussion

- Sequence accuracy is inversely proportional to sequence length and depth
- Both models fail to recombine known parts and rules, especially when a special symbol is involved
- High localism consistency and accuracy scores suggest models apply a recursive strategy
- Models adopt a strong bias for predicting empty tokens as the output of `remove_unique` and `intersection` functions

Future Work: Could the increased modelling capacity of Memory-Augmented RNNs (Suzgun et al. 2019) lead to better hierarchical composition?

References

Hupkes, Dieuwke et al. (2019). *Compositionality decomposed: how do neural networks generalise?* arXiv: 1908.08351 [cs.CL].

Kamp, Hans and Barbara Partee (1995). "Prototype theory and compositionality". In: *Cognition* 57.2, pp. 129–191. ISSN: 0010-0277. DOI: https://doi.org/10.1016/0010-0277(94)00659-9. URL: http://www.sciencedirect.com/science/article/pii/0010027794006599.

Klein, Guillaume et al. (July 2017). "OpenNMT: Open-Source Toolkit for Neural Machine Translation". In: *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, pp. 67–72. URL: https://www.aclweb.org/anthology/P17-4012.

Suzgun, Mirac et al. (2019). *Memory-Augmented Recurrent Neural Networks Can Learn Generalized Dyck Languages*. arXiv: 1911.03329 [cs.CL].

Vaswani. Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].