EFFECT OF PRESSURE FOR COMPOSITIONALITY ON LANGUAGE EMERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans use natural language compositionally to communicate complicated ideas using expressions grounded in simpler concepts. Such generalizing behaviors of natural language usage make it safe to assume compositionality is beneficial for language emergence. Several recent works, which employ methods such as generational transmission, have induced compositionality in their communications. Nevertheless, such schemes are complex to implement and can be time-consuming in training, due to additional steps integrated for specifically inducing compositionality. This paper presents a learning environment, where agents are pressured to make their emerging languages compositional, by incorporating a metric of topological similarity into the loss function. Our proposed method, which does not require supervising examples, is straightforward and can easily be integrated into any existing language emergence environment without any additional stages. We observe that agents can achieve higher generalizations and convergence speeds when this pressure is carefully adjusted, depending on the environment parameters. For a given level of generalization, increased compositional pressure makes language transmission to new learners significantly easier. Furthermore, we find that a situational correlation, between generalization and compositionality, exists even in the absence of external pressure.

1 INTRODUCTION

Traditional language processing has been an enormous success and employs large amounts of textual examples to build statistical relationships. However, language grounding and capturing of functional aspects achieved by such methods are questionable (Ren et al., 2020; Lazaridou et al., 2017; Mordatch & Abbeel, 2018; Lazaridou et al., 2018). Language emergence is an alternative to supervised approaches, where neural agents develop communications by themselves without being exposed to explicit supervising examples. Agents are deployed in a partially observable environment, such that specific agents are unable to complete their assigned tasks without gaining information from their peers (Jorge et al., 2016). Agents should develop their own language to acquire information on what they cannot observe. Hence, the emerged communications, which are born out of necessity, tend to be more pragmatic.

Agents map environmental entities or concepts with symbols while reaching a consensus for the mapping. If agents follow compositionality, agents would connect individual symbols to atomic concepts in the environment, and combine multiple symbols to form composite messages for expressing complex novel concepts, facilitating generalization. For example, *Red Square* is a compositional expression, containing attributes *Color* and *Shape*, constructed by using atomic concepts *Red* and *Square*, which corresponds to the values taken by the two attributes. Hence, whenever an agent observes a novel input, it can break down the input into a set of attributes and corresponding values. An expression describing the input can be constructed by suitably concatenating the symbols representing each value of the attributes.

It is still in debate whether compositionality is an essential or a desired property in language emergence. The simple reason for favoring compositionality is that it enables expressing complex concepts using simpler ones, implying a higher degree of generalization for unseen data. Moreover, there is strong evidence (Kirby & Hurford, 2002; Kirby et al., 2014; 2015) citing this property as advantageous for language acquisition. Such ideas and proposals have led several studies to promote compositionality as a required attribute in language emergence (Ren et al., 2020; Lazaridou et al., 2018). Nevertheless, some others (Chaabouni et al., 2020) have shown empirical evidence that, for the case of language emergence in neural agents, compositionality is not related to generalization.

(Chaabouni et al., 2020; Gupta et al., 2020; Resnick et al., 2020) have studied the relationship between compositionality and several other parameters like generalization, bandwidth, and agent complexity. (Ren et al., 2020) propose a model based on the generational transmission that favors compositional languages. They demonstrate the increased learning speed of compositional languages and propose the existence of a strong correlation between compositionality and validation performance.

Previous works discuss the usage of functional pressures in neural language emergence. (Chaabouni et al., 2019) found that agents develop anti-efficient encoding in emergent communications without complying with the principle of least effort as described by the Zipf law (Zipf, 2016). Message distributions begin to follow the Zipf law when the cost function includes a penalty for longer messages. (Choi et al., 2018) study compositional obverter technique, stressing that what speaker agents are transmitting should also be understandable to themselves. Obverter technique observes Zipf's law, making communications more efficient, as stated by the latter study. (Kirby & Hurford, 2002) imposes a similar constraint for following the principle of least effort by selecting the shortest from a set of generated strings.

Being motivated by such studies, we incorporate an auxiliary loss based on topological similarity into the training process, which pressures agents to make their messages compositional. It is different from (Chaabouni et al., 2020) where there is no such external functional pressure. (Kirby & Hurford, 2002; Kirby et al., 2014; 2015) discuss cultural evolution of language and propose compositionality as a quality stemming from iterated learning, where successive generations acquire language skills from the previous generations.(Ren et al., 2020) uses an iterated learning approach yielding languages with higher generalizations and learning speed advantages. Their method has multiple phases for learning, interacting, and transmission, which is resource-consuming. We do not employ generational transmission or additional learning phases other than the main reconstruction game. Our agents acquire compositionality within the learning phase of a single life cycle. Hence, the proposed approach is simple to implement and significantly time efficient.

We conduct experiments using inputs with varying degrees of complexity, and agents built with different neural architectures, to study agents describing a given input over a discrete communication channel. We externally constrain the structure of messages that agents can send through the channel. We measure agent performance during testing as an indicator for generalization. First, we study the relationship between compositionality and generalization subjected to agent architecture, message structure, and input complexity without providing any external influence. Then, we train agents with our proposed auxiliary loss, pressuring neural agents to various extents to adhere to compositionality in emerging communication protocols. We expect that agents trained under suitable compositional pressures will exhibit improvements over the baseline during the testing.

Our results show that improving generalization and convergence speed is possible by carefully adjusting the compositional pressure. The optimal pressure depends on the message structure, input space, and agent architecture. Furthermore, training under high compositional pressures emerge languages that are substantially easier to learn by new learners. We observed a situational correlation between compositionality and generalization even when agents learn under no external pressure. In general, well-tuned external pressure for compositionality gives improved results on average.

We summarize the contributions of this paper as follows,

- We propose a straightforward method to externally induce compositionality, which can increase generalization, convergence and transmission speed in language emergence.

- We find that, there is a situational correlation, which can be considerably strong depending on the environment parameters, between compositionality and generalization, even without external pressure.

2 LANGUAGE EMERGENCE GAME

There are several widely used environment configurations in use, most of which are inspired by the Lewis signaling game (Lewis, 2008). (Sukhbaatar et al., 2016; Mordatch & Abbeel, 2018) used a simulating environment where multiple agents can navigate in a 2-D world while coordinating with each other by exchanging discrete symbols. (Lazaridou et al., 2017; Havrylov & Titov, 2017). (Havrylov & Titov, 2017; Evtimova et al., 2018; Jorge et al., 2016) employ a ”discriminating” objective where one of the agents has to correctly identify a reference object apart from a set of distractors by listening to its peer who has a copy of the reference. (Kharitonov et al., 2020) propose ”classification” type game. In such variations speaker, who gets an image, transmits discrete signals to the receiver. The receiver then determines the class of the image referenced by the Speaker. (Resnick et al., 2020; Gupta et al., 2020) follows the ”reconstruction” game setup, where Listener has to approximate the input given to the Speaker.

Discriminating, classification, and reconstruction games frequently but not always use a value attribute environment. In such environments, inputs consist of a set of abstract attributes, and each attribute can take a finite set of values. The whole dataset is a collection of abstract vectors, which are free from noise. Such data is easier to generate, and one can easily control the dataset complexity by changing the number of attributes and values.

(Kharitonov et al., 2020; Bouchacourt & Baroni, 2018) discuss how agents can converge to a simpler protocol without capturing high-level features in a discrimination game. For example, if images are used, agents can use the average pixel intensities of the images to identify the correct image. In a value-attribute environment, such behaviors cause agents to have an excessive degree of freedom. Achieving high performance is still possible if the complexity of the distractor set is low. Hence, the final accuracy may not directly indicate a rich communication protocol or a vocabulary. Therefore, in our work, we use the reconstruction objective (Resnick et al., 2020; Gupta et al., 2020) to remove the possibility of such events.

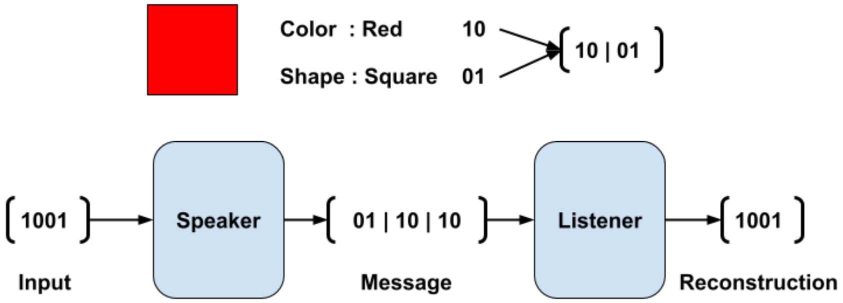


Figure 1: A Reconstruction game of 2 attributes with 2 values per each. Vocabulary consists of 2 symbols and messages has a length of 3. Each symbol in the message, and values of the attributes are represented as one-hot encoded vectors.

2.1 SETUP

Our game consist of two agents *Speaker* and *Listener*. First Speaker observes an object and transmits a single discrete message to the Listener. After reading the entire message, Listener tries to predict the object seen by the Speaker. If Listeners prediction is correct both agents are rewarded (see Figure 1). Our experiments use abstract objects from the commonly used value attribute environment (Resnick et al., 2020; Gupta et al., 2020; Ren et al., 2020). Each object is described by a set of attributes, and each attribute can take a single value from a set of values.

We create an abstract object $x \in \mathbb{X}$ composed by a set of attributes $a \in \mathbb{A}$ where each attribute takes a value $v \in \mathbb{V}$. We externally control the number of attributes $|\mathbb{A}| = N_A$ and the number of values per attribute $|\mathbb{V}| = N_V$. Hence, the total objects in $\mathbb{X}$ is $|\mathbb{X}| = {N_V}^{N_A}$. For a given object, we iterate through all of its attributes and obtain the value of each attribute. Next, each value is converted to a one-hot encoded vector of size N_V. Then all converted vectors are concatenated together to obtain the final representation of the object. For simplicity, we keep the number of values for each attribute the same. Hence, to create an object, we sample N_A random integers from a range of $0 - (N_V - 1)$ and concatenate the one-hot encoded representations of the sampled integers. (Chaabouni et al., 2020; Ren et al., 2020; Resnick et al., 2020) use environments with attributes of similar values, where the instances from different studies are identical if the number of attributes and values are the same. Nonetheless, agents' internal representations can be different depending on the occasion.

In the game, Speaker, after observing the input x, transmit a discrete message $m \in \mathbb{M}$ towards a Listener. After reading the message, Listener reconstructs an approximation x' for the original object. To construct the message, Sender repeatedly samples symbols or words $w \in \mathbb{W}$ from a finite-sized vocabulary with replacement until the message reaches its maximum allowed length L. Sampled symbols are concatenated together in order to construct the message m. Similar to inputs, each symbol is represented as a one-hot encoded vector. Then the whole message is transmitted to the Listener. The maximum number of unique messages Speaker can construct, or the message space capacity $|\mathbb{M}|$ is equal to $|\mathbb{W}|^L$. For convenience, we denote the input spaces and message space as tuples, where $X(V, A)$ and $M(W, L)$ denoting data spaces with V values, A attributes and message spaces with vocabulary size W and maximum message length L.

2.2 Agents

Sender is modeled by a LSTM cell (Hochreiter & Schmidhuber, 1997) and two MLPs. First, the input x is fed to a linear layer, and the output vector is treated as the initial hidden state and the cell state of the LSTM cell. Next, the updated hidden state of the LSTM cell is mapped to a set of logits by the second linear layer. Then the logits are used to sample a symbol from the vocabulary with replacement, and the sampled symbol is fed back to the LSTM cell as input when sampling the next symbol.

We employ two types of networks, linear and recurrent, for modeling the Listener. The Sender's message is fed to the Listener as a whole. The output logits of the Listener are used to obtain a probability vector that spans over all the attributes. It indicates the value of each attribute that constitutes the input given to the Sender. We use backpropagation in our experiments and use Gumbel-Softmax (Jang et al., 2017) approximation to make discrete messages differentiable during the backward pass. We use cross-entropy loss between the Listener's output distribution and Sender's input to measure the reconstruction loss $\mathcal{L}_r$. The compositional loss $\mathcal{L}_c$ is implemented only on the Sender, and it is added on top of the reconstruction loss for backpropagation (see Appendix A). The constant C_t defines the strength of the compositional pressure, which we vary during our experiments.

3 Metrics for Compositionality

There has been much debate about how to measure compositionality in language emergence. Although there is no universally agreed method, several studies have proposed a set of valuable metrics. For example, (Chaabouni et al., 2020) includes a measure based on topological similarity and two other intuitive measures of disentanglement, such as positional disentanglement and bag of words disentanglement. Despite having multiple intuitively plausible alternatives, metrics depending on topological similarity have been more widely used (Brighton & Kirby, 2006; Ren et al., 2020; Lazaridou et al., 2018). Consequently, we use topological similarity in all of our experiments. We first obtain all pairwise combinations of the inputs and their corresponding messages through a paring operation defined by two matrices. Distances between inputs and messages are calculated for each pair in the two groups. Then the topological similarity is defined as the Pearson correlation between distance values in two groups.

Matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in algorithm 1 in Appendix A) are used to create the pairwise formations, which solely depends on the batch size. For inputs $x_1, x_2, x_3, x_4, \ldots, x_n$, there should be $(n(n-1))/2$ pairs :$(x_1, x_2), (x_1, x_3), (x_1, x_4), \ldots, (x_{n-1}, x_n)$, depicting all possible paring arrangements. For inputs of batch size n, $\boldsymbol{A}x_n$ and $\boldsymbol{B}x_n$ yields paring vectors, which denotes elements at the first position $x_1, x_2, \ldots, x_{n-1}$ and second position $x_2, x_3, \ldots, x_n$ of all the pairs (Refer Appendix D).

We do not directly use Sender's inputs and messages in our method. Instead, we use the initial hidden state of the Sender and samples from the relaxed categorical distribution (see Algorithm 1). We use cosine similarity (Lazaridou et al., 2018) to measure the distance $\cos(i, j)$ between elements in each

Table 1: Correlation between generalization and compositionality varies with the agent architecture and input. Instances with high training accuracy shows much stronger correlation.

Train Accuracy	Recurrent		Linear		$X(10,4)$		$X(50,2)$	
	ρ	p	ρ	p	ρ	p	ρ	p
0.5	0.619	9.86e-7	0.197	0.123	-0.084	0.555	0.393	1.2e-3
0.7	0.638	7.87e-7	0.269	0.080	0.017	0.913	0.736	2.2e-16
0.8	0.738	2.69e-7	0.923	1.82e-14	0.614	1.82e-4	0.769	2.2e-16
0.95	0.883	5.07e-8	0.918	3.50e-13	0.633	4.72e-3	0.809	1.19e-7
0.98	0.898	1.81e-7	0.913	1.97e-12	0.588	2.12e-2	0.809	2.62e-7

pair. Since the maximum value of cosine distance is equal to one, we consider $d(i,j) = 1 - \cos(i,j)$ as the distance. Finally, we calculate the Pearson correlation coefficient $\rho_{h_0,m}$ between distance vectors of inputs and messages (see Algorithm 1) as an indicator for topological similarity. If the Spearman coefficient is used, its monotonicity is less restrictive than the requirements for a linear relationship in the Pearson correlation coefficient. However, Pearson correlation can be implemented as a differentiable metric, which releases the burden of implementing a differentiable approximation for the ranking operation required by the Spearman correlation.

4 Experiments

4.1 Compositionality and Generalization without External Pressure

(Chaabouni et al., 2020) has discussed the effect of compositionality on generalization within environments where there is no external pressure. While they find no correlation between these two parameters, we find that it is possible to have a situational correlation with strong evidence which is impossible to be discarded. The relationship's strength depends on the Listener architecture and how well agents succeed during their training phase. Table 1 depicts the Spearman correlation, with their corresponding p values for 5 levels of training accuracy.

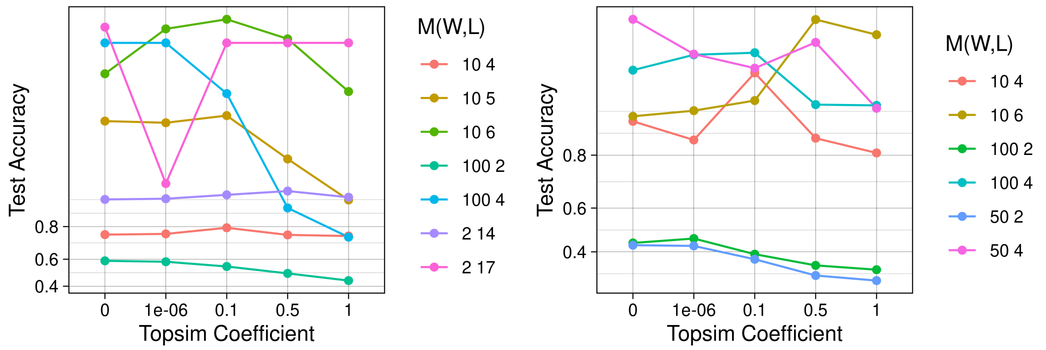
Recurrent Listeners display significant ($p < 9.86e - 7$) correlation under all levels of training, where linear models show the same strength only under the highest levels of test accuracy. For a recurrent Listener, processing of the message occurs sequentially, and high compositionality allows information to be distributed along the message instead of collapsing to a single position, hence explaining the latter behaviour.

Regardless of the Listener type, all models exhibit strong relationships when they reach high test accuracy. Furthermore, the input space also affects the degree of correlation, where instances of $X(50,2)$ show higher correlation over $X(10,4)$. In general, large number of input examples can improve the performance of most models. In the presence of increased data, agents may find turbulent but generalizing representations. However, when the data is scarce, and if the training is successfully, generalization should be closely associated with structured representations.

4.2 Generalization with External Compositional Pressure

Figures 2 and 3 shows the behaviour of test accuracy with various external pressure constants C_t. For some settings, there is a significant improvement of test accuracy (see Appendix B). We further observe increased convergence speeds for all constants $C_t > 0$, if they give better generalizations over the baseline. For some configurations, there is no C_t among the selected values giving higher or equal generalization: $\{X(10,4), M(100,2)\}$, $\{X(50,2), M(50,2)\}$ for linear Listener, and $\{X(10,4), M(100,2)\}$ for recurrent Listener are the only three seen in the experiments. Optimal value changes across different parameters, like input space and message structure, so there can be coefficients that can show better performance for the above configurations, but we do not further investigate such instances specifically.

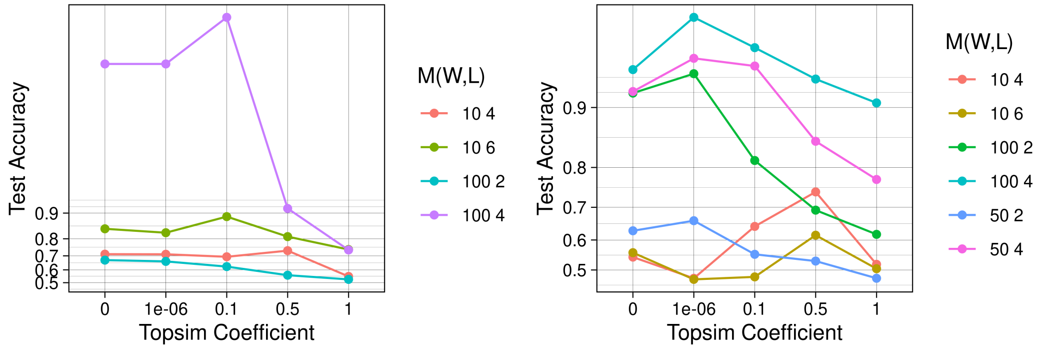
Compositional pressure causes both positive and negative effects (see Figure 2 and 3). At $C_t = 1$, we do observe rapid deterioration of the performance. Such cases denote that generalization can be



(a) Linear; $X(10, 4)$

(b) Linear; $X(50, 2)$

Figure 2: Variation of test accuracy with the topological similarity constant C_t for linear Listener.



(a) Recurrent; $X(10, 4)$

(b) Recurrent; $X(50, 2)$

Figure 3: Variation of test accuracy with the topological similarity constant C_t for recurrent Listener.

worse than the baseline performance if a pressure that does not match the input space and the message structure is applied to the agents. A relating behavior is reported in (Chaabouni et al., 2019), they incorporate a message length regularizing term to the loss function, which causes emergent messages to follow Zipf's law more strictly. However, they noticed slower convergence by adding this term, with a lesser number of successful runs. Section 4.2.2 further discuss this issue. Nevertheless, graphs still demonstrate that compositional pressure offers a situational advantage when carefully matched with other environmental parameters. Although $C_t = 1$ seems to be of no use, we show in Section 4.2.3 that such pressures can indeed be advantageous when considering the transmission speed of the emerged languages.

In $X(10, 4)$, which accounts for an input space with 10^4 distinct examples (see Figure 2 and 3). The optimal values for C_t are different from the $X(50, 2)$ experiment, signaling that the compositional pressure is dependent on the structure of the input data. For a given message type, with linear Listeners, the optimal accuracy values are notably higher than in the first experiment, generally for all C_t. We assume an increased number of data samples to be the reason behind the improved accuracy.

In appendix C we report the results we obtained by trying to improve the baseline through other regularizing techniques. We used $X(50, 2)$ setting with a recurrent Listener. For the case of dropout, we observe no increase of test accuracy that rivals the results obtained through compositional pressure. Weight decay have improved results for some message structures slightly $[M(100, 2), (100, 4)]$, but display degraded performance for the other configurations $[M(10, 4), (10, 6)]$. Even though, compositional pressure and above regularizing techniques can improve or worsen generalization, based on their applied strength, we find a crucial difference in our method. Dropout and weight decay operate at the level of neural network structure, considering either network weights or connections, whereas

Table 2: Effect of external pressure under different learning rates. $C_t = 0$ represent the baseline performance. $C_t = 1e-6$, for $M(100,2), M(100,4)$ and $C_t = 0.5$ for $M(10,4), M(10,6)$ when $C_t > 0$.

$X(W, L)$	Test Accuracy			
	lr = 0.005		lr = 0.05	
	$C_t = 0$	$C_t > 0$	$C_t = 0$	$C_t > 0$
$M(10,4)$	0.726 ± 0.053	0.750 ± 0.073	0.125 ± 0.228	0.211 ± 0.126
$M(10,6)$	0.665 ± 0.051	0.870 ± 0.005	0.328 ± 0.122	0.228 ± 0.108
$M(100,2)$	0.779 ± 0.023	0.833 ± 0.046	0.820 ± 0.046	0.858 ± 0.079
$M(100,4)$	0.973 ± 0.011	0.975 ± 0.012	0.025 ± 0.031	0.025 ± 0.000

our compositional pressure concerns only the representations used in the discrete link. Hence, they are not operationally identical and can be implemented on top of each other. We do not delve into this matter and leave it as future work.

4.2.1 External Compositional Pressure Under Different Learning Rates

To further validate the effect of external pressure, we repeat several of our experiments in $X(50,2)$ input space, with different learning rates. Table 2 shows test results for two different learning rates. For $C_t > 0$, we select coefficients best performed in the earlier experiments under the same message structure and input space. Training under positive pressure has consistently improved the generalization. The only exception occurs at $M(10,6)$ at 0.05 learning rate.

4.2.2 Effect of Input and Message Space Structure

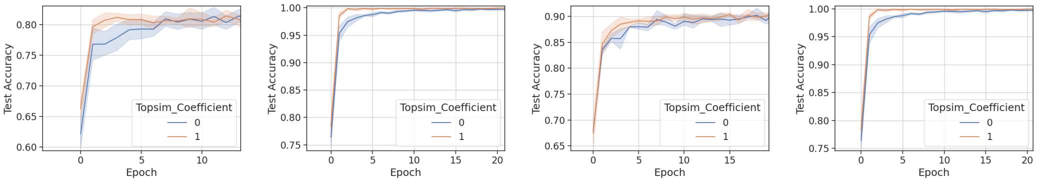
Despite the improvement, we do find that environment parameters like agent architecture, input space, and message structure often dominating over other factors. The number of available data points, subject to whether the message space is large enough, increases the test accuracy. Message spaces with identical capacities can perform differently due to differences in their vocabulary size and message length. We experiment on two Listener types and two input spaces, totalling four main experimental configurations (see Figures 2 and 3). If the message space capacity is the same, longer messages and smaller vocabularies perform better than the converse. Above effects are usually seen across all experiment configurations except at some instances with recurrent listener and $X(50,2)$ input (see Appendix B).

Observations, although not universally, suggest that stronger pressures are favored when agents have smaller vocabularies and longer dialogue lengths. This behavior is intuitively consistent with the metric used in the experiments. Symbols in a vocabulary do not have any inherent ordering and should be considered values on a nominal scale. Hence, the edit distance between symbols a and b $d(a,b) = 1$ if $a \neq b$, else $d(a,b) = 0$. The same holds for the values and attributes in the inputs, where values for a given attribute are in the nominal scale. Hence, from a linguistic perspective, the upper bound of maximum distance between input i and j, $d_M(i,j) = |\mathbb{A}|$ and $d_M(i^m, j^m) = L$, where i^m and j^m are corresponding points in the message space.

When $|\mathbb{W}| \gg |\mathbb{V}|$, agents are free to map more than one attribute to a single position in a message. If we have two points i and j in the input space, with larger vocabularies agents can invent a mapping such that, $d(i,j) > d(i^m, j^m)$. If Listener can successfully map messages back into the input space, the latter phenomena will not always hinder performance. The higher degree of freedom allows agents to invent complex mappings that are not possible in constrained message spaces. High compositional pressure will force agents to avoid this type of symbol usage, which is orthogonal to increasing the test accuracy at configurations where $|\mathbb{W}| \gg |\mathbb{V}|$. Smaller vocabularies cause agents to map inputs to messages with appropriate edit distances along a considerable message length. Hence, high compositional pressures are preferred with comparatively smaller vocabularies with longer dialogue lengths, which supports the trend in experiments.

Table 3: Agent performance with different percentages of training data. Linear Listener, Input : $X(50, 2)$

Train data (%)	$C_t = 0$	$C_t = 0.1$
72.7	0.976	1.00
81.8	0.972	0.996
90.9	0.993	0.993



(a) $M(2, 14)$; 0.83-0.85 (b) $M(2, 17)$; 0.99 (c) $M(2, 14)$; 0.91 (d) $M(10, 6)$; 0.99

Figure 4: Transmission speed of new listeners. For a given test accuracy, transmission speed is higher for languages trained with higher topological similarity constant.

4.2.3 Transmission Speed

We further investigate whether there exists any advantage for language transmission if emerging languages are compositional. From our earlier experiments, we observe that agents converge faster in configurations that give higher test accuracy in almost all instances. It implies that a suitable amount of compositional pressure can make agents capture the protocols faster while simultaneously giving higher generalizations.

Then, we select two sets of instances where agents converged to very close test accuracy but under significantly different external pressures. For this, we select $M(2, 14)$, $M(2, 17)$ and $M(10, 6)$ message structures under $X(10, 4)$ input space. Under these configurations $C_t = 0$ and $C_t = 1$ both have reached nearly the same test accuracy (see Appendix B). We first train each agent using the same seeds as the previous experiments, with and without compositional pressures. Next, we conduct experiments by initializing and then training new Listeners for each trained Speaker. For each selected instance, we train three new listeners from scratch while keeping the sender frozen. Then we select runs that yield closer levels of test accuracy and obtain the average per case.

The results indicate that if test accuracy is the same, the language that emerged with higher compositional pressures is easier to acquire by new listeners (see Figure 4). In earlier experiments, we observed configurations, when trained with high compositional pressures, displaying a lower test accuracy than the baseline cases. However, if the difference in test accuracy compared with the baseline is low, configurations trained with high external pressures should yield communication protocols that are easier to capture by new learners.

4.3 Connection between Compositionality Pressure and Generalization

According to our observations, there is no mandate for high compositional pressures always to yield better generalizations, as evident by cases where agents give the highest test accuracy when trained with low pressures. If intuitively argued, such behavior occurs when agents stop optimizing the cross-entropy loss of the primary reconstruction objective. Other popular regularizing techniques such as weight decay (Krogh & Hertz, 1992) and dropout (Srivastava et al., 2014) have similar functionalities. If the decay coefficient or the optimal probability of retention is too large, the training does not converge. Figures 2 and 3 indicates that models with large C_t may not always generalize above others. Unnecessarily pressuring agents towards compositionality can degrade the performance. For an overall evaluation, we select the experimental runs that converged to a training accuracy above 0.95. Then we extract the maximum test accuracy and plot it against test topological similarity (see Figure 5). There are two clusters of points. The cluster to the high end of the horizontal axis represents experiments conducted with $C_t = 0.1, 0.5, 1$. The other cluster represent

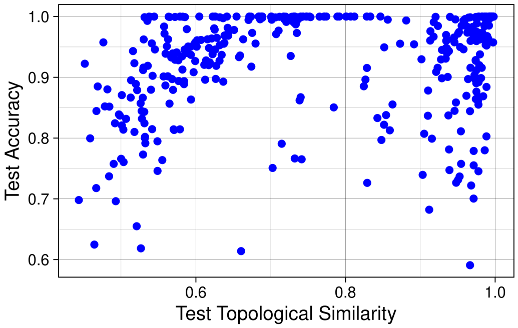


Figure 5: Variation of test accuracy against test topological similarity. We plot each experiment's topological similarity and accuracy that converged to a training accuracy greater than 0.95.

$C_t = 0, 10^{-6}$. There is a considerable amount of instances that achieve near-perfect accuracy and have moderate test topological similarity.

Overall distribution stresses that there is no mandate of high compositionality to achieve good generalizations. If the message space is large and optimal, with high amounts of training data, agents could reach good levels of test accuracy. To further investigate this, we select the $X(50, 2)$ data set and conduct the experiment with varying proportions of training data. We use C_t value of 0.1, which gave the best results in earlier experiments. Both methods have increased their test accuracy up to almost 100% (see Table 3), validating our belief that large training data can mask the advantage offered by compositional pressure.

We propose several reasons for not observing significant improvements at some configurations. First, there are instances where the inadequate message space capacity prevents agents from reaching adequate convergence. We believe that such behavior occurs due to optimization difficulties across the discrete communication channel. Previous studies have similar observations (Resnick et al., 2020; Mordatch & Abbeel, 2018) and we think other methods concerning neural network training should address this. Secondly, if the message space is very much larger than the input space, as previously explained, models can find non-compositional but generalizing representations easily due to an increased degree of freedom. In such cases, since baseline models can also reach near-perfect test accuracy levels, the advantage of compositional pressure is not visible. Nevertheless, in such scenarios, languages that emerged under high compositional pressure are more transmissible.

5 Conclusions

In this paper, we investigate the effect of external pressure for compositionality on language emergence. Our environment models a reconstruction game between two neural agents using value attribute data as the input. Compositional pressure is introduced to the loss function of the neural agents in the form of topological similarity. The relationship between compositionality and generalization is not straightforward because a situational correlation can be observed, even without external pressure. The strength of this relationship, which is determined by factors like agent architecture and training accuracy, is too strong to be discarded. In the case of externally induced compositionality, generalization gets improved for almost all configurations unless hindered by the message space. As a general rule, higher compositional pressures are suitable for situations with smaller vocabularies and longer dialogue lengths. Implying that, for longer-length communications, compositionality should be enforced. Furthermore, compositional languages are significantly easier to acquire by new learners, for a given level of generalization. The effect of compositionality is more important when training data is scarce, where it induce order in representations, improving generalization.

References

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing, pp. 981–985, 2018.

Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242, 2006.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 6293–6303, 2019.

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

Edward Choi, Angeliki Lazaridou, and Nando De Freitas. Compositional obverter communication learning from raw visual input. In *6th International Conference on Learning Representations*, 2018.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *6th International Conference on Learning Representations*, 2018.

Abhinav Gupta, Cinjon Resnick, Jakob Foerster, Andrew Dai, and Kyunghyun Cho. Compositionality and capacity in emergent languages. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 34–38, 2020.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in neural information processing systems*, pp. 2149–2159, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations*, 2017.

Emilio Jorge, Mikael Kågebäck, Fredrik D Johansson, and Emil Gustavsson. Learning to play guess who? and inventing a grounded language as a consequence. *arXiv preprint arXiv:1611.03218*, 2016.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *International Conference on Machine Learning*, pp. 5220–5230. PMLR, 2020.

Simon Kirby and James R Hurford. The emergence of linguistic structure: An overview of the iterated learning model. *Simulating the evolution of language*, pp. 121–147, 2002.

Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014.

Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, 2015.

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pp. 950–957, 1992.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *5th International Conference on Learning Representations*, 2017.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *6th International Conference on Learning Representations*, 2018.

David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020.

Cinjon Resnick, Abhinav Gupta, Jakob Foerster, Andrew M Dai, and Kyunghyun Cho. Capacity, bandwidth, and compositionality in emergent language learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1125–1133, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pp. 2244–2252, 2016.

George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

6 Reproducibility

This paper contains all the hyper parameters and random seeds used in the experiments. The algorithm used, is also presented within the paper itself (Please refer Section 4 and 2).

A Appendix A

Algorithm 1 Reconstruction Game

```
Require: $x$ : Input; $n$ : Batch-Size;
Ensure: $I_0 \leftarrow \mathrm{Embedding}(0)$
Ensure: $i \leftarrow 0$
Ensure: $m \leftarrow [\,]$
Ensure: $m^{smooth} \leftarrow [\,]$
Ensure: $(h_0^{listener}, c_0^{listener}) \leftarrow 0$

  $(\boldsymbol{A}, \boldsymbol{B}) \leftarrow Pair(n)$ ▷ Generate matrices to permute inputs and messages into pairwise formation
  $x \leftarrow \mathrm{Linear}(x)$
  $x \leftarrow \mathrm{BatchNorm}(x)$
  $(h_0, c_0) \leftarrow (x)$                ▷ Obtain the initial hidden and cell states for the LSTM cells

  for $i < L$ do                                                  ▷ Produce a single message
      $(h_i, c_i) \leftarrow \mathrm{LSTMCell}\,[I_{i-1}, (h_{i-1}, c_{i-1})]$
      $logits \leftarrow \mathrm{Linear}(h_i)$
      $w_i^{smooth} \leftarrow GumbelSoftmax(logits)$
      $w_i^{discreet} \leftarrow \mathrm{argmax}(w_i^{smooth})$
      $w_i \leftarrow w_i^{smooth} + \left(\mathrm{onehot}(w_i^{discreet}) - w_i^{smooth}\right) .\mathrm{detach}()$
      $(h_{i-1}, c_{i-1}) \leftarrow (h_i, c_i)$
      $I_{i-1} \leftarrow \mathrm{Embedding}(w_i^{discreet})$
      $m \leftarrow \mathrm{concatenate}(m, w_i)$
      $m^{smooth} \leftarrow \mathrm{concatenate}(m^{smooth}, w_i^{smooth})$
  end for

  $\mathcal{L}_c \leftarrow \rho_{h_0,m}\left[(1 - \cos(\boldsymbol{A}h_0, \boldsymbol{B}h_0)), (1 - \cos(\boldsymbol{A}m, \boldsymbol{B}m))\right]$

  if Listener is Recurrent then
      $h^{listener}, c^{listener} \leftarrow \mathrm{LSTM}[m, (h_{i-1}^{listener}, c_{i-1}^{listener})]$
      $\hat{x} \leftarrow \mathrm{Linear}(h^{listener})$
  else if Listener is Linear then
      $\hat{x} \leftarrow \mathrm{Linear}\,(\mathrm{flatten}(m))$
  end if

  $\mathcal{L}_r \leftarrow \mathcal{L}_{BCE}(x, \hat{x})$
  $\mathcal{L} \leftarrow \mathcal{L}_r + C_t\mathcal{L}_c$
```

B Appendix B

Table 4: Test Accuracy for all configurations

Linear Listener : $X(10,4)$					
$M(W,L)$	0	$1e-06$	0.1	0.5	1
$(10,4)$	0.759 ± 0.042	0.763 ± 0.096	$\mathbf{0.794 \pm 0.097}$	0.757 ± 0.033	0.751 ± 0.125
$(10,5)$	0.990 ± 0.040	0.989 ± 0.035	$\mathbf{0.991 \pm 0.017}$	0.968 ± 0.118	0.899 ± 0.040
$(10,6)$	0.997 ± 0.004	0.999 ± 0.002	$\mathbf{1.000 \pm 0.001}$	0.999 ± 0.001	0.996 ± 0.006
$(100,2)$	$\mathbf{0.588 \pm 0.009}$	0.582 ± 0.001	0.547 ± 0.027	0.495 ± 0.105	0.441 ± 0.034
$(100,4)$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.995 ± 0.019	0.875 ± 0.279	0.744 ± 0.092
$(2,14)$	0.900 ± 0.034	0.902 ± 0.014	0.912 ± 0.018	$\mathbf{0.921 \pm 0.041}$	0.906 ± 0.019
$(2,15)$	0.926 ± 0.178			$\mathbf{0.963 \pm 0.024}$	
$(2,17)$	0.999 ± 0.001	0.936 ± 0.275	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$
Linear Listener : $X(50,2)$					
$M(W,L)$	0	$1e-06$	0.1	0.5	1
$(10,4)$	0.882 ± 0.179	0.841 ± 0.224	$\mathbf{0.948 \pm 0.122}$	0.846 ± 0.127	0.806 ± 0.264
$(10,6)$	0.892 ± 0.139	0.901 ± 0.126	0.917 ± 0.080	$\mathbf{0.980 \pm 0.036}$	0.974 ± 0.046
$(100,2)$	0.440 ± 0.052	$\mathbf{0.460 \pm 0.024}$	0.390 ± 0.063	0.342 ± 0.055	0.324 ± 0.037
$(100,4)$	0.951 ± 0.032	0.963 ± 0.015	$\mathbf{0.964 \pm 0.074}$	0.911 ± 0.129	0.910 ± 0.122
$(50,2)$	$\mathbf{0.430 \pm 0.041}$	0.427 ± 0.032	0.367 ± 0.038	0.301 ± 0.055	0.281 ± 0.079
$(50,4)$	$\mathbf{0.980 \pm 0.040}$	0.963 ± 0.040	0.952 ± 0.069	0.970 ± 0.093	0.905 ± 0.025
Recurrent Listener : $X(10,4)$					
$M(W,L)$	0	$1e-06$	0.1	0.5	1
$(10,4)$	0.711 ± 0.046	0.710 ± 0.033	0.694 ± 0.036	$\mathbf{0.733 \pm 0.161}$	0.549 ± 0.018
$(10,6)$	$0.8460.145$	0.829 ± 0.090	$\mathbf{0.890 \pm 0.178}$	0.811 ± 0.126	0.740 ± 0.173
$(100,2)$	$\mathbf{0.671 \pm 0.030}$	0.662 ± 0.048	0.624 ± 0.053	0.559 ± 0.020	0.526 ± 0.063
$(100,4)$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	0.912 ± 0.081	0.737 ± 0.099
Recurrent Listener : $X(50,2)$					
$M(W,L)$	0	$1e-06$	0.1	0.5	1
$(10,4)$	0.543 ± 0.418	0.472 ± 0.331	0.643 ± 0.451	$\mathbf{0.742 \pm 0.024}$	0.519 ± 0.202
$(10,6)$	0.558 ± 0.345	0.468 ± 0.042	0.477 ± 0.043	$\mathbf{0.615 \pm 0.541}$	0.504 ± 0.381
$(100,2)$	0.916 ± 0.124	$\mathbf{0.934 \pm 0.092}$	0.815 ± 0.099	0.692 ± 0.097	0.618 ± 0.258
$(100,4)$	0.938 ± 0.106	$\mathbf{0.968 \pm 0.027}$	0.953 ± 0.027	0.930 ± 0.051	0.905 ± 0.126
$(50,2)$	0.629 ± 0.040	$\mathbf{0.661 \pm 0.057}$	0.553 ± 0.063	0.530 ± 0.061	0.472 ± 0.053
$(50,4)$	0.918 ± 0.048	$\mathbf{0.946 \pm 0.020}$	0.940 ± 0.055	0.851 ± 0.173	0.773 ± 0.071

C Appendix C

Table 5: Baseline Performance with Dropout for $X(50, 2)$ with recurrent Listener.

band	Test Accuracy	
	$Dropout = 0.5$	$Dropout = 0.7$
$M(10, 4)$	0.478 ∓ 0.040	0.499 ∓ 0.002
$M(10, 6)$	0.455 ∓ 0.108	0.498 ∓ 0.001
$M(100, 2)$	0.811 ∓ 0.365	0.512 ∓ 0.012
$M(100, 4)$	0.879 ∓ 0.013	0.501 ∓ 0.009

Table 6: Baseline Performance with Weight Decay of $1e-6$ for $X(50, 2)$ with recurrent Listener.

$M(W, L)$	Test Accuracy
$M(10, 4)$	0.407 ∓ 0.073
$M(10, 6)$	0.507 ∓ 0.074
$M(100, 2)$	0.941 ∓ 0.124
$M(100, 4)$	0.983 ∓ 0.011

D Appendix D

D.1 Pairing Matrices

Lets assume three inputs $X =< x_1, x_2, x_3 >^T$, and their corresponding messages $M =< m_1, m_2, m_3 >^T$, in a situation where the batch size is equal to 3. Then we define the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\boldsymbol{B} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Which yields pairing vectors as follows,

$$\boldsymbol{A}X = X_A =< x_1, x_1, x_2 >^T$$
$$\boldsymbol{B}X = X_B =< x_2, x_3, x_3 >^T$$
$$\boldsymbol{A}M = M_A =< m_1, m_1, m_2 >^T$$
$$\boldsymbol{B}M = M_B =< m_2, m_3, m_3 >^T$$

Using the vectors X_A, X_B, the distance between each input pair can be calculated,

$$d_x = X_A - X_B$$

Similarly, the corresponding distances for messages can also be obtained to calculate the topological similarity.

E Appendix

E.1 Hyperparameters

We conduct our experiments in ablation format to test the advantage of compositional pressure against the baseline performance. Unless stated, each experiment was repeated six times with different random seeds $(2, 3, 5, 7, 11, 13)$. Input space, defined by the value and attribute parameters, is partitioned into train and test subspaces. During the training and testing phases, samples are drawn randomly from the corresponding partitions. We train agents using mini-batch training, with a batch size of 64. During each epoch, agents are exposed to examples up to five times the size of the input space. We use a learning rate of 0.01 and conduct training for 100 epochs. We use approximately 65% of data for training and the rest for testing.

We conducted our experiments with both recurrent and linear architectures for the Listener, while the Sender was always kept as a recurrent model. We use value attribute environments of $X(50, 2)$ and $X(100, 4)$, having 2500 and $10,000$ input instances respectively. For each configuration, we vary the message space structure and constant C_t, where $C_t = 0$ represents the baseline scenario. We do experiments for $C_t = 0, 1e-6, 0.1, 0.5, 1$. Some of the seeds did not train properly at higher learning rates, especially for the baseline experiments. Hence, for all experiments, we obtain the average of the top three best performing runs.