Agents can generalize to novel levels of abstraction with the help of language

Anonymous ACL submission

Abstract

We study abstraction in an emergent communication paradigm. In emergent communication, two artificial neural network agents develop a language while solving a communicative task. In this study, the agents play a concept-level reference game. This means that the speaker agent has to describe a concept to a listener agent, who has to pick the correct target objects that satisfy the concept. Concepts consist of multiple objects and can be either more specific, i.e. the target objects share many attributes, or more generic, i.e. the target objects share fewer attributes. We test two directions of zero-shot generalization to novel levels of abstraction: When generalizing from more generic to very specific concepts, agents utilize a compositional strategy. When generalizing from more specific to very generic concepts, agents utilize a more flexible linguistic strategy that involves reusing many messages from training. Our results provide evidence that neural network agents can learn robust concepts based on which they can generalize with the help of language. This research can help improve artificial systems and provide new hypotheses on human abstraction.

1 Introduction

One of the most fundamental goals of Artificial Intelligence (AI) and Natural Language Processing (NLP) research is to build models which can generalize well to unseen data. This is, after all, one of the crucial abilities observed in human intelligence. Importantly, there can be no generalization without abstraction, i.e. without first abstracting the knowledge that is needed for making generalizations (Yee, 2019). For example, to form the concept MAMMAL and generalize well to newly observed kinds of mammals, it is not sufficient to store different instances of mammals in our mind. Rather, we abstract knowledge that we have about mammals, and this enables us to classify new instances of this class, i.e. to generalize. This means that generalization and abstraction are intricately linked and that for achieving the goal of well-generalizing models, we need to understand abstraction.

Humans naturally use abstraction to solve complex tasks and to communicate about strategies and solutions. Well-designed AI and NLP systems cannot only benefit from good abstraction abilities in, for example, reasoning and solving complex tasks (e.g. Ho et al., 2019; Zheng et al., 2024), but interactive systems should also be able to deal with human language inputs which can involve abstractions (e.g. Lachmy et al., 2022). Many researchers studying human abstraction argue for a role of language therein (see e.g., Yee, 2019; Sloutsky and Deng, 2019; Gentner and Asmuth, 2019; Lupyan and Lewis, 2019). The main idea of these accounts is that the lexicalization of concepts, i.e. having a label for a concept, helps to acquire and structure information we obtain about an entity and to observe commonalities within members of a concept in the first place. These ideas can also be tested in computational systems. The goals of the current research are to inform the improvement of AI and NLP systems towards achieving human-like abstraction abilities and to use computational modeling to understand abstraction in humans better.

Starting from the assumption that language is useful for abstraction, we study abstraction in a communicative setting and investigate how abstraction is achieved with the help of language. To gain insights into the principled mechanisms of abstraction and the role of language for abstraction, we use a language emergence scenario. In language emergence research, the idea is to define a set of assumptions and then observe how these assumptions change a language-like system that emerges during interaction (see e.g. Lazaridou et al., 2017; Galke et al., 2022; Rodríguez Luna et al., 2020; Chaabouni et al., 2020). In our model, two artificial neural network agents solve a reference game,

where a speaker agent has to communicate a target concept to a listener agent who needs to select the correct target concept in a context. We operationalize a concept as a set of target objects following previous work (Kobrock et al., 2024a,b; Mu and Goodman, 2021). Our target concepts are designed in a hierarchical fashion, ranging from very specific concepts consisting of objects where all attributes are fixed to a certain value, e.g. 'small blue circle', to very generic concepts consisting of objects where only one attribute is fixed, e.g. 'circle'. We can study abstraction by making use of this abstraction hierarchy. Here, we are interested in a specific kind of abstraction, namely the zero-shot generalization to concepts at novel levels of the concept hierarchy, or, to concepts at *novel levels of abstraction* (following the terminology of seminal research from Cognitive Psychology by Rosch et al., 1976). We will not only look at the generalization performance of the trained models, but also at the linguistic strategies the agents employ. Specifically, we are interested in the properties of the emergent lexicon and the use of novel vs. established labels during abstraction.

2 Method

2.1 General Setup

We use an emergent communication paradigm (e.g. Lazaridou et al., 2018; Chaabouni et al., 2019) and build on the concept-level reference game developed in previous work (Mu and Goodman, 2021; Kobrock et al., 2024a). We train two artificial neural network agents, one speaker and one listener agent. Over several iterations, these agents develop a communication system by solving the following task: The speaker agent has to communicate a concept, i.e. a set of target objects, to the listener agent whose task is to identify the correct targets among a set of distractors. In other words, the listener has to identify the target concept in a certain context given a message generated by the speaker. Concepts can range from very specific, where all attributes are shared among the target objects, to very generic, where only one attribute is shared among the targets. Contexts can range from very fine, where all but one attributes are shared between targets and distractors, to very coarse, where no attribute is shared between targets and distractors. Both agent networks are trained in a Reinforcement Learning paradigm with the Gumbel-Softmax relaxation (Jang et al., 2017) on a joint loss that depends on whether the listener correctly identifies the targets and distractors given the speaker-generated message.

2.2 Zero-shot Conditions and Hypotheses

We test the zero-shot generalization abilities of the trained networks in two conditions (see Figure 1): The first condition, "to specific", tests whether agents are able to generalize to the most specific concepts when having seen more generic concepts during training. In this condition, we expect the emerging communication system to encode more generic concepts (such as "blue" or "circle"). For a successful zero-shot generalization, these more generic concepts would need to be combined to describe a specific concept (such as "blue circle"). Here, agents will need to combine previously learned attributes compositionally to describe a more specific concept. The second condition, "to generic", tests whether agents are able to generalize to the most generic concepts when having seen more specific concepts during training. In this condition, we expect the emerging communication system to encode more specific concepts (such as "blue circle" or "orange circle"). For a successful zero-shot generalization, agents will need to abstract away from contextually irrelevant features and find the common attribute that all targets share (e.g. "circle").

2.3 Dataset

The agents are trained on six symbolic datasets developed in previous work (Kobrock et al., 2024a). These datasets contain all possible concepts, ranging from specific to generic, and contexts, ranging from fine to coarse, for a given number of attributes and values. For example, dataset D(3,4) contains all possible concepts and contexts given that objects in this dataset have three attributes and each attribute can take four different values. If we think of the three attributes as shape, color and size, an example for a specific concept would be "small blue circle" and an example for a generic concept would be "square". In a fine context, objects belonging to the concept "small blue circle" would need to be discriminated against objects that are also small and blue. In a coarse context, distractor objects do not share any attributes with the target concept. This also means that there are more possible contexts for specific concepts than for generic concepts and the datasets reflect this relationship. We use a scaling factor of 10 to construct the datasets, i.e. each

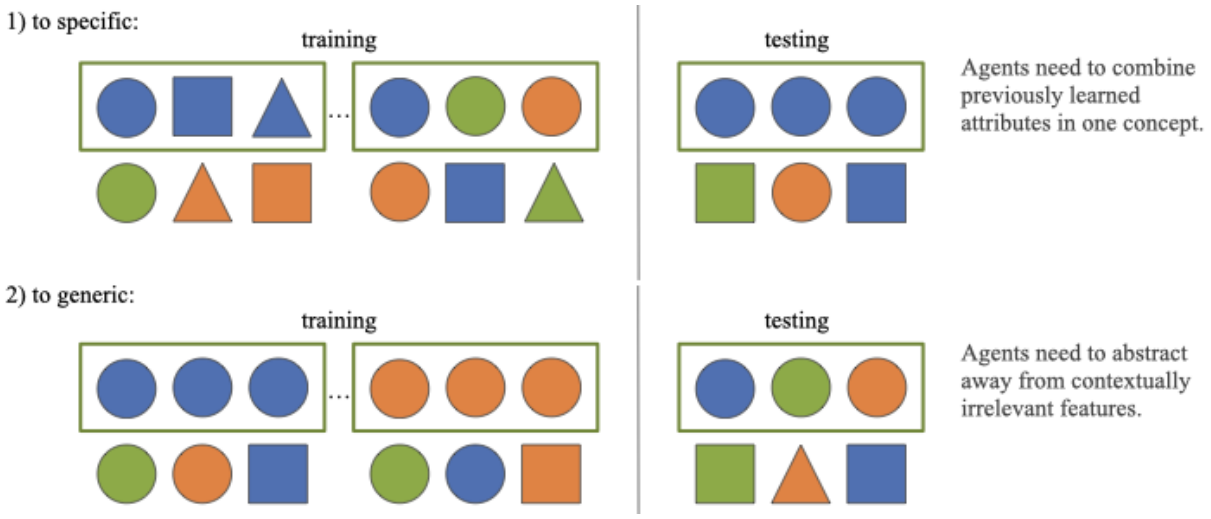

Figure 1: The two zero-shot test conditions: In the "to specific" condition, agents are trained on more generic concepts such as "blue" and "circle" and need to generalize to very specific concepts. In the "to generic" condition, agents are trained on more specific concepts and need to generalize to very generic concepts such as "circle".

concept is included in a dataset 10 times.[1] This ensures that the datasets contain enough training data.

For the zero-shot dataset generation, we manipulate the training, validation and test splits of the data. In the "to specific" condition, the test split contains all most specific concepts available, i.e. those where all attributes are shared among the targets. The training and validation splits are composed of the remaining concepts which are more generic with 75% of the data used for training and 25% of the data used for validation. In the "to generic" condition, the test split contains all most generic concepts available, i.e. those where only one attribute is shared among the targets. The training and validation sets contain the remaining more specific concepts with 75% of the data used for training and 25% of the data used for validation. Dataset sizes can be inspected in the appendix and are comparable between zero-shot conditions.

2.4 Architecture and Training

We use the EGG framework for emergent communication games (Kharitonov et al., 2019, MIT license) in our implementation.[2] A communication game between a speaker S and a listener L is defined as $G = (T^S, D^S, T^L, D^L)$, where $T^S = \{t_1^T, ..., t_g^T\}$ and $D^S = \{d_1^S, ..., d_g^S\}$ are the inputs to the speaker, i.e. sets of game size g targets and distractors, and T^L and D^L are the analogously defined inputs to the listener. For these inputs, $T^S \neq T^L$ and $D^S \neq D^L$ hold, i.e. the targets and distractors presented to the speaker differ from the targets and distractors presented to the listener (Mu and Goodman, 2021; Kobrock et al., 2024a). In each round of the game, S generates a message $m = (s_j)_{j \leq M}$, where s_j is a symbol from vocabulary V and M is the maximal message length[3], based on the inputs T^S and D^S. L in turn, receives m and an input $X^L = \{x_1^L, ..., x_i^L\}$, where $i = 2 \cdot g$ which contains the targets T^L and distractors D^L shuffled. L then predicts a label $y_i^L \in \{0, 1\}$ (0: distractor, 1: target) for each object x_i^L in its input. Both agents are implemented in a similar fashion following previous work (Mu and Goodman, 2021; Kobrock et al., 2024a; Ohmer et al., 2022). Dense layers with 64 units serve as embedding layers for the input objects. The speaker targets and distractors are embedded separately and then concatenated into a joint embedding. The listener input objects are processed by just one embedding layer. For message encoding and decoding, both speaker and listener networks use single-layer Gated Recurrent Units (GRUs) (Cho et al., 2014) with a hidden layer size of 128 that can deal with sequential inputs of varying lengths. A speaker-listener pair is trained with binary cross entropy

[1] We use this scaling factor only to construct the train and validation dataset splits. The zero-shot test is performed on a test split that contains the novel concepts only once.

[2] All code and analysis scripts are available at https://anonymous.4open.science/r/zero-shot-abstraction-5EBD/.

[3] The end-of-sequence symbol 0 can be used to terminate a message before M is reached.

loss

$$\mathcal{L}_{BCE}(S, L, G) = -\sum_i \log p^L(y_i^L | x_i^L, \hat{m}), \quad (1)$$

where $\hat{m} \sim p^S(m|T^S, D^S)$ and $p^L(y_i^L|x_i^L, \hat{m}) = \sigma(\text{GRU}^L(\hat{m}) \cdot \text{embed}(x_i^L))$ maximizing the probability that the listener correctly identifies targets and distractors with a label $y_i \in \{0, 1\}$ (0: distractor, 1: target) for each object x_i. To ensure differentiability for backpropagation, we use the straight-through Gumbel-Softmax trick (Jang et al., 2017) with temperature $\tau = 2$ and a decay rate of 0.99. These and other hyperparameters were determined in a grid search that we conducted for all parameters over the different dataset sizes aiming for maximal validation accuracy. We train with batch size 32 and learning rate 0.001. For our simulations, we use game size 10, i.e. 10 target objects form a concept and 10 distractor objects form the context. The maximum message length M is defined as the total number of attributes in a dataset plus the End of Sequence (EOS) symbol 0. The vocabulary size for each dataset corresponds to the total number of attribute values present. We establish a minimal vocabulary size for each dataset as the sum of the number of attribute values plus one additional symbol. This minimal vocabulary size is then scaled by a factor of $f = 3$, as suggested by previous research (Ohmer et al., 2022).

3 Results

We trained the models on six symbolic datasets with varying numbers of attributes and values. In a dataset $D(n, k)$, objects have n attributes which each can take k different values. For all metrics, we report means and standard deviations over five individual runs per dataset.

3.1 Generalization Performance

We evaluate the agents' performance on the test datasets to assess their zero-shot generalization abilities. Accuracies are calculated as a percentage over the objects that the listener classifies as targets or distractors. An accuracy of 0.9 means that 90% of the objects, i.e. 18 objects with a game size of 10, have been classified correctly as targets or distractors. Or, in other words, two objects have been misclassified.

First, we look at the training and validation accuracies for both conditions to assess whether the agents succeeded in the task and learned the relevant concepts. Training and validation accuracies for both conditions are >=0.97 indicating that indeed the agents have learned the task and achieved high performance on both the training and the validation data splits (see Tables 1 and 2).

	training	validation
D(3,4)	0.99 ± 0.00	0.98 ± 0.00
D(3,8)	0.98 ± 0.01	0.98 ± 0.01
D(3,16)	0.97 ± 0.01	0.97 ± 0.01
D(4,4)	1.00 ± 0.00	0.99 ± 0.00
D(4,8)	0.98 ± 0.00	0.98 ± 0.01
D(5,4)	0.99 ± 0.01	0.98 ± 0.01

Table 1: **To specific:** Training and validation accuracies.

	training	validation
D(3,4)	0.99 ± 0.00	0.98 ± 0.00
D(3,8)	0.98 ± 0.01	0.98 ± 0.01
D(3,16)	0.97 ± 0.01	0.97 ± 0.01
D(4,4)	1.00 ± 0.00	0.99 ± 0.00
D(4,8)	0.98 ± 0.00	0.98 ± 0.01
D(5,4)	0.99 ± 0.01	0.98 ± 0.01

Table 2: **To generic:** Training and validation accuracies.

Second, we look at the zero-shot accuracies, i.e. the performance achieved on the test data splits. Table 3 summarizes the mean test accuracies over the five runs conducted on each dataset for both conditions. All zero-shot test accuracies are >=0.63 indicating that the listeners correctly identify more than 60% of the 20 objects as targets or distractors. This corresponds to a number of 12 correctly identified objects. Agents achieve higher performance in the "to specific" condition compared to the "to generic" condition in all datasets. Comparing test accuracies between datasets, generalization performance is better for datasets with more attributes. Specifically, on datasets with at least four attributes, agents achieve generalization accuracies of 0.82 or higher in both conditions. This means that speakers choose expressions to describe the held-out concepts at novel levels of abstraction that enable listeners to classify at least 16 of the 20 objects correctly.

3.2 Concept lexicalization

We assess whether agents lexicalized the concepts during training, i.e. the more generic concepts in the "to specific" condition, and the more specific concepts in the "to generic" condition. To achieve

	to specific	to generic
D(3,4)	0.92 ± 0.02	0.71 ± 0.04
D(3,8)	0.85 ± 0.01	0.68 ± 0.07
D(3,16)	0.82 ± 0.03	0.63 ± 0.03
D(4,4)	0.95 ± 0.00	0.82 ± 0.02
D(4,8)	0.95 ± 0.01	0.82 ± 0.07
D(5,4)	0.96 ± 0.01	0.84 ± 0.06

Table 3: Zero-shot test accuracies for both conditions.

	to specific	to generic
D(3,4)	0.93 ± 0.03	0.87 ± 0.04
D(3,8)	0.95 ± 0.01	0.82 ± 0.02
D(3,16)	0.87 ± 0.01	0.77 ± 0.02
D(4,4)	0.94 ± 0.01	0.87 ± 0.05
D(4,8)	0.84 ± 0.03	0.83 ± 0.03
D(5,4)	0.87 ± 0.02	0.83 ± 0.04

Table 4: NMI scores for both conditions.

this, we calculate the Normalized Mutual Information (NMI) score over messages and concepts after training. It is calculated as follows:

$$\text{NMI}(C, M) = \frac{H(M) - H(M|C)}{0.5 \cdot (H(C) + H(M))}, \quad (2)$$

with C being the set of concepts and M being the set of messages. The NMI score is maximal (i.e., 1.0) if for all messages and concepts seen during training, every message maps to exactly one concept and vice versa. In other words, a maximal score indicates that the agents developed a lexicon that includes only one-to-one mappings between words and concepts, i.e. no ambiguity. We expect high but not maximal NMI scores which would indicate that the agents have learned a structured but not unambiguous lexicon. The mean NMI scores calculated on messages and concepts during training over five runs range between 0.84 and 0.95 in the "to specific" condition, i.e. when trained on more generic concepts and between 0.77 and 0.87 in the "to generic" condition, i.e. when trained on more specific concepts (see Table 4). This indicates that a structured lexicon has emerged in both conditions, while more lexical ambiguity arises when training the agents on more specific concepts in the "to generic" condition.

3.3 Generalization strategies

If agents compositionally combine symbols to refer to the specific concepts in the "to specific" condition, we expect symbols but not entire messages to be reused during testing. In other words, in the "to specific" condition, we expect many novel messages to refer to novel concepts and fewer reused messages that have already been produced during training. This would indicate that the agents make use of a compositional strategy when referring to novel concepts. An alternative strategy would be that the agents mainly reuse messages that have been uttered during training and do not produce many novel messages. This strategy leads to the production of underinformative messages in certain contexts, i.e. messages that do not provide enough information for the listener to unambiguously identify the target concept (e.g. Engelhardt et al., 2006; Deutsch and Pechmann, 1982; Grice, 1975). For example, a message that has been used to refer to the concept "blue circle" during training might be used to refer to a "small blue circle" during testing. In some contexts, e.g. when all blue circles are small blue circles, this is efficient. In other contexts, however, e.g. when small blue circles need to be discriminated from large blue circles, this strategy is underinformative and not effective.

In the "to generic" condition, we can also think of two different strategies the agents might use: Either, the agents reuse messages from training or they produce novel messages to refer to novel concepts. If the speaker agents produce novel messages to refer to novel concepts, they might come up with a highly efficient mapping, but they also run into the risk that the listener might not work out what the novel word refers to. This is due to the fact that the agents cannot draw on a compositional strategy when communicating only a single relevant attribute. However, if the speaker agents reuse messages from training, this will result in overinformative messages, i.e. messages that provide more information than necessarily required for the listener to unambiguously identify the target concept (e.g. Grice, 1975; Degen et al., 2020; Rubio-Fernandez, 2021), in certain contexts, e.g. when producing "small blue circle" in reference to a CIRCLE that needs to be discriminated against other shapes. In other contexts, however, reused messages are highly efficient: As concepts are presented in a variety of contexts during training, there are communicative situations (namely coarse contexts) in which the speaker agent can choose to communicate only a single relevant attribute and rely on context to resolve ambiguity. This might lead to the lexicalization of generic concepts such

as CIRCLE already during training, even though they are never explicitly presented. Reusing these messages during testing will thus be highly efficient. This is why we expect the agents to use more reused than novel messages in the "to generic" condition.

To test these predictions, we look at the messages generated during testing on the novel concepts. First, we define the set of test concepts C_{test} and the set of test messages M_{test}. These are the messages produced and the concepts described during interactions on the test data split. Next, we define a message-concept ratio as the ratio between the number of test messages and the number of test concepts M_{test}/C_{test}. The resulting ratio is 1.0 if the number of messages is equal to the number of novel concepts, or, in other words, if for each novel concept, the agents produce one message during testing. Scores lower than 1.0 indicate that the agents produce fewer distinct messages than there are novel concepts. We calculate the ratios to ensure comparability between the "to specific" and "to generic" conditions because the test sets contain different amounts of novel concepts.

We also look at the overlap between messages used during training and validation $M_{trainval}$ and messages used when generalizing to the test data split M_{test}. Both sets only contain unique message counts from training, validation and testing. We define novel messages as those messages that have been produced during testing but have not been produced during training and validation, i.e. the set difference $M_{test} - M_{trainval}$. We calculate a novelty rate as the ratio between the number of novel messages and the total number of unique messages used during testing $|M_{test} - M_{trainval}|/|M_{test}|$. We define reused messages as those messages that have been used in training and validation and then reused in testing, i.e. the intersection of the two sets of messages $M_{trainval} \cap M_{test}$. We calculate a reuse rate as the ratio between the number of reused messages and the number of total unique messages used during testing $|M_{trainval} \cap M_{test}|/|M_{test}|$. If reuse rate and novelty rate are balanced, this means that agents invent equally many new messages as they reuse old messages from the training. If the percentages shift to one or the other extreme, this means that agents reuse more old messages than they invent new ones or vice versa.

In the "to specific" condition, we find that ratios between distinct messages and novel concepts in the test set range between 0.23 and 0.92 (see Table 5). The dataset with the highest ratio close to 1.0 is D(4,4) with a score of 0.92, where almost for each novel concept, a distinct message is produced. Strikingly, there are many datasets with a low message-concept ratio, which suggests that one message is used to refer to many concepts. As test accuracies are generally high (see Table 3), this probably reflects the emergence of a very efficient language, where many concepts can be described with a small lexicon. As accuracies are not maximal, though, this efficient strategy might come at the loss of some objects being misclassified by the listener. Message-concept ratios are higher for datasets with fewer values. One reason for this might be that the emerging language is more structured for these datasets. A more structured language allows the agents to use previously established meanings and combine them in a compositional fashion to novel meanings. As the novel meanings are generated on the fly, the agents might vary which previously established meanings they use, how they combine them and in which order, leading to a large amount of novel messages. Indeed, when looking at the reuse and novelty rates, we find that agents trained in the "to specific" condition, invent at least 34% new messages during testing (see Table 5). For half of the datasets, the novelty rate even exceeds the reuse rate, suggesting that the agents combine symbols more often to new messages rather than reuse old messages.

In the "to generic" condition, we observe message-concept ratios of distinct test messages to novel concepts that are very close to 1.0, indicating that the agents produce almost exactly one message for each novel concept (see Table 6).We find that agents trained in the "to generic" condition reuse more messages than invent new ones (see Table 6). When testing on the dataset D(5,4), agents even only reuse messages and invent no novel messages. If these messages have encoded all relevant attributes of a target concept during training, then these messages are necessarily overinformative when produced during testing, but might still lead to high communicative success. Another possibility is that some of these messages have not encoded all relevant attributes during training, but that they were underinformative during training and required the context to resolve ambiguity. These messages might be just on the appropriate level of reference during testing. This might explain the highly efficient reuse of messages in the "to generic" condition.

	test concepts	test messages	message-concept ratio	reuse rate	novelty rate
D(3,4)	64	54.0 ± 3.6	0.84 ± 0.06	0.54 ± 0.07	0.46 ± 0.07
D(3,8)	512	348.8 ± 28.5	0.68 ± 0.06	0.45 ± 0.07	0.55 ± 0.07
D(3,16)	4096	948.6 ± 123.7	0.23 ± 0.03	0.45 ± 0.07	0.55 ± 0.07
D(4,4)	256	236.2 ± 9.6	0.92 ± 0.04	0.44 ± 0.08	0.56 ± 0.08
D(4,8)	4096	2380.8 ± 437.7	0.58 ± 0.11	0.66 ± 0.12	0.34 ± 0.12
D(5,4)	1024	911.0 ± 34.9	0.89 ± 0.03	0.65 ± 0.09	0.35 ± 0.09

Table 5: **To specific:** Number of concepts and messages for the test split and their ratio. Percentage of reused and novel messages from the total set of unique novel messages.

	test concepts	test messages	message-concept ratio	reuse rate	novelty rate
D(3,4)	12	11.8 ± 0.4	0.98 ± 0.03	0.80 ± 0.19	0.20 ± 0.19
D(3,8)	24	22.2 ± 0.4	0.93 ± 0.02	0.83 ± 0.07	0.17 ± 0.07
D(3,16)	48	44.4 ± 0.8	0.93 ± 0.02	0.75 ± 0.06	0.25 ± 0.06
D(4,4)	16	15.6 ± 0.8	0.97 ± 0.05	0.74 ± 0.14	0.26 ± 0.14
D(4,8)	32	28.8 ± 3.1	0.90 ± 0.10	0.93 ± 0.06	0.07 ± 0.06
D(5,4)	20	19.6 ± 0.5	0.98 ± 0.02	1.00 ± 0.00	0.00 ± 0.00

Table 6: **To generic:** Number of concepts and messages for the test split and their ratio. Percentage of reused and novel messages from the total set of unique novel messages.

3.4 Compositionality

In the previous section, we hypothesized that agents in the "to specific" condition use a compositional strategy, whereas agents trained in the "to generic" condition do not seem to rely on composition but rather reuse messages from training to describe novel concepts. To test the compositionality in the emerging languages, we use the topographic similarity metric which has been introduced and successfully used in previous work (e.g., Ohmer et al., 2022; Lazaridou et al., 2018; Brighton and Kirby, 2006). We calculate topographic similarity between messages and concepts by first calculating two distance vectors: one containing the pairwise cosine distances between concepts and one containing the pairwise cosine distances between messages. Then we correlate these two distance vectors by using Spearman correlation to obtain the topographic similarity score between 0 and 1.0. The higher the score, the more compositional are the messages.

We find that the mean compositionality scores over five runs for concepts and messages seen during training range between 0.06 and 0.49 for the "to specific" condition, and that they range between 0.19 and 0.44 for the "to generic" condition. Compositionality scores are higher for datasets with fewer values in both conditions. This is in line with our finding from above, when we observed that the message-concept ratio is higher for datasets with fewer values in the "to specific" condition and hypothesized that this is due to a more structured language emerging for datasets with fewer values. When looking at the topographic similarity scores calculated on the sets of messages and concepts from the test data split, we see a diverging picture: For the "to specific" condition, compositionality scores are higher during testing, suggesting that the agents use a highly compositional strategy when describing very specific concepts. For the "to generic" condition, on the other hand, we observe a drop of compositionality scores almost towards zero. This means that agents do not use a compositional strategy when being tested on the most

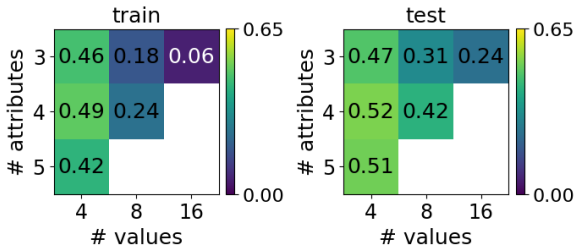

Figure 2: **To specific:** Topographic similarity scores calculated on messages from the train and test splits.

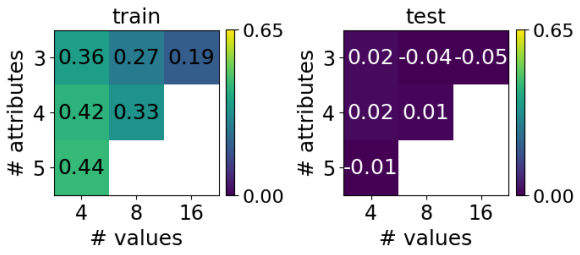

Figure 3: **To generic:** Topographic similarity scores calculated on messages from the train and test splits.

generic concepts. This observation supports the conclusion that agents use a compositional strategy when generalizing to the most specific concepts and that they use a more flexible strategy that is characterized by a large amount of reused messages when generalizing to the most generic concepts.

4 Discussion

In this study, we set out to investigate the linguistic strategies agents use to generalize to concepts at novel levels of abstraction. Our main finding is that the abstraction abilities and linguistic strategies differ depending on the direction of zero-shot generalization: When the agents generalize to very specific concepts, they make use of a compositional strategy. When they generalize to very generic concepts, their strategy is mostly characterized by reusing already established messages.

We observe lower performance when testing zero-shot generalization to very generic concepts than zero-shot generalization to very specific concepts. An intuitive reason for this might be that combining learned symbols for attributes like "circle" or "blue" into a message "blue circle" is easier to achieve with a finite lexicon than abstracting to novel generic concepts. If a language does not encode specific concepts, novel meanings can always be generated by combining established meanings. However, if a language does not encode generic concepts, a compositional strategy is not an option. Novel meanings, however, cannot be established in a zero-shot generalization, so the only chance the agents have is to recur to already established meanings. But previously learned words that encode specific meanings are only useful to a certain extent, making the "to generic" direction of zero-shot generalization a harder task for our trained agent models which is reflected in the accuracies.

This idea is supported by evidence we obtained from an in-depth analysis of the emerging lexica. First, we have shown that the concepts seen during training are lexicalized in both conditions. We have observed that more lexical ambiguity emerges when training the agents in the "to generic" condition, licensed by lower NMI scores on the train dataset in this condition. This lexical ambiguity likely is what enables the agents to perform fairly well on the generalization task, where we observe that they mostly reuse messages from the training phase to refer to novel concepts. Lexical ambiguity and the use of messages that rely on context to resolve ambiguities that remain after interpreting a message can lead to an efficient strategy that agents rely on mainly in coarse context conditions (see Kobrock et al., 2024a). Indeed, when training agents in the "to generic" condition, we train them on the more specific concepts which can, per definition, occur in a more diverse range of contexts, including coarse contexts.

Second, we have shown that agents employ different strategies for generalizing to the most specific or most generic concepts. Specifically, agents come up with more novel messages that have a high degree of compositionality when being tested on the most specific concepts. This is in line with results from previous research on the role of compositionality in generalization (Chaabouni et al., 2020). Agents trained in the "to generic" condition, on the other hand, come up with fewer novel messages. Here, the successful strategy seems to be a high reuse rate of messages the meaning of which has been established during training. The agents trained in the "to generic" condition thus make use of the lexical ambiguity that has emerged during training, accepting that these messages may be overinformative when describing the most generic concepts.

We have shown how the use of labels that have emerged during interaction in a communicative setting can support generalization to novel levels of abstraction together with the ability to compositionally combine meanings. These results can inform the improvement of AI and NLP systems by using the insight that generalization to more specific or to more generic concepts might require different strategies. Our results also add to existing evidence that language in the form of labels are important for abstraction in humans and our results can help to formalize new hypotheses about the role of compositional and novel vs. established labels in abstraction.

Limitations

The experiments presented here have been conducted as a proof-of-concept on symbolic data. A validation on a more natural dataset is planned for future work and is expected to improve the generalizability of these results.

Acknowledgments

References

Henry Brighton and Simon Kirby. 2006. Understanding Linguistic Evolution by Visualizing the Emergence of Topographic Mappings. *Artificial Life*, 12(2):229–242.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and Generalization In Emergent Languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4427–4442.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019. Anti-efficient encoding in emergent communication. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint*.

Judith Degen, Robert D. Hawkins, Caroline Graf, Elisa Kreiss, and Noah D. Goodman. 2020. When redundancy is useful: A Bayesian approach to "overinformative" referring expressions. *Psychological Review*, 127(4):591–621.

Werner Deutsch and Thomas Pechmann. 1982. Social interaction and the development of definite descriptions. *Cognition*, 11(2):159–184.

Paul E. Engelhardt, Karl G. D. Bailey, and Fernanda Ferreira. 2006. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, 54(4):554–573.

Lukas Galke, Yoav Ram, and Limor Raviv. 2022. Emergent Communication for Understanding Human Language Evolution: What's Missing? In *Emergent Communication Workshop at ICLR 2022*.

Dedre Gentner and Jennifer Asmuth. 2019. Metaphoric extension, relational categories, and abstraction. *Language, Cognition and Neuroscience*, 34(10):1298–1307. Publisher: Routledge.

Paul Herbert Grice. 1975. Logic and Conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and semantics*, volume 3, speech acts, pages 41 – 58. NY: Academic Press, New York.

Mark K Ho, David Abel, Thomas L Griffiths, and Michael L Littman. 2019. The value of abstraction. *Current Opinion in Behavioral Sciences*, 29:111–116.

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

Eugene Kharitonov, Rahma Chaabouni, Marco Baroni, and Diane Bouchacourt. 2019. EGG: A toolkit for research on emergence of language in games. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Proceedings of System Demonstrations*, pages 55–60.

Kristina Kobrock, Xenia Isabel Ohmer, Elia Bruni, and Nicole Gotzner. 2024a. Context Shapes Emergent Communication about Concepts at Different Levels of Abstraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3831–3848, Torino, Italia. ELRA and ICCL.

Kristina Kobrock, Charlotte Uhlemann, and Nicole Gotzner. 2024b. Superordinate referring expressions in abstraction: Introducing the concept-level reference game. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Royi Lachmy, Valentina Pyatkin, Avshalom Manevich, and Reut Tsarfaty. 2022. Draw Me a Flower: Processing and Grounding Abstraction in Natural Language. *Transactions of the Association for Computational Linguistics*, 10:1341–1356.

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

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

Gary Lupyan and Molly Lewis. 2019. From words-as-mappings to words-as-cues: the role of language in semantic knowledge. *Language, Cognition and Neuroscience*, 34(10):1319–1337. Publisher: Routledge.

Jesse Mu and Noah Goodman. 2021. Emergent Communication of Generalizations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 17994–18007.

Xenia Ohmer, Marko Duda, and Elia Bruni. 2022. Emergence of Hierarchical Reference Systems in Multi-agent Communication. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5689–5706.

Diana Rodríguez Luna, Edoardo Maria Ponti, Dieuwke Hupkes, and Elia Bruni. 2020. Internal and external pressures on language emergence: least effort, object constancy and frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4428–4437, Online. Association for Computational Linguistics.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. 1976. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Paula Rubio-Fernandez. 2021. Color discriminability makes over-specification efficient: Theoretical analysis and empirical evidence. *Humanities and Social Sciences Communications*, 8(1):147.

Vladimir M. Sloutsky and Wei (Sophia) Deng. 2019. Categories, concepts, and conceptual development. *Language, Cognition and Neuroscience*, 34(10):1284–1297. Publisher: Routledge.

Eiling Yee. 2019. Abstraction and concepts: when, how, where, what and why? *Language, Cognition and Neuroscience*, 34(10):1257–1265.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models.

A Dataset sizes

Tables 7 and 8 show the sizes of the datasets.

	training	validation	test	total
D(3,4)	810	270	64	1144
D(3,8)	3060	992	512	4564
D(3,16)	11880	3936	4096	19912
D(4,4)	7320	2432	256	10008
D(4,8)	52080	17344	4096	73520
D(5,4)	55350	18432	1024	74806

Table 7: **To specific:** Number of concepts in each dataset for each dataset split.

	training	validation	test	total
D(3,4)	780	308	12	1036
D(3,8)	3435	1429	24	4376
D(3,16)	15606	7945	48	19504
D(4,4)	7429	2683	16	9871
D(4,8)	55972	21339	32	73248
D(5,4)	56140	19507	20	74643

Table 8: **To generic:** Number of concepts in each dataset for each dataset split.

B Computational Budget

We ran the experiments reported in this paper on a High-Performance-Computing Cluster (HPC3) on a single gpu core, using up to 400GB memory. We estimate the computing time for reproducing all results reported here, including generating the datasets and training the models on the six datasets for five runs at <72h with comparable resources.