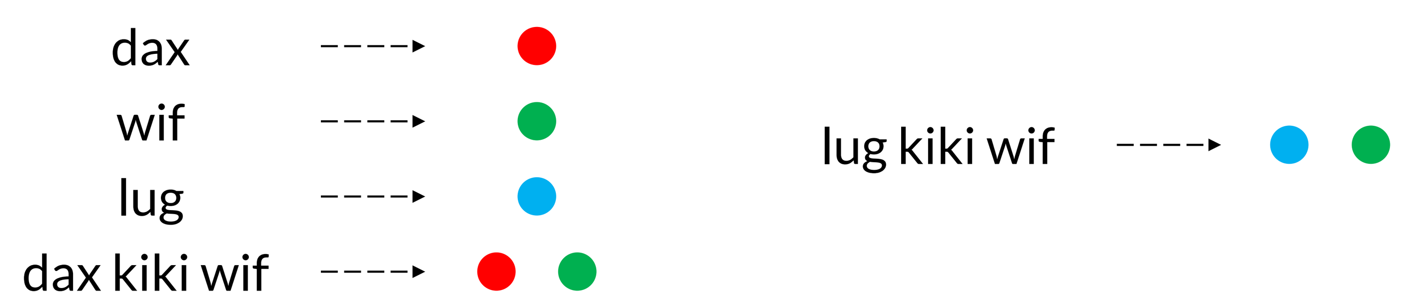
Compositional Generalization by Learning Analytical Expressions

Qian Liu[1], Shengnan An[2], Jian-Guang Lou[3], Bei Chen[3], Zeqi Lin[3], Yan Gao[3], Bin Zhou[1], Nanning Zheng[2], Dongmei Zhang[3]

[1]Beihang University; [2]Xi'an Jiaotong University; [3]Microsoft Research

Introduction

When using language, humans have a remarkable ability to recombine known parts to understand novel sentences they have never encountered before. For example, once humans have learned the meanings of "*dax*", "*wif*", "*lug*" and "*dax kiki wif*", it is effortless for them to understand the meaning of "*lug kiki wif*".



Understanding language compositionality is a basic and essential capacity for human beings, but modern translation models dramatically fail to obtain a satisfactory performance on compositional generalization. Back to the above example, current models fail to generalize to understand "*lug kiki wif*".

Motivated by work in cognition which argues compositionality can be captured by variable slots with symbolic functions, we propose a memory-augmented neural model to achieve compositional generalization by ***L**earning* **An**alytical **E**xpressions (**LANE**). It seizes a great ability of compositional generalization, reaching 100% accuracies on a well-known synthetic benchmark. As far as we know, our model is the first neural model to pass all compositional challenges addressed by previous works on the benchmark without extra resources.

Compositional Generalization Assessment

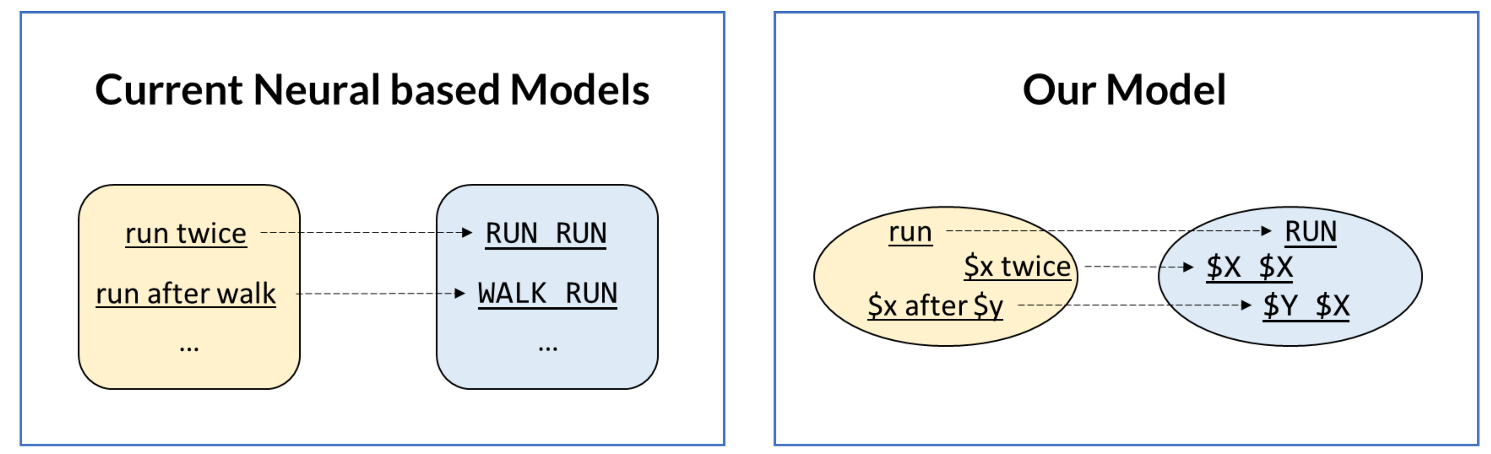
Since the study on compositional generalization of deep neural models is still in its infancy, most previous works employ artificial datasets to conduct assessment. As one of the most important benchmarks, SCAN [1] is proposed to evaluate the compositional generalization ability of translation models. SCAN describes a simple navigation task that aims to translate compositional navigation sentences into executed action sequences such as from "*jump thrice*" to `JUMP JUMP JUMP`. To evaluate different nature of compositional generalization, SCAN is split into different ways, making up three compositional challenges: Add jump, Around Right, Length Generalization.

	Add Jump	Around Right	Length Generalization
Train	jump walk twice walk around left	turn around left turn opposite right walk around left	look around left look around left twice look around left twice after look
Test	jump around left	turn around right	look around left twice after look around left
	No complex command of jump in training	*"around right" is held out from the training set*	*Train: length of the action sequence is shorter than 24 actions; Test: all action sequences longer than or equal to 24 actions.*

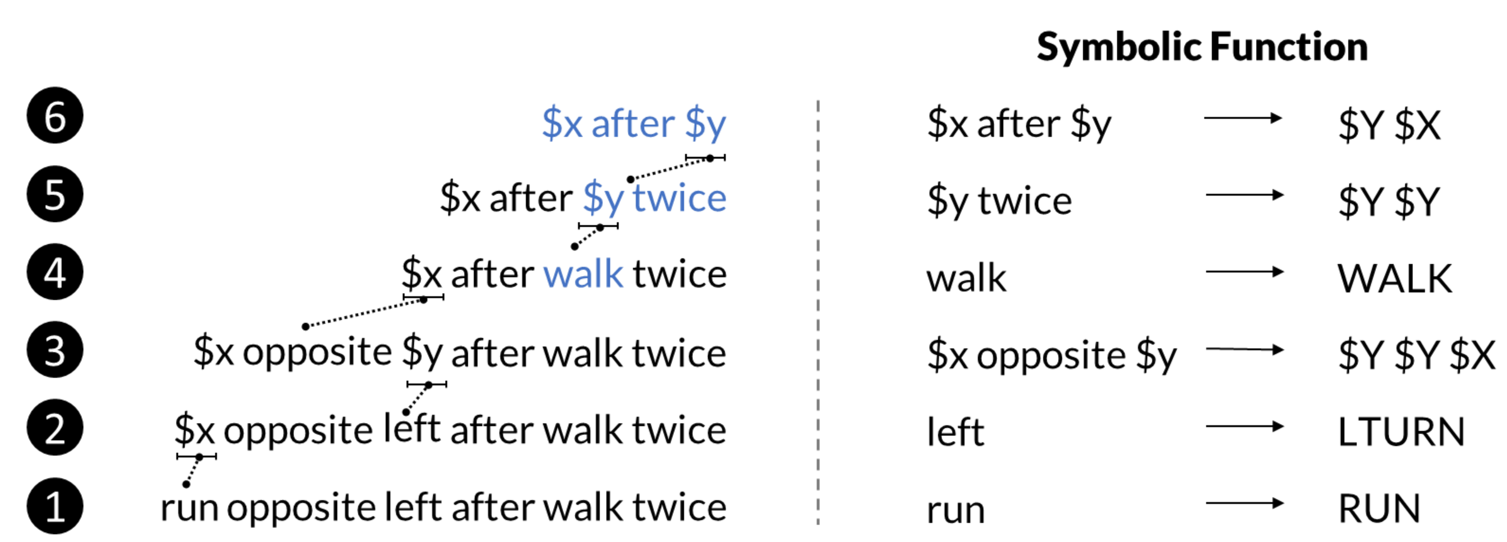
Modeling Compositionality

The compositionality of language constitutes an algebraic system, of the sort that can be captured by symbolic functions with variable slots [2].

- Current neural models fit a mapping from source instances into destination instances.
- Our model learns a mapping from source analytical expressions into destination analytical expressions, just as symbolic functions.

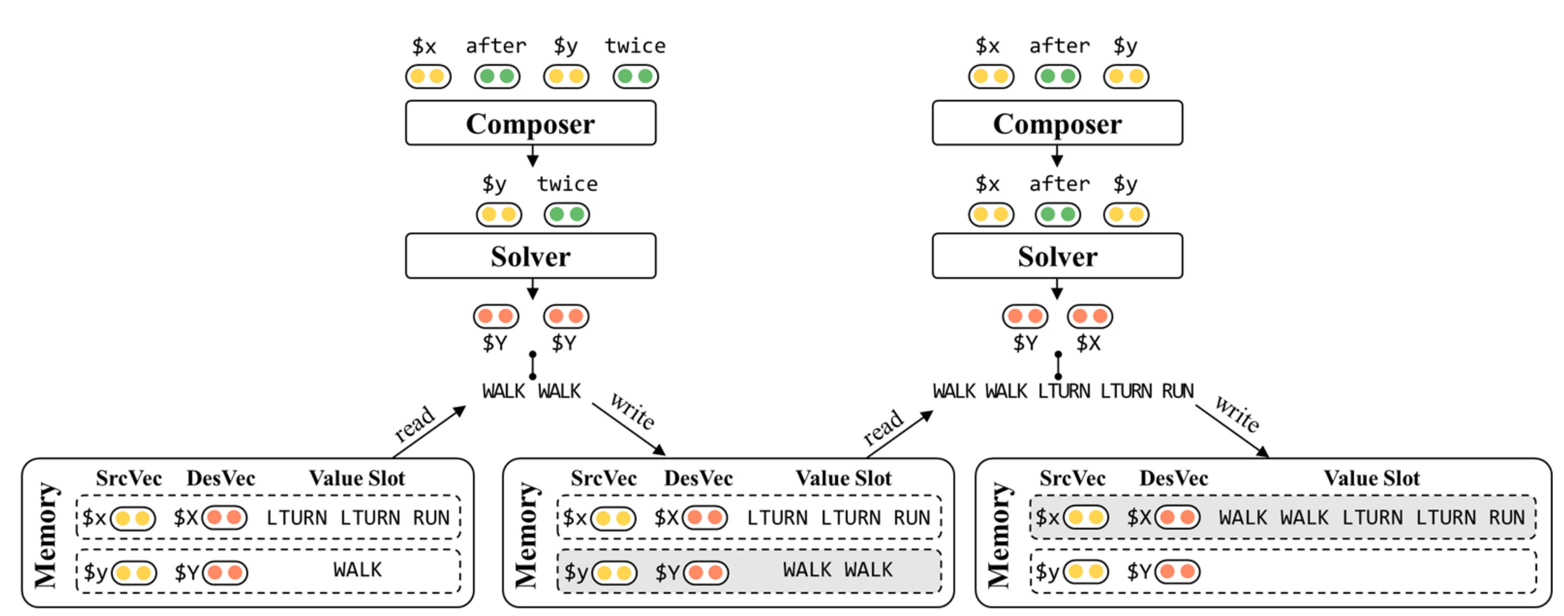


Given an input utterance "*run opposite left after walk twice*", our method will first recognize "*run*" and supersede it with a variable "$x". Step by step, the utterance is decomposed into a sequence of source analytical expressions, and then our method solves them by symbolic functions.



Memory-Augmented Model

Similarly, we propose a memory-augmented neural model to achieve compositional generalization by automatically learning the above analytical expressions. It understands an utterance via interaction between Composer, Solver and Memory.



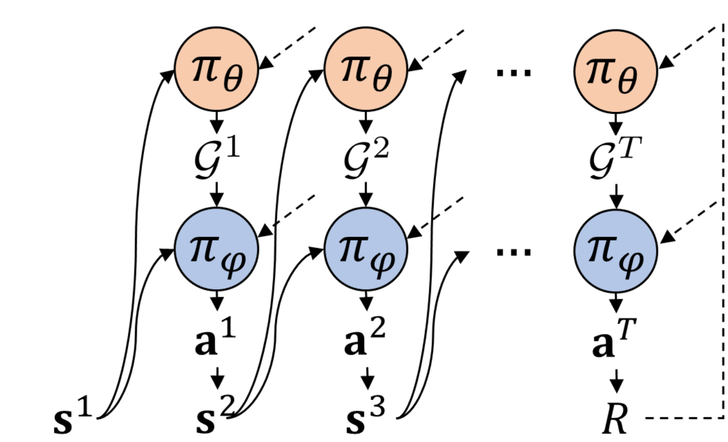
Our model takes several steps to understand a sentence. At the beginning of a step, "*$x after $y twice*" is fed into Composer. Then Composer finds "*$y twice*" and sends it to Solver. Receiving "$y twice", Solver first translates it into "*$Y $Y*". Using "*$Y $Y*" as the skeleton, Solver obtains `WALK WALK` by replacing "*$Y*" with its corresponding value slot `WALK` in Memory. Meanwhile, since "*WALK*" has been used, the value slot which stores WALK is set to empty. Next, Solver writes WALK WALK into an empty value slot. Finally, "*$y twice*" in "*$x after $y twice*" is superseded by "$y", producing "*$x after $y*" for the next step.

Hierarchical Reinforcement Learning

Training our proposed model is non-trivial for two reasons since there is no supervision about intermediate results between Composer and Solver. We employ Hierarchical Reinforcement Learning: Composer is regarded as a high-level agent, while Solver is a low-level agent.

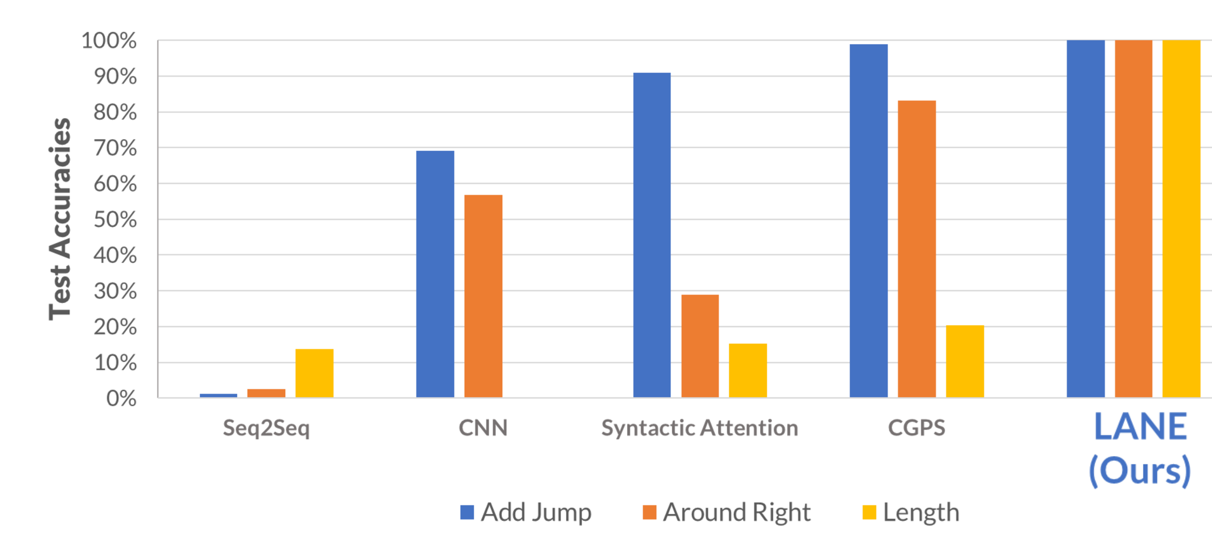
States $\mathbf{s}^t$ contains:
- Memory
- Input for Composer

Action represents:
- output of Composer
- output of Solver



Experimental Results

On the SCAN benchmark, **LANE** achieves stunning100% test accuracies on all tasks. Compared with state-of-the-art baselines without extra resources, **LANE** achieves a significantly higher performance. As far as we know, **LANE** is the first neural model to pass all tasks without extra resources



References

[1] B. M. Lake and M. Baroni. (2018). "Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent network.", In: Proceedings of the 35th International Conference on Machine Learning (ICML).

[2] M. Baroni. (2020) "Linguistic generalization and compositionality in modern artificial neural networks". In: Philosophical Transactions of the Royal Society.

Contact

qian.liu@buaa.edu.cn

Code