Multi-Agent Language Learning: Symbolic Mapping

Yicheng Feng
School of Computer Science
Peking University
fyc813@pku.ecu.cn

Zongqing Lu[†]
Peking University
BAAI
zongqing.lu@pku.edu.cn

Abstract

The study of emergent communication has long been devoted to coax neural network agents to learn a language sharing similar properties with human language. In this paper, we try to find a 'natural' way to help agents learn a *compositional* and *symmetric* language in complex settings like dialog games. Inspired by the theory that human language was originated from simple interactions, we hypothesize that language may evolve from simple tasks to difficult tasks. We propose a curriculum learning method called *task transfer*, and propose a novel architecture called *symbolic mapping*. We find that task transfer distinctly helps language learning in difficult tasks, and symbolic mapping promotes the effect. Further, we explore *vocabulary expansion*, and show that with the help of symbolic mapping, agents can easily learn to use new symbols when the environment becomes more complex. All in all, we find that a process from simplicity to complexity can serve as a natural way to help multi-agent language learning, and the proposed symbolic mapping is effective for this process.

1 Introduction

Agent communication has been a popular research field in the context of multi-agent reinforcement learning (Foerster et al., 2016; Sukhbaatar et al., 2016; Jiang and Lu, 2018; Eccles et al., 2019). Recent work has focused on the emergence of language in cooperative tasks where neural network agents learn a communication protocol from scratch to solve problems together (Lazaridou et al., 2017; Das et al., 2017; Havrylov and Titov, 2017; Kottur et al., 2017; Li and Bowling, 2019; Ren et al., 2020). An array of work has empirically shown that agents can make use of their developed language to successfully complete the tasks. Beyond that, some work probed into the process of language emergence, and tried to figure out whether the learned language could share similar properties with human language like *compositionality* (Mordatch and Abbeel, 2018; Resnick et al., 2020; Chaabouni et al., 2020; Choi et al., 2018) and *symmetry* (Graesser et al., 2019; Dubova and Moskvichev, 2020; Dubova et al., 2020).

Most of these studies on emergent communication are based on *referential games* (Lewis, 1969) and have shown that compositionality can be induced with suitable environmental pressures. Some have explored the influential factors on the symmetry of protocols among a group of agents. However, tasks in these studies are often simple, and some of these methods are hard to implement in complex settings like dialog games. Kottur et al. (2017) found that in a two-agent multi-round dialog game, language with compositionality does not naturally emerge, unless strict conditions are imposed to agents, such as deprivation of memory.

Language emergence only in simple games is obviously not satisfactory. In this paper, we tend to find a new way to make compositional and symmetric language emerge 'naturally' in complex settings. Psychological studies suggest that human language was originated from simple gestures like pointing and pantomiming (Tomasello, 2010). This may explain why referential games are suitable for emergent language studies: these games are similar to 'pointing' in pragmatic process. However, from another perspective, the theory may also imply that communication protocols like human language cannot be formed *directly* from complex interactions. Instead, a natural process is probably that a language is first formed in simple tasks, and then applied in more complex tasks, meanwhile it evolves to become more complicated and complete, similar to the concept of *curriculum learning* (Bengio et al., 2009). Hence, we propose a method called *task transfer* to implement this process on emergent communication between neural network agents, and explore whether the process could help

[†] Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 7756–7770
July 9-14, 2023 ©2023 Association for Computational Linguistics

language learning in complex settings through empirical experiments. We also design two tasks for the experiment, including a two-player referential game and a multi-round dialog game involving a group of agents.

Straightforward task transfer may not work well, since agents, even if using a same language all the time, can have different speaking policies across tasks. So instead of transferring the policies directly, we tend to make agents learn a common function for communication. We propose a novel architecture called *symbolic mapping*, which maps the input to related symbols, as a basic component of communication system of agent. The intuition is that when presented with the same input, we always associate it with the same pile of words and concepts, and this kind of association is consistent across tasks, so can be transferred. Our experiments show that agents with symbolic mapping perform better in task transfer.

As we explore the learning process of agents from simple tasks to difficult tasks, we are also curious about how the language evolves if old conventions are not enough in new environments. Language learning should not be accomplished overnight. In a more natural scene, agents should first learn a simple language in a initial environment, and after entering a more complicated environment, they will learn something new and the language develops. We conduct the experiment about *vocabulary expansion*, also in a curriculum learning manner. We find that through vocabulary expansion, agents can accomplish tasks in complex environments where they would fail if they are asked to learn a language directly. This result reveals again that a process from simplicity to complexity is crucial for multi-agent language learning. And we also find that symbolic mapping agents perform better in vocabulary expansion.

2 Related work

Cooperative games. Different kinds of cooperative games have been proposed in emergent communication literature. A popular one is referential game (Lewis, 1969), where one agent, often noted as the speaker, has to send a message describing a target (*e.g.*, an image) which it has just observed to the other agent. Then the other agent, often noted as the listener, must select the target from several candidates containing the target and some distractors, after receiving the message (Lazaridou et al., 2017; Havrylov and Titov, 2017). We use a variant of referential game to serve as the simple task in our experiments, similar to the game in Chaabouni et al. (2020) where the listener should reconstruct the target. The difference is that we train the listener model by reinforcement learning, while they use the cross-entropy loss.

Our difficult task is inspired by the *Task & Talk* game proposed by Kottur et al. (2017), which is a multi-round dialog game. In the Task & Talk game, there are two agents, one always asks questions while the other answers these questions. However, our task involves a group of homogeneous agents who do not play specific roles.Besides, our task has unfixed number of rounds, making the game more realistic while more complex. Other studies (Mordatch and Abbeel, 2018; Graesser et al., 2019; Fitzgerald, 2019) also concern emergent language in a group of agents, and Evtimova et al. (2018) proposed a multi-step referential game.

Properties of communication protocols. A mainstream research direction in emergent communication is to find out whether neural network agents can produce communication protocols which exhibit some properties of human language. The most extensively studied property is compositionality. Many studies (Lazaridou et al., 2018; Li and Bowling, 2019; Ren et al., 2020; Resnick et al., 2020) have found that in referential games, once given appropriate environmental pressures, like changing learning environments, communication capacities or agents' model capacities, compositionality could be improved. Kottur et al. (2017) found that compositionality does not emerge naturally in dialog games, which is also verified by our experiments. In the studies where a group of agents learn their languages together, another important property is symmetry. That means an agent community should converge on a shared communication protocol. Dubova et al. (2020) investigated the impact of different social network structures on language symmetry, while Dubova and Moskvichev (2020) explored some other factors including supervision, population size and self-play. In this paper, we focus on improving the two properties through a process from simplicity to complexity.

Evolution of communication. Recent studies, inspired by linguistic theories, have brought evolution into the research of emergent communication. Cogswell et al. (2019) investigated the benefit from cultural transmission, while Dagan et al. (2021)

integrated both cultural evolution and genetic evolution. Ren et al. (2020) proposed a neural iterated learning algorithm, where agents in a new generation are partially exposed to the language emerged from the previous one. Li and Bowling (2019) let the speaker interact with new listeners periodically, while Graesser et al. (2019) analyzed how the language evolves when different linguistic communities come in contact with each other. Most similar to our approach, Korbak et al. (2019) explored language learning across games of varying complexity by template transfer. Different from their work where a hard task is decomposed into several parts and the transferred agent is the listener, we explore language transfer from simple interactions to different tasks involving more complex communication forms, and the speaker is not reinitialized so that the language evolution is consistent. And we also explore the expansion of vocabulary.

Symbolic representation. Previous studies have explored symbolic representation in the deep reinforcement learning (RL) framework (Garnelo et al., 2016; Garnelo and Shanahan, 2019), and found that a compositionally structured representation could help address several shortcomings inherent in the deep RL systems. Symbolic mapping can be seen as a kind of symbolic representation in its function. However, unlike prior work, symbolic mapping is learned and constructed through emergent communication instead of representation learning techniques and is trained end-to-end by RL. That means agents form the symbolic representation when learning to communicate.

3 Method

3.1 Task transfer

Our main hypothesis is that multi agent language learning should benefit from a process from simplicity to complexity, which brings us to curriculum learning. So to prove this, we propose to make agents learn language in a simple task first, and then continue learning in the difficult task, which is a two-stage curriculum. We call this method *task transfer*, since we hope the learned language can be transferred across tasks.

We focus on multi-round dialog games as target tasks in this paper. One question is how to choose the starting point for task transfer. From results in psychological studies, language should first be originated from simple interactions like pointing and pantomiming. Then referential game becomes a reasonable option, since referring to objects is similar to these interactions in pragmatic process. We use description game, a variant of referential game, as the starting point in our experiments.

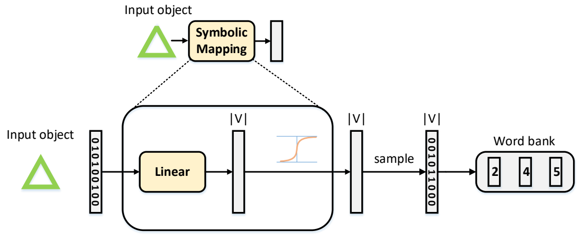


Figure 1: The architecture of symbolic mapping.

3.2 Symbolic Mapping

Curriculum learning is usually helpful in machine learning literature, but language transfer across different tasks is not expected to be a natural outcome. Actually, we think straightforward task transfer may not work that well. Curriculum learning helps policy generalization to similar tasks, but what we explore is language learning across different kinds of games where agents need different policies. So instead of directly transferring the policies, we tend to design a fundamental component of communication system in the architecture which can be shared all the time. Therefore, we propose an architecture called *symbolic mapping*, which maps input to related symbols. Before thinking about which symbols to communicate, we first think about which symbols are relevant, and this kind of association is consistent across tasks.

The illustration of symbolic mapping is shown in Figure 1. Concretely, it is realized by a linear layer followed by a sigmoid function which maps the input object to a vector with dimension $|V|$, which is the vocabulary size, and each element of the vector corresponds to the degree of relevance between a symbol and the object. Several symbols are sampled using the Bernoulli distribution for each element of the vector according to the probability given by the output of the sigmoid function, and then stored as the agent's *word bank*. The number of sampled symbols, namely the size of the word bank, is not predefined or limited, so the mapping process is not restricted but learned with freedom.

Then we propose an architecture that implements symbolic mapping with LSTM based agents so that agents can communicate making use of it. Now that the number of symbols in the word bank is unfixed, we use a speaking network to estimate whether

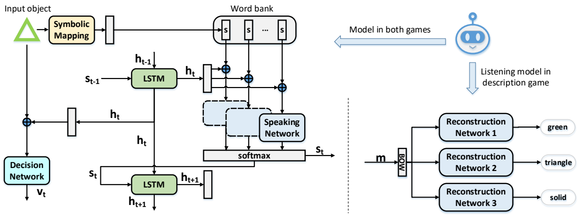


Figure 2: Agent architecture.

each relevant symbol is useful at each time step. The speaking network is realized by a 2-layer MLP, and takes the concatenation of each symbol and the hidden state of LSTM as input, then outputs a score for each symbol. Note that all symbols in the word bank get scores by a shared speaking network. Then we pass all scores through a softmax function to get a probability distribution over the word bank. At training time we sample a symbol from it, while at test time we select symbols using argmax. An illustration can be found in Figure 2 (left).

3.3 Game Settings

Here we describe the two tasks used in our experiments. Discrimination game is the difficult task, a multi-round dialog game, illustrated in Figure 3b. The simple task, description game, is a variant of referential game, as depicted in Figure 3c.

Discrimination game. Discrimination game involves two datasets, object dataset $\mathcal{D}$ and pair dataset $\mathcal{P}$. Each object in $\mathcal{D}$ comprises n attributes. For each attribute $a \in \{1, 2, \ldots, n\}$, there are $m^{(a)}$ possible values. For a given n and a tuple of value numbers $m = (m^{(1)}, m^{(2)}, \ldots, m^{(n)})$, we note the corresponding object dataset as $\mathcal{D}_{n,m}$, and the number of different objects will be $|\mathcal{D}_{n,m}| = \prod_{a=1}^{n} m^{(a)}$. Given an object dataset $\mathcal{D}$, the pair dataset $\mathcal{P}$, as illustrated in Figure 3a, is then constructed as for each pair (o_i, o_j) where $o_i, o_j \in \mathcal{D}$, $o_i = o_j$ or o_i and o_j have only one different attribute. If the objects are selected from $\mathcal{D}_{n,m}$, we note the pair dataset as $\mathcal{P}_{n,m}$. Note that different orders of o_i and o_j mean different pairs, since o_i will be observed by agent i who will speak first in a game episode. Moreover, each pair $p = (o_i, o_j) \in \mathcal{P}$ has a label l_p. If $o_i = o_j$, then $l_p = 0$; otherwise $l_p = a$ where a is the different attribute between o_i and o_j.

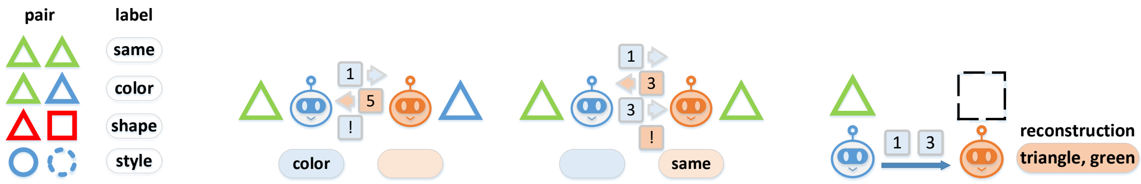
In discrimination game there is a group of homogeneous agents which we call a community. Each game episode involves two agents i and j which are randomly sampled from the community. A pair $p = (o_i, o_j)$ is sampled from $\mathcal{P}$, and the two agents are presented with object o_i and o_j respectively. Then they start the dialog. At each time step t, the speaking agent should choose a symbol s_t from a shared vocabulary V and send it to the other agent. Any agent, after receiving a symbol, can choose to continue or terminate the dialog. If the choice is to continue, then the receiving agent becomes the speaker at the next time step, and the players take turns to speak until the dialog is terminated. Suppose agent j chooses to end the dialog, then it must answer whether o_i and o_j are the same; if not, then which attribute is the different one. In other words, it must pick the true label l_p for the pair (o_i, o_j). If the answer is correct, then both agents succeed and get a reward $r = 1$. Otherwise, they fail and get no reward ($r = 0$). If the number of dialog rounds reaches the upper limit $T_{\max}$, the agents also fail.

Description game. Description game proceeds as follows. First, an agent i observes an input object o_i from $\mathcal{D}_{n,m}$. Next, it chooses a fixed-length (n) sequence of symbols from vocabulary V to describe o_i, and sends it to listener j. Then j consumes all symbols and outputs $\hat{o}_i$. If $o_i = \hat{o}_i$, the agents succeed. The reward for speaker i is according to the game result, namely $r = 1$ if they succeed or $r = 0$ if they fail. The listener j gets rewards according to its reconstruction of each attribute. In our setting, the listener has separate reconstruction models for each attribute, and each of them gets $r = 1$ if its corresponding attribute is reconstructed correctly and gets $r = 0$ otherwise.

4 Experimental setting

For each attribute a, we represent it as a N_a-dimension one-hot vector, where $N_a = m^{(a)}$. An input object o from $\mathcal{D}_{n,m}$ is then represented by the concatenation of all its attributes. Symbolic mapping $\mathrm{map}(\cdot)$ chooses symbols for o and gets the word bank W. The hidden state of the LSTM h_t serves as the memory. When speaking at time step t, one-hot encodings of symbols in W are concatenated to the hidden state h_t and passed to speaking network $g_{\text{sp}}(\cdot)$ to get the probability distribution $\pi_{\text{sp}}(\cdot)$ to produce the symbol.

In discrimination game, we initialize the hidden state h_0 as a zero vector, and each time a symbol s is transmitted in the dialog, s is fed into LSTM $f(\cdot)$. Symbols transmitted in the dialog are encoded as one-hot embeddings. To differentiate the speaker of each symbol, we concatenate a flag to



(a) pair dataset (b) discrimination game (c) description game

Figure 3: Dataset and games.

the embeddings. If the speaker is the agent itself, the flag is zero; otherwise the flag is one. Note that the agent does not know the *identity* of its partner. Whenever receiving a symbol from the partner at time step t, the concatenation of hidden state h_t and input object o is fed to the decision network $\pi^j_{\text{dec}}(\cdot)$, realized by a 2-layer MLP followed by a softmax activation, which outputs an action v_t. The action means continuing the dialog or an answer.

In description game, the speaking process is the same. We fix the message length to n, corresponding to one symbol per attribute. To do this, after the speaker produces a symbol s_t at time step t, the symbol is fed into its LSTM $f(\cdot)$, and the next symbol s_{t+1} is sampled at time step $t+1$. This process proceeds until the fixed message length is reached. The listener is instantiated by n linear layers, which are called reconstruction networks. The message sent by the speaker is represented by the *bag-of-words model* and consumed by the listener. Then each of its reconstruction network outputs an action to predict the value of each attribute.

We use REINFORCE (Williams, 1992) to train each agent. We apply entropy regularization in the loss function to encourage exploration, and use the Adam optimizer with a learning rate of 0.001 in all settings. We run all our experiments three times with different random seeds and present the mean and standard deviation of the results.

5 Metrics

Compositionality. In our setting, the evaluation criterion of compositionality is whether agents can communicate different attributes independently. Note that compositionalty in natural language has more complicated forms, but we only consider the juxtaposition of independent symbols to represent an overall meaning because we hypothesize that compositionality was rather simple when language was formed in the early stage and thus the proposed form is adequate for our research. Inspired by *positional disentanglement* in Chaabouni et al. (2020), we propose a metric called **referential disentanglement** (*refdis*), which measures whether a specific symbol refers to a specific attribute. We ignore the positional information because we need a language suitable for different kinds of interactions, and if symbols' positions are informative, the language is hard to transfer to dialogs.

For each symbol s, we denote a_1^s the attribute that has the lowest conditional entropy given s : $a_1^s = \arg\min_a \mathcal{H}(a|s)$. Similarly, we denote $a_2^s = \arg\min_{a \neq a_1^s} \mathcal{H}(a|s)$. Then we define *refdis* as:

$$\textit{refdis} = \sum_s \left(\frac{\mathcal{H}(a_2^s|s)}{\mathcal{H}(a_2^s)} - \frac{\mathcal{H}(a_1^s|s)}{\mathcal{H}(a_1^s)} \right) \cdot k(s),$$

where $k(s)$ is the frequency of occurrence of symbol s. The intuition of *refdis* is that each symbol should only be informative about one attribute. The best case is when one attribute is determined but all other attributes are totally uncertain given any specific symbol, with *refdis* being 1, and in the worst case the *refdis* is 0. *Context-independence* (CI) proposed in Bogin et al. (2018) shares similar concept with *refdis*, but *refdis* evaluates compositionality according to symbols while CI focuses on the alignment between symbols and concepts.

Symmetry. We evaluate the symmetry of the learned language by computing the Jensen-Shannon divergence between pairs of agents' distributions of different values of attributes, given a specific symbol. For a pair of agents i and j, we define **referential divergence** (*refdiv*) as:

$$\textit{refdiv} = \frac{1}{|V| \cdot n} \sum_s \sum_a \text{JSD}\left(p(m_i^a|a,s) \| p(m_j^a|a,s)\right),$$

where $p(m_i^a|a,s)$ is the value distribution of attribute a of agent i given symbol s. The value of *refdiv* is also between 0 and 1, and a perfectly symmetric communication protocol will get *refdiv* $= 0$.

	Training (%)	Testing (%)	*refdis* ↑	*refdiv* ↓
LSTM	47.62(2.54)	8.42(1.27)	0.07(0.03)	0.87(0.11)
IL	45.67(0.66)	13.47(1.27)	0.06(0.01)	0.87(0.02)

Table 1: The performance of the agent community playing discrimination game on $\mathcal{P}_{3,(3,3,3)}$. LSTM refers to vanilla LSTM-based agents, while IL refers to LSTM agents trained with iterated learning. The first and second column shows the success rate in training set and testing set respectively. Both methods get poor performance.

6 Experiments and results

6.1 Language Learning in Discrimination Game

We first examine the performance of neural network agents learning language in discrimination game. We test two methods: vanilla LSTM, which is aimed to show the performance of simple LSTM-based agents without particular training methods, and iterated learning (IL), which is a framework proposed by evolutionary linguists to simulate the language evolution process, and is believed to help compositional languages emerge (Kirby et al., 2014). To apply IL in our setup, we modify the neural iterated learning algorithm (NIL) proposed by Ren et al. (2020). The implementation details of LSTM and IL can be found in appendix. We use dataset $\mathcal{P}_{3,(3,3,3)}$, where objects have three attributes and each attribute has three values, and split the dataset into the training set and the testing set to explore the generalization ability of the learned languages to unseen objects, which can also reflect compositionality. We set agent number to 3, and the vocabulary size is set to 9. The upper limit for the number of dialog rounds is $T_{\max} = 3$ (each agent has three turns to speak).

Table 1 shows the results, where *refdiv* is averaged over all pairs of agents. Both two methods get poor performance. The success rates reveal that agents encounter difficulties in learning a good policy to accomplish the game, and their learned communication protocols are overfitting the training set, which implies that the language is not compositional. The low *refdis* also verifies this. The results of *refdiv* show that the agents do not converge on symmetric communication protocols. These results confirm that the multi-round dialog game is challenging for a good language to emerge. Methods like iterated learning may also not work well in complex settings, though the IL agents achieve relatively higher testing success rate.

We conjecture that the difficulty may come from the following reasons. For compositionality, the instability of dialogs may push the agents to convey more information each time (*e.g.*, using one symbol to express both two attributes), ending up in a non-compositional communication protocol. For language symmetry, in an agent group, different partners may decode a same message in different ways, and as a result the training will be unstable and hard to converge on a shared communication protocol. Therefore, learning language directly in discrimination game is hard.

6.2 From Simple Tasks to Difficult Tasks

In this section, we want to verify our hypothesis that language can evolve from simple tasks to difficult tasks, and this process, which we call as *task transfer*, helps language learning in difficult tasks. To do this, we first carry out description game on the agent community, and then train the learned speakers to play discrimination game. And we want to investigate whether our proposed symbolic mapping architecture can indeed promote task transfer, so we use LSTM and IL introduced in the previous section to serve as our baselines.

6.2.1 Language Learning in Description Game

To conduct a speaker-listener game in an agent community, most studies make each agent both speaker and listener to simulate a human community (Dubova and Moskvichev, 2020; Dubova et al., 2020). However, since neural agents' speaking and listening policies are not tied together like humans, this setting can be seen as multiple speakers speaking to multiple listeners, making the learning unstable. The multi-listener problem is inevitable in dialog games, but can be avoided in referential games to encourage language symmetry. And through task transfer, the emerged symmetry may be maintained, which becomes a natural way to form symmetric language in dialog games.

Therefore, instead of giving each agent a listening model to interact with all other agents, we choose to use a *shared listener* to simplify and stabilize the language learning and encourage the emergence of language symmetry.

We use dataset $\mathcal{D}_{3,(3,3,3)}$, and set agent number in the community to 3 and vocabulary size to 9, the same as in Section 6.1, and we introduce another agent to play the shared listener role. The message

		Success Rate (%)	*refdis* ↑	*refdiv* ↓
LSTM		100.00(0.00)	0.48(0.07)	0.06(0.04)
IL		100.00(0.00)	0.71(0.09)	0.19(0.03)
SM	protocol	100.00(0.00)	**0.89**(0.06)	0.12(0.03)
	mapping		0.71(0.20)	0.04(0.04)

Table 2: The performance of the agent community playing with a shared listener in description game on $\mathcal{D}_{3,(3,3,3)}$. SM refers to agents with the proposed architecture. The two metrics are calculated on both symbolic mapping and communication protocol for SM agents. All methods get perfect success rate, while SM agents performs the best.

		Training(%)	Testing(%)	*refdis* ↑	*refdiv* ↓
LSTM		85.80(2.82)	51.01(10.14)	0.34(0.05)	0.28(0.08)
IL		51.13(4.87)	15.66(5.82)	0.05(0.03)	0.75(0.09)
SM	protocol	**94.17**(4.98)	**85.35**(8.27)	**0.62**(0.08)	0.18(0.06)
	mapping			0.37(0.09)	**0.06**(0.01)

Table 3: The performance of the agent community playing discrimination game after they have learned to accomplish description game. LSTM and IL show the benefit of task transfer, and SM proves its contribution to task transfer.

length is set to 3. The results are shown in Table 2. SM refers to agents with the proposed architecture in Section 3.2, and for SM agents we calculate the two metrics on both symbolic mapping (which symbols are stored into word bank) and the actual communication protocol (which words are sent to another agent) to explore their relationship. All methods can learn to accomplish the game perfectly, and results of *refdiv* show that agents can converge on symmetric languages more easily now. Besides, the languages that emerge in this game present much higher compositionality compared with language learned in discrimination game, confirming that simple tasks are more suitable for agents to learn language with good properties.

Among the three methods, LSTM agents achieve relatively poor compositionality, showing that agents cannot learn compositionality so well without any environmental pressure, in line with conclusions in other studies. IL agents perform much better in terms of compositionalty, so the method can indeed help in this simpler game. The relatively poor symmetry may be caused by the supervised learning phase in iterated learning, where each new agent learns language from different agents in the past generation. Languages learned by SM agents present best compositionality. This may be because that the symbolic mapping naturally promotes compositionality, since the association between input and symbols can be easily disentangled. High *refdis* and low *refdiv* calculated on symbolic mapping also indicate that after language learning, the mapping can encode good language properties.

6.2.2 Task transfer

After the agents have successfully learned to accomplish description game, we then train the speakers to play discrimination game. For LSTM agents, we use the learned model directly in the new task. For IL agents, we use the learned model to perform task transfer in the first generation. For SM agents, we load the learned symbolic mapping to reinitialized models without fixing the symbolic mapping so that it can continue to evolve. The experiment settings are the same as in Section 6.1.

The results are shown in Table 3. The performance improvement of LSTM and IL compared with that in Table 1 proves the effectiveness of task transfer. Further, the best performance of SM agents confirms the benefit of our proposed architecture. In different kinds of games, agents need different speaking policies, so LSTM and IL agents, who transfer the speaking policies directly, cannot generalize so well to the new game. IL agents perform relatively bad in task transfer probably because in the last few generations when training in the simple game, they reinforce the successful policy again and again, and they learn the policy for the simple game so firmly that the generalization to a new task becomes more difficult. In contrast, SM agents learn a new speaking policy from scratch in the new game, while the symbolic mapping provides knowledge about the learned language implicitly. The results show that this architecture greatly promotes the effect of task transfer.

6.3 Vocabulary Expansion

We have empirically shown that agents' language can evolve in task transfer, and in this section we explore a curriculum on another dimension. In natural language, it is common that vocabulary changes continually over time and new words are created endlessly, so we hope language emerged by agents can also develop. Besides, the emergence of language should not be achieved overnight, and a natural process is to form the language step by step. So we explore the curriculum where the number of objects' attributes increases in a same task. And

		Success Rate (%)	*refdis* ↑	*refdiv* ↓
LSTM		1.56(0.00)	0.00(0.00)	1.00(0.00)
SM	protocol	2.77(0.60)	0.09(0.06)	0.75(0.07)
	mapping		0.03(0.01)	0.18(0.07)

Table 4: The performance of the agent community playing with a shared listener in description game on $\mathcal{D}_{3,(4,4,4)}$.

		Success Rate (%)	*refdis* ↑	*refdiv* ↓
LSTM		100.00(0.00)	0.64(0.12)	0.11(0.06)
SM	protocol	100.00(0.00)	0.84(0.06)	0.12(0.02)
	mapping		0.59(0.18)	0.05(0.01)

Table 5: The performance of the agent community playing with a shared listener in description game on $\mathcal{D}_{2,(4,4)}$.

through this experiment we also want to find out whether symbolic mapping is still useful when the task is the same but the difficulty changes.

We conduct the experiment called *vocabulary expansion*. We first carry out description game using LSTM and SM agents on dataset $\mathcal{D}_{3,(4,4,4)}$ which contains 64 objects. We set agent number to 3 and vocabulary size to 12. The results are shown in Table 4. It is surprising that in this bigger dataset, both methods fail in the simple task. LSTM agents learn only to speak a single word all the time, while the symbolic mappings learned by SM agents are nearly random. The reason is probably that the chance to succeed in this environment is very small at the beginning ($1/64$ here), so the reward is too sparse for reinforcement agents.

Now we try to make agents learn the language from a simpler start. We first train the agents on a smaller dataset $\mathcal{D}_{2,(4,4)}$, and then we introduce a new attribute into the environment and train them on $\mathcal{D}_{3,(4,4,4)}$ with four new symbols available. We also try to reinitialize the speaker network and the LSTM network of SM agents, only retaining the symbolic mapping, to investigate the effect of symbolic mapping in vocabulary expansion. The details of the implementation of the experiments can be found in Appendix B.

Table 5 and Table 6 show the results of the two experiments. While agents can learn good language in the small environment, they can also achieve good performance in the bigger environment now via vocabulary expansion. This demonstrates that language can evolve to become more complicated as the environment develops, and again confirms

		Success Rate (%)	*refdis* ↑	*refdiv* ↓
LSTM		83.85(22.65)	0.47(0.25)	0.14(0.05)
SM	protocol	100.00(0.00)	**0.91**(0.03)	0.11(0.02)
	mapping		0.73(0.10)	0.05(0.01)
SM-RE	protocol	100.00(0.00)	**0.91**(0.01)	0.12(0.04)
	mapping		0.72(0.04)	0.06(0.02)

Table 6: The performance of the agent community playing with a shared listener in description game on $\mathcal{D}_{3,(4,4,4)}$ after vocabulary expansion. SM-RE means the speaker network and the LSTM network of SM agents are reinitialized. Vocabulary expansion is effective, and SM agents perform better.

our hypothesis that the process from simplicity to complexity is crucial for agents to learn language in complex environments. The results also reveal that SM agents are better at vocabulary expansion, as they can not only express new attributes with the help of new symbols, thus achieving higher success rate, but also use the symbols more compositionally. Note that the reinitialized model performs close to the not reinitialized model, showing that symbolic mapping plays a deterministic role for SM agents in vocabulary expansion.

We present an example of the frequencies of different attribute values observed by LSTM and SM agents corresponding to four new symbols in Figure 4. SM agents mainly use the new symbols to express values of the new attribute, showing good compositionality. In contrast, LSTM agents fail to use the new symbols to express accurate meanings after vocabulary expansion. From this perspective, in the curriculum where the task is not changed, the proposed architecture is still helpful.

7 Conclusion

In this paper, we hypothesize that a process from simplicity to complexity is a natural way to help multi-agent language learning. We propose a curriculum learning method called *task transfer*, which uses referential games as the starting point of language learning. We propose *symbolic mapping* and implemented it in LSTM-based agents. This architecture can be applied in different kinds of interactions, so that it can help realize language transfer across different tasks. We also explore another curriculum *vocabulary expansion*. Our results show that learning from simplicity to complexity indeed helps, while symbolic mapping greatly promotes the effect of both task transfer and vocabulary expansion. In summary, we verify our hypothesis

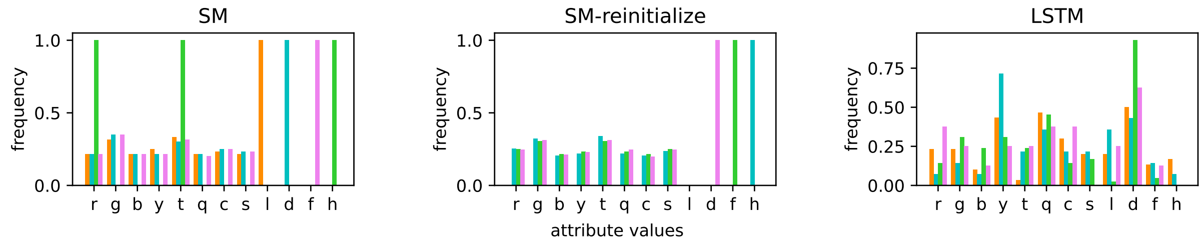


Figure 4: The frequencies of attribute values observed by LSTM and SM agents corresponding to four new symbols in the vocabulary expansion experiment. The four colors of bars correspond to four new symbols respectively. The *x* label is abbreviations of attribute values, and the last four are values of the new attribute.

from two aspects, language transfer and language development, and our proposed architecture symbolic mapping shows remarkable effect.

Limitations

In this section, we discuss some limitations of our work. We conduct preliminary experiments to verify the influence of task transfer and vocabulary expansion on language learning in complex forms, and to explore the effectiveness of our proposed architecture, symbolic mapping, and we assume that language was formed through simple interactions in the early stage. Therefore, additional experiments involving more complex games or other input forms like real images have not been studied and are left for future work. Besides, more advanced language properties and syntax are temporarily not studied in this work. As for task transfer, we verify the effectiveness of a two-stage curriculum starting from referential games, while more advanced curriculum are left for future work, where more cognitive science findings should be involved.

Ethics Statement

We believe our work has no potential risks or negative social impacts now.

Acknowledgements

This work was supported in part by NSF China under grant 62250068. The authors would like to thank the anonymous reviewers for their valuable comments.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Ben Bogin, Mor Geva, and Jonathan Berant. 2018. Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549*.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *ACL*.

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. 2018. Compositional obverter communication learning from raw visual input. In *ICLR*.

Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. 2019. Emergence of compositional language with deep generational transmission. *arXiv preprint arXiv:1904.09067*.

Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2021. Co-evolution of language and agents in referential games. In *EACL*.

Abhishek Das, Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*.

Marina Dubova and Arseny Moskvichev. 2020. Effects of supervision, population size, and self-play on multi-agent reinforcement learning to communicate. In *ALIFE*.

Marina Dubova, Arseny Moskvichev, and Robert Goldstone. 2020. Reinforcement communication learning in different social network structures. *arXiv preprint arXiv:2007.09820*.

Tom Eccles, Yoram Bachrach, Guy Lever, Angeliki Lazaridou, and Thore Graepel. 2019. Biases for emergent communication in multi-agent reinforcement learning. In *NeurIPS*.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. 2018. Emergent communication in a multi-modal, multi-step referential game. In *ICLR*.

Nicole Fitzgerald. 2019. To populate is to regulate. *arXiv preprint arXiv:1911.04362*.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *NeurIPS*.

Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. 2016. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*.

Marta Garnelo and Murray Shanahan. 2019. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. 2019. Emergent linguistic phenomena in multi-agent communication games. In *EMNLP*.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *NeurIPS*.

Jiechuan Jiang and Zongqing Lu. 2018. Learning attentional communication for multi-agent cooperation. In *NeurIPS*.

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

Tomasz Korbak, Julian Zubek, Lukasz Kucinski, Piotr Milos, and Joanna Raczaszek-Leonardi. 2019. Developmentally motivated emergence of compositional communication via template transfer. *arXiv preprint arXiv:1910.06079*.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally'in multi-agent dialog. In *EMNLP*.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. In *ICLR*.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*.

David K. Lewis. 1969. *Convention: A Philosophical Study*. Wiley-Blackwell.

Fushan Li and Michael Bowling. 2019. Ease-of-teaching and language structure from emergent communication. In *NeurIPS*.

Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *AAAI*.

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

Cinjon Resnick, Abhinav Gupta, Jakob N. Foerster, Andrew M. Dai, and Kyunghyun Cho. 2020. Capacity, bandwidth, and compositionality in emergent language learning. In *AAMAS*.

Sainbayar Sukhbaatar, arthur szlam, and Rob Fergus. 2016. Learning multiagent communication with backpropagation. In *NeurIPS*.

Michael Tomasello. 2010. *Origins of human communication*. MIT press.

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

A Training and implementation details

In all of our experiments, each agent's LSTM has a hidden state of size 50, the dimensions of the hidden layers of all MLPs are the same as their input size, and the entropy regularization parameter λ_H is set to 0.05. We train LSTM and SM agents for 10000 epochs in description game and 20000 epochs in discrimination game, unless the agents achieve 100% success rate ahead of time. Our experiments are done using a single GPU GTX 1080 Ti. Most experiments can be done within several hours, while training of IL agents may take more time depending on the number of generations.

The LSTM agents are implemented as LSTM networks with hidden states of size 50. When an LSTM agent observes an object, a linear layer maps the input embedding into the agent's initial hidden state h_0. When speaking, we map the agent's hidden state into a probability distribution over the whole vocabulary with an MLP and a softmax function, and we randomly sample a symbol from the probability distribution. The generated symbol will then be fed back into the LSTM. The decision network is the same as SM agents.

We modify the neural iterated learning algorithm to apply iterated learning in our setup. The IL agents' architecture are the same as LSTM agents. The algorithm runs for several generations, and there are three phases in each generation: learning phase, interacting phase and transmitting phase. At the beginning of each generation, all agents are randomly initialized. When training description game, in the learning phase, each agent in the community learns from data collected in the previous generation with cross-entropy, and the shared listener is pre-trained with REINFORCE by interacting with the pre-trained agent community. In the interacting phase, the agent community plays description game with the shared listener and they are trained the same way as LSTM agents. In the transmitting phase, all objects are fed to each speaking agent, and the corresponding messages generated are stored in a dataset for the next generation. When training discrimination game, in the learning phase, two agents are randomly sampled to learn dialogs with supervised learning from data collected in the previous generation, and the rest agent is pre-trained with REINFORCE by interacting with the pre-trained other two agents. In the interacting phase, the agent community plays discrimination game and they are trained the same way as LSTM agents. In the transmitting phase, two agents are randomly sampled, and the whole training set is fed to them to collect the generated dialogs into a dataset for the next generation. In description game training, we set generation number to 20, pre-train iteration number to 2000 for supervised learning and 3000 for reinforcement learning. We train agents for 2000 epochs in the interacting phase. In discrimination game training, we set generation number to 10, pre-train iteration number to 40000 for supervised learning and 100000 for reinforcement learning. We train agents for 4000 epochs in the interacting phase. We tried a set of hyperparameters and use the ones with the best performance.

B Implementation details of vocabulary expansion

When training the description game on $\mathcal{D}_{2,(4,4)}$, we use zero-padding to object representations and symbol embeddings to encode the new attribute and new symbols, and we set message length to 2. The vocabulary size is set to 8 at first. For LSTM agents, the output number of the speaker network is set to 12, but we mask 4 of them in the first training. When training the three attribute game, the message length is added to 3, and the vocabulary size is expanded to 12. We use the learned model directly for LSTM agents. For SM agents, we reinitialize the agents' symbolic mapping as a linear layer with output dimension $dim = 12$ and set the weights to be zero. Then we load the parameters of the learned symbolic mapping into it. We also try to reinitialize the speaker network and the LSTM network of SM agents, only retaining the symbolic mapping, to investigate the effect of symbolic mapping in vocabulary expansion.

C Examples of the learned symbolic mapping and communication protocol

To show what symbolic mapping learns and how it helps task transfer, we conduct the task transfer experiment on a smaller dataset $\mathcal{D}_{2,(3,3)}$ and present here some examples. We refer to the attributes as *color* and *shape*, and each of them has 3 values (*i.e.*, red, green, blue, triangle, square, circle). The vocabulary size is set to 6, the message length is set to 2 in description game and the upper limit for the number of dialog rounds in discrimination game is $T_{\max} = 2$.

Examples of the learned symbolic mapping in

	red	green	blue
triangle	3,4	0,3,4	2,3,4
square	5	0,5	2,5
circle	1	0,1	1,2

	red	green	blue
triangle	3,4	0,3,4	2,3,4
square	5	0,5	2,5
circle	1,4	0,1	1,2

	red	green	blue
triangle	3,4	0,3,4	2,3,4
square	5	0,5	2,5
circle	1	0,1	1,2

Table 7: The learned symbolic mapping of the three agents in the community when playing with a shared listener in description game on $\mathcal{D}_{2,(3,3)}$.

	red	green	blue
triangle	3,4	0,3,4	2,3,4
square	4,5	0,3,4,5	2,3,5
circle	1,4	0,1,4	1,2

	red	green	blue
triangle	3,4	0,3,4	2,4
square	4,5	0,5	2,5
circle	1,4	0,1,4	1,2

	red	green	blue
triangle	3,4	0,3,4	2,3,4
square	3,4,5	0,3,5	2,3,5
circle	1,3,4	0,1,3	1,2,3

Table 8: The learned symbolic mapping of the three agents in the community when playing discrimination game after they have learned to accomplish description game.

	red	green	blue
triangle	3,4	0,3	2,3
square	5,5	5,0	5,2
circle	1,1	1,0	1,2

	red	green	blue
triangle	4,4	0,4	4,2
square	5,5	5,0	5,2
circle	1,1	1,0	1,2

	red	green	blue
triangle	4,4	0,4	2,4
square	5,5	5,0	5,2
circle	1,1	1,0	1,2

Table 9: The learned communication protocols of the three agents in the community when playing with a shared listener in description game on $\mathcal{D}_{2,(3,3)}$.

	red	green	blue
triangle	4	0	2
square	4,5	0	2,5
circle	1	0	1,2

	red	green	blue
triangle	3	0,4	2,4
square	4,5	0,5	2,5
circle	1,4	0,1	1,2

	red	green	blue
triangle	3	0	2
square	3,5	0,5	2,5
circle	1,3	0,1	1,2

Table 10: The learned communication protocols of the three agents in the community when playing discrimination game after they have learned to accomplish description game.

the agent community is shown in Table 7 and Table 8. They verify that symbolic mapping is not changed greatly across two tasks, so the learned language can be transferred. In both games, all agents associate symbol ‘0’ with attribute ‘green’, ‘1’ with ‘circle’, ‘2’ with ‘blue’ and 5 with ‘square’, which presents good compositionality and symmetry. Symbol ‘3’ and ‘4’ have relatively ambiguous meanings, which is changed between two tasks, but they mainly cover the attributes ‘red’ and ‘triangle’ which cannot be expressed by other symbols. So agents can form compositional structure in symbolic mapping through emergent communication, and the properties like compositionality and symmetry shown in symbolic mapping can explain why symbolic mapping helps language learning through task transfer and why the learned language properties in simple tasks can be maintained in complex tasks by SM agents.

We also present the corresponding communication protocols learned by the agents in the experiment in Table 9 and Table 10. As discrimination game can be terminated at any time, agents may not have chance to express complete information. So in Table 10 we only present all symbols that the agent has spoken in different games after observing a specific object in discrimination game.

Compared with Table 7 and Table 8, the communication protocols make use of the compositional words in symbolic mapping faithfully in both games, so the language is indeed transferred across tasks. Besides, good compositionality and symmetry exhibited in description game are also transferred, which helps success rate in discrimination game.

It may seem odd that the first agent only speaks symbol '0' after observing all green objects in discrimination game. We point out that it results from its game policy: it always expresses 'green' and wait the other agent to communicate about the shape. That may explain why we think speaking policy should not be transferred directly like LSTM agents: policies can be specific to tasks, while only more basic components like symbolic mapping can carry general information about a language.

We should also point out that though the third agent associates symbol '3' with all objects in discrimination game in symbolic mapping, it only speaks it when presented with red objects. This may explain why *refdis* can be higher in protocol compared with mapping.

D Fixed random mapping

We compare the performance of the learned symbolic mapping with a fixed random mapping to show whether the benefit is provided by the reduction of dimensionality. We stop the gradient passed to the symbolic mapping when training so the mapping is randomly initialized and fixed. Since the symbolic mapping is fixed now, it cannot learn anything in the simple task, so the task transfer cannot be performed, and we only keep the mapping the same in the two tasks. We present the results of agents playing in the description game and the discrimination game respectively in Table 11 and Table 12. We run five seeds for each experiment.

Surprisingly, the performance of the fixed random mapping in the simple task is very poor, while the success rate in the difficult task is higher than LSTM agents. From the metrics of the mapping we can find that the random mapping does not show any good properties as the learned symbolic mapping, so it cannot help the policy learning. The poor success rate in description game then shows that dimensionality reduction does not ensure the performance improvement, though it really helps in discrimination game. The reason may be that a random mapping cannot make agents communicate about all attributes well, harmful to the performance in description game, but agents can find ways to accomplish discrimination game when the attributes that can be expressed are limited. However, from the metrics and the success rate in the testing set we can find that the learned language in discrimination game is not compositional, and agents cannot learn a symmetric language with fixed random mappings. So the reduction in dimensionality probably merely helps agents to overfit.

So we can conclude that the performance of symbolic mapping does not benefit from the dimensionality reduction solely, and the learning process is crucial for language emergence with good properties.

		Success Rate (%)	*refdis* ↑	*refdiv* ↓
SM-fix	protocol	47.69(4.92)	0.14(0.03)	0.38(0.04)
	mapping		0.03(0.02)	0.24(0.11)

Table 11: The performance of the agent community with fixed random mapping playing with a shared listener in description game on $\mathcal{D}_{3,(3,3,3)}$.

		Training(%)	Testing(%)	*refdis* ↑	*refdiv* ↓
SM-fix	protocol	80.73(2.66)	50.10(4.72)	0.17(0.02)	0.53(0.06)
	mapping			0.03(0.02)	0.23(0.12)

Table 12: The performance of the agent community with fixed random mapping playing discrimination game.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. We do not use any artifacts.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. We do not use any artifacts.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3.2, Section 3.3

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Our data does not involve these information or content.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. We describe our data in Section 3.3. It is very simple so there is no need for a documentation.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.3, Section 6

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.3, Section 6, Appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6, Section 4

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. We do not use existing packages.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.