Defending Compositionality in Emergent Languages

Michal Auersperger and **Pavel Pecina**
Charles University
Faculty of Mathematics and Physics
{auersperger,pecina}@ufal.mff.cuni.cz

Abstract

Compositionality has traditionally been understood as a major factor in productivity of language and, more broadly, human cognition. Yet, recently, some research started to question its status, showing that artificial neural networks are good at generalization even without noticeable compositional behavior. We argue that some of these conclusions are too strong and/or incomplete. In the context of a two-agent communication game, we show that compositionality indeed seems essential for successful generalization when the evaluation is done on a proper dataset.

1 Introduction

Compositionality is a property of language that describes its specific hierarchical structure. Multiple atomic units (e.g., words) can be combined to produce more complex units (e.g., sentences), while the meaning of the larger units can be inferred from the simpler parts and the way they are combined.

Of course, there are inherently noncompositional structures in language, idioms being a prime example. If someone is *making waves*, it usually means that the person is causing trouble and no water is implied. Still, many reputable researchers (Nowak et al., 2000; Pinker, 2000; Fodor and Lepore, 2002; Lake et al., 2017) have seen compositionality as a key ingredient that enables language to be used productively, i.e., in an infinite number of novel situations. Often, this productivity is juxtaposed with the learning of Artificial Neural Networks (ANNs), whose performance is known to suffer when tested in new scenarios.

Recently, a strand in the literature has been making waves (*sic*) by calling into question the proposed benefits of compositionality. Various papers have shown that ANNs can generalize well to unseen contexts (be productive) even if they work with internal representations that are noncompositional (Kottur et al., 2017; Andreas, 2019; Baroni, 2019; Chaabouni et al., 2020; Kharitonov and Baroni, 2020).

The goal of this paper is to counterbalance these claims.[1] More precisely, we would like to relativize some of the stronger conclusions that are provided and show (by running modified experiments) that the reported experimental results can be harmonized with the view that compositionality is necessary for successful knowledge transfer.

We will do this by first providing an overview of the related research in Section 2. Then in Section 3, we give our arguments against some of the presented assumptions and/or conclusions. These arguments are further supported by experiments that are described in Section 4. We conclude the paper with a short discussion in Section 5.

2 Emergent languages generalize without compositionality

Much of the critical research comes from experimentation with languages that emerge during a communication game. Here, two agents (ANNs) are trained to communicate in order to perform a certain task. The first agent (sender/speaker) encodes the input into a message, a sequence of discrete symbols. The second agent (receiver/listener) does not have direct access to the original input, but only sees it as represented by the message. The receiver's goal is to transform the message into a desired output.

The output can take many forms depending on the task: reconstructing the input fully (Chaabouni et al., 2020; Andreas, 2020) or partially (Kottur et al., 2017) or reconstructing the input after going through some deterministic transformation (Kharitonov and Baroni, 2020).

Inputs can be conceptualized as representations of objects by means of independent nominal at-

---

[1] We consider the cited research valuable and important. In some cases we argue with some statements that were not even the main topic of the paper in question.

tributes, e.g., *blue circle, red square, orange triangle* for two-dimensional input vectors (color and shape). A uniform random sample of all such objects is then held out for testing, the rest of the data (or its part) is used for training.

Andreas (2019) and Chaabouni et al. (2020) develop custom metrics to measure the compositionality of messages passed between the sender and the receiver. The metrics compare each message to the corresponding input and try to assess to what extent each part of the input (attribute value) can be isolated in the message regardless of the context (i.e., other attribute values). They report runs in which models communicate through messages with low compositionality scores, but still achieve good generalization on unseen data.

Kharitonov and Baroni (2020) replace training the first agent (sender) by hand-coding the messages. This gives them the advantage of having direct control over the emergent language. They find that good generalization can sometimes be achieved using a non-compositional language (sometimes leading to even better results than using a compositional language).

3 Allowing compositionality to have an effect

Our main argument is that the process of selecting the test data plays a crucial role in evaluating the effects of compositionality. Sampling examples with uniform probability is a mainstay in machine learning, and algorithms have been shown many times that they can perform very well under these conditions. However, once we move away from the static world of i.i.d. data samples into the dynamic world of ever-changing distributions, the limitations of such models become obvious.

We argue that this is where compositionality is supposed to be helpful. Analyzing the world (or data points to keep the discussion down to earth) through a hierarchy of parts and their relations enables inferring a 'rule-based algebraic system', which is 'an extremely powerful generalization mechanism' (Baroni, 2019). Systematic compositionality exploited by human learners enables them to be sample efficient, i.e., quickly learn a new task seeing just one or a limited number of training examples (Lake et al., 2017). Therefore, we conclude that showing compositionality not being correlated with generalization on in-domain held-out data is not very informative. Instead, it is

preferable to control the exposition of certain patterns in the training and testing data as illustrated by, e.g., the SCAN benchmark (Lake and Baroni, 2018).

We also point out that Andreas (2019) and Chaabouni et al. (2020) implement both agents (sender and receiver) as relatively standard encoder/decoder architectures with recurrent (LSTM, GRU) layers. These architectures are not necessarily known for their ability to produce or utilize compositional sequences.

The assumption seems to be that when running an experiment many times (with different random initialization of the models), some runs will be successful in the sense that the agents will develop a more or less compositional language by chance. Moreover, the degree of compositionality would have to be large enough to influence the generalization of the models (should such an effect be real). We do not think this assumption is justified.

Kharitonov and Baroni (2020) avoid a part of the above problem by creating compositional messages manually. Their main conclusion is that one can devise different tasks, and in some of them a compositional representation of the input data might even prove disadvantageous. Indeed, it is possible to create, for example, an arbitrary bijective function from the input space to the output space and train a model to learn such a mapping. Not surprisingly, such a model will fail at the test time. However, if we first encode the input with the same arbitrary transformation (thus creating a non-compositional representation of the input data), the model is then asked to learn the identity function, which it might achieve quite well. Therefore, we agree with the conclusion that 'in isolation from the target task, there is nothing special about a language being . . . compositional.'

Yet, as mentioned above, compositionality is often discussed in conjunction with natural language or human cognitive abilities more generally. In the lived experience of biological species (to be less human-centric), it is reasonable to expect that the 'underlying factors of variation' or 'explanatory factors' (Bengio et al., 2013) behind the input data a) repeat in many different situations and b) are directly relevant for different tasks. For example, it is reasonable to expect that many languages have a word for *water* rather than a word for *water and the wind is blowing*, simply because the first is more useful. Therefore, we believe that the em-

Model	train	in-domain test	out-of-domain test
sender (ours)	1.00 ± 0.01	1.00 ± 0.00	$\mathbf{0.99 \pm 0.06}$
sender (Chaabouni et al., 2020)	1.00 ± 0.04	1.00 ± 0.03	0.83 ± 0.19
receiver (ours)	1.00 ± 0.00	1.00 ± 0.00	$\mathbf{1.00 \pm 0.02}$
receiver (Chaabouni et al., 2020)	1.00 ± 0.02	1.00 ± 0.00	0.44 ± 0.32

Table 1: Learning alone experiment: Comparing the accuracy (mean ± std over 20 runs) of two types of architecture in different data splits. Each agent (sender/receiver) is trained independently of the other by using fixed messages.

phasis on compositionality in research can be more than 'a misguided effect of our human-centric bias' (Kharitonov and Baroni, 2020).

4 Experiments

We follow the communication game experiments of Chaabouni et al. (2020). We create a set of instances, each of which is represented by i_{att} attributes. Each attribute has n_{val} possible values. The messages passed between the agents are limited by the maximum length (c_{len}) and the size of the vocabulary (c_{voc}). The receiver's goal is to reconstruct the input.

As messages are sequences of discrete symbols, which prevents gradients from passing through, the sender must be trained with the REINFORCE algorithm Williams (1992). The receiver is trained using backpropagation. We use the EGG toolkit (Kharitonov et al., 2019) to implement the experiments.[2]

We focus mainly on the setting of ($i_{att} = 2$, $n_{val} = 100$, $c_{len} = 3$, $c_{voc} = 100$). In this case, the dataset contains isntances such as $(12, 34)$, $(0, 99)$, $(99, 0)$ etc. We create three splits. The *out-of-domain (OOD) test set* contains all pairs where 0 appears, apart from three examples: $(0, 0)$, $(0, 1)$, and $(1, 0)$. The *training set* contains these three zero examples together with 90% of the remaining (nonzero) examples (random sample). The rest of the data constitute *in-domain (IND) test set*. In other words, we designate a special symbol (0), which appears only in a limited number of contexts in training. We then separately evaluate how the models perform on unseen examples with ordinary symbols (IND test set) and on unseen examples with the special symbol (OOD test set).

Given the absence of any incentive for the models to develop compositional messages during training, we opt for architectural biases in our experiments. We use the models that have been proven to be successful in OOD generalization in the SCAN

benchmark (Li and Bowling, 2019; Russin et al., 2020; Auersperger and Pecina, 2021). Each agent is implemented as a separate seq2seq encoder-decoder architecture with recurrent layers and a modified attention mechanism. Details are provided in Appendix A

4.1 Learning alone

We first want to know whether the architectures used for implementing the agents are capable of achieving systematic compositionality on their own, that is, outside of the context of the 2-agent communication game. To do this, we handcode the messages that are to be created (sender) or received (receiver) and train each agent using regular backpropagation on the corresponding task.

We first create an arbitrary bijective mapping from the input vocabulary to the message vocabulary. Furthermore, to introduce variability in the length of the messages, we duplicate the occurrences of all tokens with odd indices in the message vocabulary. For example, having a mapping $tr : 0 \rightarrow 3, 1 \rightarrow 8, 2 \rightarrow 2, \ldots$, we produce (among others) the following input–message pairs: $(0, 0) - (3, 3)$, $(1, 0) - (8, 8, 3)$, $(2, 1) - (2, 2, 8, 8)$.

We train 2 types of agent for both sender and receiver comparing our architecture with the default used by Chaabouni et al. (2020). There are 500 training epochs and 20 runs with different random initializations for each agent. The results are presented in Table 1. They show that at least for this simple task, adding architectural bias helps with compositional generalization to out-of-domain data. The results also suggest that the default architectures are unlikely to provide systematic generalization in the communication game since they are unable to achieve it even if their communication partner does never make mistakes (i.e. the messages are guaranteed to be produced/received correctly).

We experimented with limiting the capacity of the default architecture to see if such kind of regularization could help with generalization perfor-

---

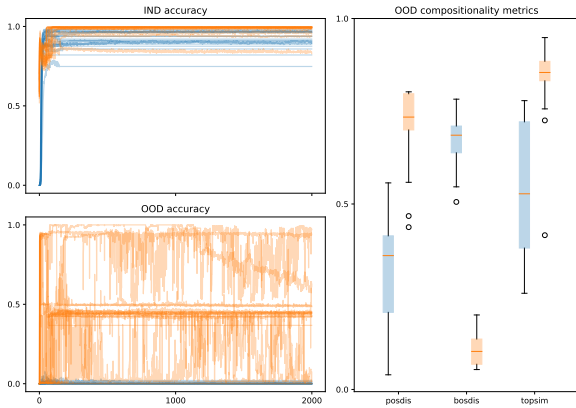[2]The code is available at https://github.com/michal-au/emlang-compos.git

Figure 1: Communication game experiment: Training and OOD accuracy during training. Compositionality measures at the end of training. Orange represents runs of our architecture, blue represents runs of the architecture used by Chaabouni et al. (2020). There were 20 runs for each architecture.

mance. Besides the original size (500), four additional sizes of hidden layers were tested (100, 200, 300, 400). For the default receiver, a smaller capacity (100) improved the OOD accuracy from 0.44 (± 0.32 std) to 0.81 (± 0.21 std).

We point out that it is the out-of-domain test set that reveals the difference between the architectures.

4.2 Communication game

Having seen that in some tasks our architectures are capable of approaching systematic compositionality, we turn our attention to the full communication game. We train both the default and our modified architectures on the full task for the maximum number of 2,000 epochs. Similarly to Chaabouni et al. (2020), we use early stopping when training accuracy reaches 99.999%, however, we evaluate all the runs, even those that never reach perfect training accuracy. Each experiment was repeated 20 times with different random initializations. The training progress is visualized in Figure 1 and the results are given in Table 2.

The experiments demonstrate that our changes to the default architecture lead to some out-of-domain generalization, but we were unable to guarantee such behavior for each run (accuracy 0.42 ± 0.27 std). In contrast, the original architecture does never succeed (accuracy 0.00 ± 0.01 std).

We evaluated the three compositionality metrics used by Chaabouni et al. (2020) and found significant differences between the two architectures. In two of the metrics, namely *positional disentangle-*

ment and *topographic similarity*, we achieve higher scores than the original architecture, while in *bag-of-symbol disentanglement* the situation is reversed. We show the relationship between compositionality and generalization in Figure 2.
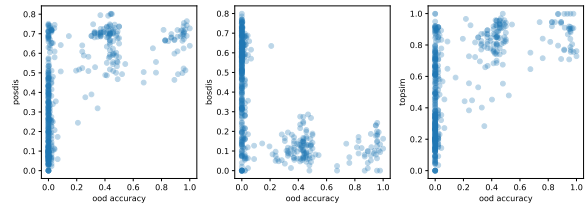

Figure 2: Compositionality measures and generalization in out-of-domain dataset. Successful generalization was possible only with large posdis and/or topsim scores.

Both Andreas (2019) and Chaabouni et al. (2020) claim that compositionality is not a necessary condition for good generalization, but that it might be a sufficient condition (Chaabouni et al., 2020). Choosing the out-of-domain data for evaluation and training models whose architecture is biased towards utilizing composationality of a language, we arrive at the opposite conclusion: compositionality is a necessary but not sufficient condition for good generalization. In other words, we often observe runs where both agents communicate through relatively compositional messages, but fail to generalize. However, we never observe a run where generalization is successful in spite of a low compositionality (posdis or topsim) score.

However, given the size of the input space (100×100) and the proportion of training data (about 90%), we did not expect to find such a noticeable difference in performance in the in-domain test set. This suggests that such test data is not completely agnostic to the notion of compositionality (which is somewhat contrary to our previous argumentation). Yet, we still maintain that the out-of-domain dataset is much more informative with respect to evaluating the benefits of compositionality.

Similarly to the previous experiment, we tested additional sizes of hidden layers of the original architecture (100, 200, 300, 400) but were not able to match the IND accuracy of our architecture.

Looking at the OOD generalization performance of our models, it is notable that most models operate in one of three regimes: the accuracies tend to cluster above 0, below 0.5, or below 1. Manual inspection of the agents' communication showed that most of the time agents successfully reconstruct the

Model	train	IND test	OOD test	posdis	bosdis	topsim
Chaabouni et al. (2020)	0.99 ± 0.07	0.91 ± 0.09	0.00 ± 0.01	0.32 ± 0.15	0.67 ± 0.07	0.53 ± 0.18
ours	0.99 ± 0.03	0.98 ± 0.05	**0.42 ± 0.27**	0.71 ± 0.11	0.11 ± 0.04	0.83 ± 0.11

Table 2: Communication game experiment: Accuracy (mean ± std) measured in different data splits, Three compositionality measures of the messages (mean ± std) evaluated in out-of-domain test data.

non-zero symbols, which means that most of the errors are caused by wrongly reconstructing the zero symbol. These errors are also systematic, meaning that, given the position in the string, the zero symbol is replaced by the same symbol regardless of its neighbor. Thus, agents successfully reconstructing zero at both positions achieve accuracies close to 1, agents successfully reconstructing zero only in a single position achieve scores close to 0.5 and in the rest of the runs, agents fail regardless of the position.

5 Discussion

There are many questions that remain for further analysis. The distinction between the in-domain and out-of-domain data is not clear-cut. One might object that seeing just one or two examples of a given symbol in the training data is too little for ANNs to learn its embedding and reliably map it close to other 'similar' symbols in the semantic space (Lake and Baroni, 2018; Loula et al., 2018). This is actually the issue as it seems that human learners unlike ANNs are able to succeed in such a scenario and work with limited data or, in other words, 'not-yet-converged embeddings'. See Lake et al. (2017) for a more thorough discussion.

The goal of this paper was to show that some conclusions in the literature on compositionality are too strong or incomplete. However, there are other arguments that remain untackled. Baroni (2019) gives examples of neural networks that generalize (partially) well to out-of-domain data. For instance, Dessi and Baroni (2019) show that a simple convolutional network is enough to improve accuracy from 1.2% to 60% in a difficult task from the SCAN benchmark. Gulordava et al. (2018) demonstrate that a language model is capable of preferring grammatical nonsense sentences (certainly not seen in training) to ungrammatical ones. In general, the practical success of ANNs in many applications can serve as a proof of their strong generalization abilities (Lake and Baroni, 2018; Baroni, 2019).

In response, we would like to point out that such success often coincides with new developments in neural architectures (convolutional NNs in vision,

attention in NLP). These developments might actually point in the direction of compositionality. A trained convolutional NN actually detects primitive shapes (at least by the filters in the lower layers) and combines these into composit representations. Similarly, a trained attention-based encoder-decoder language model represents each input as a sequence of contextualized embeddings of the original units. Some of these embeddings might primarily represent the corresponding input units, and others might represent their collections.

We also acknowledge that for practical applications, especially in the short term, focusing on compositionality is not guaranteed to help (Kharitonov and Baroni, 2020). The most practical way so far has been to enable training on as much data as possible. However, it is likely that such an approach will eventually result in diminishing returns.

There are many potential aspects that favor compositional behavior: inductive architectural biases (e.g., attention); limited channel capacity relative to the input space (Nowak et al., 2000); ease of transmission within the population (Chaabouni et al., 2020); generalization pressure (contrary to Chaabouni et al. 2020), if such a pressure is allowed to have some effect (e.g., by meta-learning); .... It is also possible that systematic compositionality might emerge[3] only as a result of multiple such factors.

Acknowledgements

References

Jacob Andreas. 2019. Measuring compositionality in representation learning. In *International Conference on Learning Representations*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual*

---

[3]As is often the case, *emerge* here means *somehow come about* and is used to hide the fact that the authors cannot explain things any further.

Meeting of the Association for Computational Linguistics, pages 7556–7566, Online. Association for Computational Linguistics.

Michal Auersperger and Pavel Pecina. 2021. Solving SCAN tasks with data augmentation and input embeddings. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 86–91, Held Online. INCOMA Ltd.

Marco Baroni. 2019. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Roberto Dessi and Marco Baroni. 2019. CNNs found to jump around more skillfully than RNNs: Compositional generalization in seq2seq convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3919–3923, Florence, Italy. Association for Computational Linguistics.

Jerry A. Fodor and Ernest Lepore. 2002. *Compositionality Papers*. Oxford University Press UK.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Eugene Kharitonov and Marco Baroni. 2020. Emergent language generalization and acquisition speed are not tied to compositionality. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 11–15, Online. Association for Computational Linguistics.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on emergence of lanGuage in games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 55–60, Hong Kong, China. Association for Computational Linguistics.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882, Stockholmsmässan, Stockholm Sweden. PMLR.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253.

Fushan Li and Michael Bowling. 2019. Ease-of-teaching and language structure from emergent communication. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

João Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Martin A. Nowak, Joshua B. Plotkin, and Vincent A. A. Jansen. 2000. The evolution of syntactic communication. *Nature*, 404(6777):495–498.

Steven Pinker. 2000. Survival of the clearest. *Nature*, 404(6777):441–442.

Jacob Russin, Jason Jo, Randall O'Reilly, and Yoshua Bengio. 2020. Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 313–327, Online. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

A Appendix

A.1 Architecture

Both the sender and the receiver are implemented by the following encoder-decoder architecture:

The **encoder** produces two embeddings (size 500) for each input symbol, one syntactic and one semantic. A uni-directional GRU (Chung et al., 2014) layer (size 500) transforms the syntactic embeddings to contextualized embeddings.

The autoregressive **decoder** embeds the last produced symbol (or the start-of-sequence symbol) (size 500) and transforms it with another GRU layer (size 500). This contextualized embedding is then used as a query in the dot-product attention (Luong et al., 2015) and matched against the contextualized embeddings produced by the encoder. The attention weights are then used to produce the weighted sum of the semantic embeddings of the input symbols. This vector (size 500) is added to the query and transformed by a linear layer to the output symbol logits.