Do Language Models See a Space?

Jan Hůla[1,2], Jiří Janeček[2], David Mojžíšek[2], and Mikoláš Janota[1]

[1] Czech Technical University in Prague, Prague, Czech Republic
[2] University of Ostrava, Ostrava, Czech Republic

1 Introduction

It is widely accepted that Neural Networks (NNs) can produce accurate answers for tasks that are inherently symbolic and require some sort of reasoning, i.e. a sequencing of previously unseen combination of facts [1, 2]. This is often demonstrated by testing the network on inputs that are novel combinations of inputs that the network saw during training.

In this work, our aim is to take a closer look at this intriguing ability of NNs and to provide a mechanistic explanation of the process behind it. We choose a simple domain of geometric reasoning where the task is to guess the positions of selected points from a hidden figure which the model sees only through a set of relations/constraints that uniquely determine the figure in a discrete 2D grid. To correctly guess the positions of the queried points, the model needs to learn the semantics behind the language and also needs to be able to "reason" with the learned "model". We generate a synthetic dataset that allows us, in a controlled fashion, to test different modes of generalization of the trained model and debug the trained model.

2 Description of the Task

On a high level, the input to the model is a sequence of tokens describing geometric constraints together with the position of several points that uniquely determine positions of all the remaining points mentioned in the constraints. This means that the input describes a hidden figure and the task the model is trained for is to predict the positions of selected points mentioned in the constraints.

Example For simplicity, let us assume a simplified language for describing figures in the 2D grid using two different constraints: $\mathit{square}(x_1, x_2, x_3, x_4)$, $\mathit{equi}(x_1, x_2, x_3, x_4)$, where the meaning of the first constraint is as expected and the second constraint says that the segment between points x_1 and x_2 has same angle and length as the segment between points x_3 and x_4, i.e., the segment (x_3, x_4) is a translation of segment (x_1, x_2). Using these two types of constraints, we can generate a dataset of training sequences by following a simple procedure to create one training sequence:

1. Sample several random constraints where for each constraint, we either create new variables or reuse variables that were already used in constraints generated before. The number of constraints is chosen randomly from an interval $[3, 9]$.

2. Use an SMT solver to instantiate values to a subset of variables so that the values of the rest of the variables are uniquely determined and the whole figure fits into a grid of size 30×30 points.

3. Select one of the unassigned variables as a variable for which the language model should predict the correct value.

4. Concatenate the constraints, instantiated variables, and the query variable into a sequence of tokens which will be used as an input to the language model and use the uniquely determined value of the query variable as a label to be predicted by the language model.

For the simplest case of only one constraint, we could generate the following training example:

Input: $square(a, b, c, d); a = (0, 0), b = (0, 1), c = (1, 0)?d$, **Output:** $(1, 1)$ The string is tokenized so that individual characters form one token except for names of constraints and point coordinates which are both assigned to single token (i.e., *square* would be one token and $(0, 0)$ would also be one token.)

3 Results

We show that a Transformer language model is able to learn to predict the correct assignment of the query variable with $\sim 50\%$ accuracy and that we can recover the hidden figure from the last-layer embeddings of the model. We also show that a Graph Neural Network which operates on a bi-partite graph of constraints and variables can learn to predict the correct assignment by $\sim 90\%$ accuracy which points to a gap that could hopefully be filled with better architectures.

References

[1] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. hsin Chi, F. Xia, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.