Compositionality in a simple corpus

Manuel Vargas Guzmán, Maria Boritchev, Jakub Szymanik, Maciej Malicki

Compositionality in a simple corpus

Manuel Vargas Guzmán[1,2], Maria Boritchev[1], Jakub Szymanik[3], Maciej Malicki[1]
(1) IMPAN, Jana i Jędrzeja Śniadeckich 8, 00-656 Warsaw, Poland
(2) University of Warsaw, Krakowskie Przedmieście 26/28, 00-927 Warsaw, Poland
(3) CIMeC and DISI - University of Trento Rovereto (TN) - Italy
m.vargas-guzman@uw.edu.pl, mboritchev@impan.pl, mmalicki@impan.pl,
jakub.szymanik@gmail.com

RÉSUMÉ

Nous étudions la capacité des réseaux de neurones à apprendre des structures compositionnelles en nous concentrant sur un corpus logique simple bien défini, et sur les phénomènes de compositionnalité centrés sur les preuves. Nous menons notre étude dans un cadre minimal en créant un corpus logique simple, où tous les phénomènes liés à la compositionnalité proviennent de la structure des preuves, car toutes les phrases du corpus sont des implications en logique propositionnelle. En entraînant des réseaux de neurones sur ce corpus, nous testons différents aspects de la compositionnalité, à travers des variations de la longueur des preuves et des permutations des constantes en jeu.

ABSTRACT

We investigate the capacity of neural networks (NNs) to learn compositional structures by focusing on a well-defined simple logical corpus, and on proof-centered compositionality. We conduct our investigation in a minimal setting by creating a simple logical corpus, where all compositionality-related phenomena come from the structure of proofs as all the sentences of the corpus are propositional logic implications. By training NNs on this corpus we test different aspects of compositionality, through variations of proof lengths and permutations of the constants.

MOTS-CLÉS : Compositionalité, logique, raisonnement, réseaux de neurones.

KEYWORDS: Compositionality, logic, reasoning, neural networks.

1 Introduction

Compositionality is a vastly discussed subject across natural language semantics, logic, but also natural language processing and nowadays, neural networks. Partee (1984) defines compositionality as the principle according to which"the meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined". In the past years, the investigation of the capacity of neural networks (NNs) to compositionally use/produce/deduce rules and/or sentences has gained more and more importance in the natural language processing field, in particular through the scope of natural language understanding tasks. Oscillating between mathematics and sentences in English, works such as Bowman *et al.* (2015), Saxton *et al.* (2019), and most recently Ontanon *et al.* (2022) show different ways in which NNs can be seen as more or less compositional, depending on

Research supported by the National Center of Science [grant 2020/37/B/HS1/04220]

the task and the mean of testing.

Bowman *et al.* (2015) test the capacity of LSTMs, using explicit clues, to discover and implicitely use recursive compositional structures. To do so, the authors implement the task as a classification one, outputting one of 7 possible logical relations between a given pair of sentences. Saxton *et al.* (2019) take a very different approach to the compositionality question by investigating the capacity of neural models to solve mathematical problems, in English, of various types (arithmetic, algebra, probability, calculus). The goal of the authors is to study the capacity of Transformers and Recurrent Neural Networks (RNNs) to compose and generalize mathematical concepts and operations. The results and performances vary greatly from one type of mathematical problem to another. Ontanon *et al.* (2022) introduce a dataset designed to evaluate the capacity of NNs to perform logical inferences. The dataset is composed of sentences that use propositional logic, a fragment of first order logic, and English. Hupkes *et al.* (2020) present a systematic set of tests for NNs capacity to compositionally generalize a rule set: (1) capacity of the NN to recombine known parts and rules to produce results it has never been exposed to before; (2) capacity of the NN to extend its predictions to data longer than the one it has been exposed to in training; (3) preference of the NN to compose in a local or in a global way; (4) robustness of the NN's predictions w.r.t synonym substitution; (5) preference of the NN towards rules or exceptions during training. The authors then instantiate the test suite on an artificial dataset and apply it to a RNN, a convolution-based neural network, and a transformer. The work presented in our article is placed in the footsteps of this latter approach.

The motivation for our work is two-fold: we want to investigate the capacity of neural networks to produce natural reasoning, but the approach we are taking grows from mathematical reasoning first. Indeed, implication is a fundamental element of natural language inference and understanding, as well as logic. Because of this, we begin by considering a simple corpus of propositional logic implications. Following approaches developed in previous work, in particular Hupkes *et al.* (2020), we focus on compositionality. To our knowledge, we are the first to study inference in the presence of multiple premises, and to work specifically on proof-structure compositionality and its different aspects. We do a fine-grained analysis of the output errors of our models by computing the Hamming distances between expected outputs and actual outputs of our models. As Hamming distances measure the number of differences between the two compared vectors, they constitute a tool that allows us to quantify how far away from the right answer the wrong answers are. In the following, we present our data in section 2, then we develop our experimental set-up in section 3. In this section we conduct an error analysis, and we introduce two compostionality tests. The data, code and materials for this article are shared in the GitHub respository `https://github.com/MBoritchev/compostitionality-simple-corpus`.

2 A simple corpus

The logical language of our corpus is defined as follows: let $\mathcal{C} = \{X_1 \dots X_n\}$ be a set of *constants*, we build *formulas* as logical implications: "$X_i \to X_j$". Then, we define a *knowledge base* $\mathcal{KB}$ which is a set of formulas, called *premises*. From a $\mathcal{KB}$, we can prove new formulas by using a unique derivation rule:

$$\frac{X_i \to X_j \qquad X_j \to X_k}{X_i \to X_k}$$

Given a $\mathcal{KB}$, a formula h is a *valid hypothesis* or a *conclusion*, if $h \in \mathcal{KB}$ or if it can be derived from

a set of premises $\pi \subseteq \mathcal{KB}$ by using the derivation rule. We write $\pi \vdash h$ to denote that there exists such a proof; if h cannot be proved from $\mathcal{KB}$, we write $\mathcal{KB} \nvdash h$, and call it an *invalid hypothesis*.

We represent $\mathcal{KB}$ as a directed graph $G = (V, E)$ where V is a set of vertices and $E \subseteq \{(u, v) \,|\, (u, v) \in V^2 \text{ and } u \neq v\}$ is a set edges. In this graph, each vertice is a constant and each edge corresponds to a premise. A proof corresponds then to a (directed) path between two vertices.

In our study, $\mathcal{KB}$ has the form of a tree. Since in this case, there is at most one path between any two vertices, for any h such that $\mathcal{KB} \vdash h$, there is a unique shortest proof, comprised of premises from a set π, that witnesses it. In particular, the *length* of the proof of h is the size of π.

Example : Let $\mathcal{KB}$ be the graph from figure 1, then $\mathcal{KB} \vdash X_3 \rightarrow X_{24}$ as there is a path from vertice X_3 to vertice X_{24} in the graph, in other words, $X_3 \rightarrow X_{24}$ can be derived from $\pi = \{X_3 \rightarrow X_9, X_9 \rightarrow X_{13}, X_{13} \rightarrow X_{24}\}$; therefore, $X_3 \rightarrow X_{24}$ is a valid hypothesis/conclusion. On the other hand, we have that $\mathcal{KB} \nvdash X_{30} \rightarrow X_{27}$ since there is no path that connects those two vertices and hence, the formula $X_{30} \rightarrow X_{27}$ is an invalid hypothesis.

We investigate neural models such that, for a given knowledge base $\mathcal{KB}$ and a hypothesis h, the model provides the necessary set π of premises from $\mathcal{KB}$ to prove h, if they exist. Otherwise, it indicates that there is no such π. We consider two architectures: a multilayer perceptron (MLP) and a recurrent neural network (RNN).

Data encoding To train a model, we use a multi-label approach that consists of a neural network $f(X) = \hat{y}$, where the input $X = [\mathcal{KB}, h]$ is a vector that encodes the set $\mathcal{KB} = \{p_1, \ldots, p_n\}$ of all premises and a hypothesis h, and the output $\hat{y}$ is the predicted value of the label y, a binary vector of size n such that for each element $i \in y$

$$i = \begin{cases} 1 & \text{if } p_i \in \pi \\ 0 & \text{otherwise} \end{cases}$$

where $\pi \subseteq \mathcal{KB}$ is the set of necessary premises to prove h. Moreover, if $\mathcal{KB} \nvdash h$, then y consists of n zeros. Therefore in the sequel we also refer to invalid hypotheses as hypotheses with proof length 0, premises in $\mathcal{KB}$ are hypotheses with proof length 1, etc.

Example : Let $\mathcal{KB} = \{X_1 \rightarrow X_3,\ X_3 \rightarrow X_6,\ X_6 \rightarrow X_4,\ X_6 \rightarrow X_2,\ X_6 \rightarrow X_5\}$ and $h = X_1 \rightarrow X_5$. Then, the vectors X and y are built as follows:

$$\begin{aligned} X &= [X_1 \rightarrow X_3,\ X_3 \rightarrow X_6,\ X_6 \rightarrow X_4,\ X_6 \rightarrow X_2,\ X_6 \rightarrow X_5,\ X_1 \rightarrow X_5] \\ y &= [1\ 1\ 0\ 0\ 1] \end{aligned}$$

The above vector encodes a proof of length 3.

Each formula from input X is encoded in a vector of dimension $2n$ (where n is the the size of the set $\mathcal{C}$) as a *one-hot* fashion. For instance, let $\mathcal{C} = \{X_1,\ X_2,\ X_3,\ X_4,\ X_5\}$, then the formula $X_2 \rightarrow X_5$ is encoded as

$$[0\ 1\ 0\ 0\ 0 \,|\, 0\ 0\ 0\ 0\ 1]$$

where the first n digits represent X_2 and the constant X_5 is encoded within the last n bits from the vector.

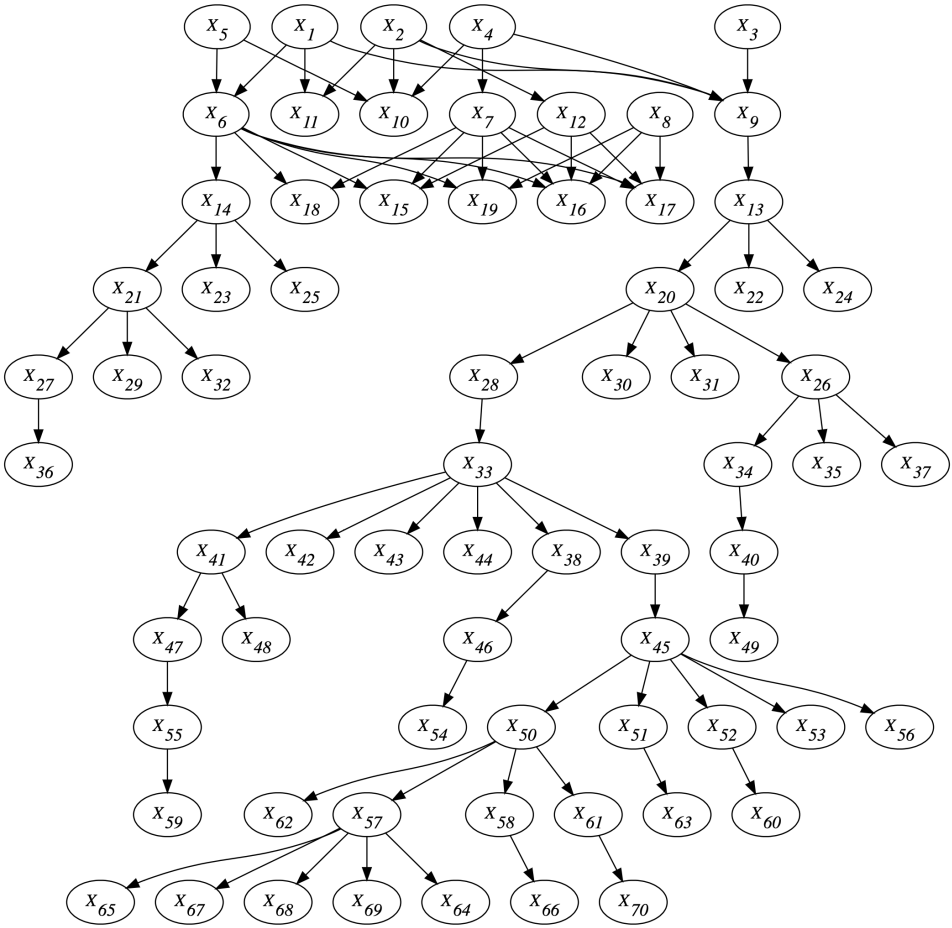


FIGURE 1 – A Knowledge Base built with 70 constants and 82 formulas

3 Tests and limits

In the preliminary study we present here, we only consider one $\mathcal{KB}$, depicted in figure 1: it is randomly built, with 70 constants and 82 premises. Among the $4{,}830 = 70 \times 69$ hypothesis, 531 are valid in the $\mathcal{KB}$.

3.1 Initial experiment

First, we trained and tested our models with 75/25% split of the $\mathcal{KB}$, stratified by length of proofs. Table 1 shows the detailed data distribution for each class.

Length of proofs	Total data	Train data	Test data
0	4299	3224	1075
1	82	62	20
2	75	56	19
3	63	47	16
4	60	45	15
5	51	38	13
6	54	40	14
7	41	31	10
8	42	32	10
9	35	26	9
10	28	21	7

TABLE 1 – Data distribution for our initial experiment

Neural networks setup We trained both the MLP and the RNN using the *Adamax* algorithm to optimize the weight values with its default learning rate of 0.001. The MLP has a single hidden layer of 2500 neurons, and the RNN is equiped with two hidden layers, each with 200 neurons. For both architectures, the *hyperbolic tangent* function (*tanh*) is used in the hidden layers, and the *sigmoid* function in the output layer. Moreover, every layer has its respective *bias* weight. We run between 200 and 300 epochs with a *batch size* of 20.

Accuracy The overall performance of the models is as follows: 97.76% and 97.93% of correct predictions for the MLP and the RNN, respectively, and for valid hypotheses, the MLP achieved 80.45% and the RNN predicted correctly 82.71% of the test data. The results by length of proofs (table 2) show a counter-intuitive shape: the models have poor accuracy for proofs of length 1, which correspond to the task of recognizing the conclusion in the set of formulas; then, as the length of the proofs goes up, so does accuracy, reaching 100% at length 5, with one exception.

This result is at least partly explained by the exploration of data we performed: the structure of our $\mathcal{KB}$ entails that for long proofs (length ≥ 2), the model sees both the long proof and a number of its sub-proofs in training. Therefore, the longer the proof, the more the model has learned about it, see table 2 for details.

We computed the Hamming distances between the expected output and the rounded up actual output of our NNs.

Example : Let y be the expected output vector (label) for our NN, $\hat{y}$ the actual output vector (prediction):

$$y = [1\ 1\ 0\ 0\ 1]$$
$$\hat{y} = [0.68\ 0.98\ 0.33\ 0.12\ 0.46]$$

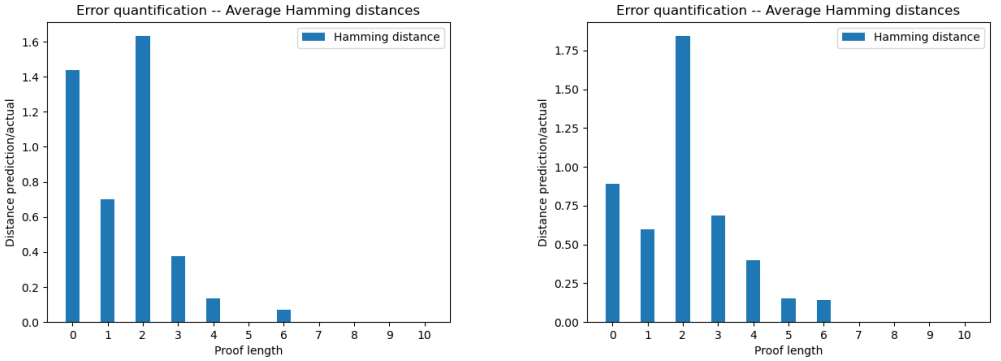
Then $\hat{y}$ rounded is $[1\ 1\ 0\ 0\ 0]$. The Hamming distance between y and $\hat{y}$ rounded is equal to 1, as the two vectors differ only in one of the coordinates, the last one.

Figure 2 shows the average Hamming distances. For proofs of length 1, these were at most 1, which corresponds to only one wrong selected/not-selected formula, which suggests that the NNs do perform the task that is expected of them while getting the formula wrong. For length 2, the distances were at

Length of proofs	MLP test	RNN test	hypothesis	# of sub-proofs	# of formulas
0	99.91%	99.81%	3224	0	0
1	40.00%	50.00%	62	62	62
2	57.89%	63.16%	56	138	194
3	68.75%	75.00%	47	202	362
4	93.33%	93.33%	45	331	695
5	100.00%	100.00%	38	416	1017
6	100.00%	100.00%	40	640	1756
7	100.00%	100.00%	31	645	2002
8	100.00%	90.00%	32	839	2869
9	100.00%	100.00%	26	845	3178
10	100.00%	100.00%	21	827	3384

TABLE 2 – Models achieve better accuracy on longer proofs because of overlaps in training

most 2, and for all larger lengths, the average distances are lower than 0.5, for MLP and RNN alike.



(a) MLP

(b) RNN

FIGURE 2 – Average Hamming distances between the expected output and the actual output for the initial experiment

3.2 Compositionality tests

Then, we explored compositionality tests in the context of our simple corpus, through: (1) variations in the number of formulas needed to prove a hypothesis; (2) permutations in the order of constants. (1) allows us to test the productivity of implication: as implication is transitive, it can be composed; then, with (2), we want to test the capacity of the model to abstract away from the order of constants, which is irrelevant for the derivation rule considered in the study.

For (1), the NNs are trained on proofs from length n_1 to n_2, $n_1 < n_2$; then, we test their performance by predicting all unseen hypothesis from the $\mathcal{KB}$. Test (1) is split in two parts. First, we train the NNs on proofs of length 0 to n_2 and then test them on proofs of length larger than n_2; this is the *unseen*

longer proofs setting. Second, we train the NNs on proofs of length n_1 to 10 and then test them on proofs of length smaller than n_1; this is the *unseen shorter proofs* setting.

Accuracies Table 3 shows the results of test (1) in two settings: 3a when the difference between train and test are the longer proofs, 3b when those are the shorter proofs. For the unseen longer proofs test, we obtained better results by changing the train/test data split only for invalid hypothesis to 20/80%. The models perform better predicting unseen longer proofs than unseen shorter proofs, though in both cases the accuracy goes down quickly. The RNN shows a better accuracy than the MLP.

Train data	MLP test	RNN test
0-9	100.0%	100.0%
0-8	92.06%	93.65%
0-7	73.33%	82.86%
0-6	45.89%	65.07%
0-5	17.0%	46.0%

(a) Unseen longer proofs

Train data	MLP test	RNN test
1-10	44.1%	48.08%
2-10	40.95%	41.61%
3-10	37.5%	40.13%
4-10	7.37%	11.06%
5-10	1.81%	2.16%

(b) Unseen shorter proofs

TABLE 3 – Compostionality tests: variations in the number of formulas

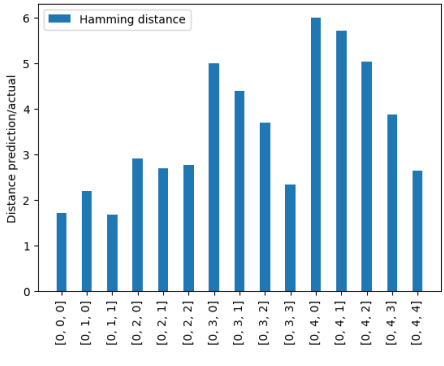
Test (2) is the following: if the model has been trained on a given $\mathcal{KB}$, then a permutation function $\sigma : \mathcal{C} \to \mathcal{C}$ has been applied to $\mathcal{KB}$ to obtain $\mathcal{KB}$', how will the model behave with inputs from $\mathcal{KB}$'? Our first results are straightforward, the models are incapable of adapting to permutation of constants, the accuracy is of 17.84% for MLP, 66.43% for RNN for invalid hypotheses, and 0% for both models for all valid hypotheses, with the exception of a 1.33% for RNN for length 2.

Figures 3 and 4 show the average Hamming distances for experiment (1). In each triple [n1,n2,m] on the x-axis, n1 and n2 stand for the minimal and maximal lengths of unseen proofs, and m corresponds to the length of evidence on which the model is being tested: for example, the triple [0,3,1] corresponds to the model that has not been exposed to proofs of length 0 to 3 in training and is being tested on proofs of length 1. The shape of the curves is similar to the ones for the initial experiment, while the values of the average Hamming distances are quite large. In particular, for unseen lengths 6-10 for tested length 10, where the distance is higher than 8 for MLP, higher than 3.5 for RNN. Thus, the analysis of Hamming distances corroborates the results of the initial experiment.

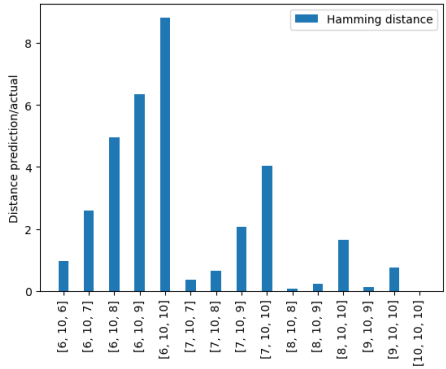
4 Conclusion

Even though NNs appear to be able to pick up some structure from data, our results show that the models have a hard time generalizing it to unseen proof lengths. Moreover, NNs appear to be substantially more sensitive to the order of constants than to the overall structure of the $\mathcal{KB}$. On the other hand, the experiments show an increase in performance for longer seen lengths of proofs, which, correlated with the number of sub-proofs seen by the model in training, suggests some amount of compositionality. Our hypothesis is that the NNs may be able to use information learned on smaller lengths of proofs to improve the performance for larger lengths.

The compositionality tests that we performed show results that are consistent with ones presented in

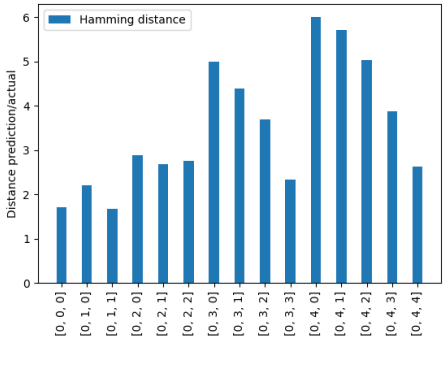


(a) Minimal length of unseen proof: 0

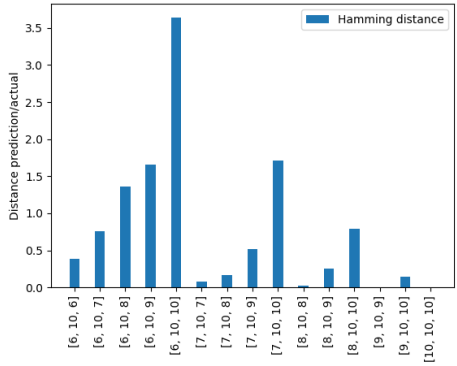


(b) Maximal length of unseen proof: 10

FIGURE 3 – Average Hamming distances between the expected output and the actual output for the unseen length experiment, MLP



(a) Minimal length of unseen proof: 0



(b) Maximal length of unseen proof: 10

FIGURE 4 – Average Hamming distances between the expected output and the actual output for the unseen length experiment, RNN

Hupkes *et al.* (2018). When confronted with sequences longer than the ones they where trained on, the accuracy of NNs from Hupkes *et al.* (2018) drops significantly.

The preliminary study we present here shows that the overall performance cannot be used as the main tool in evaluating the capacity of a model to compositionally select the right premises to prove a given conclusion. The question this observation rises is what is the representation of the data that the NN builds in training? We would like to investigate this through visualisation and diagnostic techniques for NNs such as the ones presented in Hupkes *et al.* (2018).

An interesting remark that has been raised to us is the fact that our dataset is relatively small. Another direction of investigation we are undertaking has to do with this dataset size: for economi-

nal/ecological/ethical reasons, we would like to run our training experiments on the smallest possible datasets while not compromising on the quality of results. Therefore, we are conducting a comparative study through different dataset sizes. Following the same logic, we also investigate other types of encoding. The next step for our research will be its extension to other architectures of NNs, such as Transformers.

Références

BOWMAN S. R., MANNING C. D. & POTTS C. (2015). Tree-structured composition in neural networks without tree-structured architectures. *arXiv preprint arXiv :1506.04834*.

HUPKES D., DANKERS V., MUL M. & BRUNI E. (2020). Compositionality decomposed : how do neural networks generalise ? *Journal of Artificial Intelligence Research*, **67**, 757–795.

HUPKES D., VELDHOEN S. & ZUIDEMA W. (2018). Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, **61**, 907–926.

ONTANON S., AINSLIE J., CVICEK V. & FISHER Z. (2022). Logicinference : A new datasaet for teaching logical inference to seq2seq models. In *ICLR2022 Workshop on the Elements of Reasoning : Objects, Structure and Causality*.

PARTEE B. (1984). Compositionality. *Varieties of Formal Semantics*, **3**, 281–311.

SAXTON D., GREFENSTETTE E., HILL F. & KOHLI P. (2019). Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv :1904.01557*.