Exploring Natural Language Emergence from Multi-Agent Collaboration

Jinan Zhou
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
jinan.zhou@link.cuhk.edu.hk

Abstract

Teaching computers to speak like human is one of the most ambitious goals of artificial intelligence. In order to seal the gap between learning textual statistics and truly understanding human languages, task oriented language models were proposed, where agents learn to communicate while trying to complete certain task under certain environment. However, it was found that the languages invented by machines do not resemble that of human. To solve this problem, we propose an imitation learning scheme. We argue that computers cannot learn to speak like human unless it is told how. Following this idea, we resort to the instructor agents who carry preassigned linguistic knowledge. Instead of learning from stretch, agents are trained against those instructors while performing the given task. Experiments show that the computer can develop human speaking patterns under such settings.

1 Introduction

Learning how human beings speak has been a long-lasting challenge in artificial intelligence. To realize this ambition, tremendous efforts have been devoted into natural language processing, knowledge graph and many other related topics. Recently, the data-driven statistical methods have become the standard practice in solving natural language processing (NLP) tasks. In these methods, statistical models, which are mostly neural networks, are carefully designed by human experts. Then the model will read huge amount of human-generated corpora to learn the structural and statistical features of human languages. While this approach can learn the relations between symbols, it fails to contact the true meaning due to its inability in learning three distinctive features of human language. First, by looking at the text only, data-driven methods are intrinsically prevented from knowing the the mapping of the physical concepts to words, which are also called grounding of languages. Second, they cannot learn the language compositionality, which means how simple concepts are combined to represent more complicated ones. Lastly, statistical models cannot understand the pragmatics of words, namely why this conversation happens and how it is going to change the environment.

Inspired by those drawbacks as well as the fact that human babies learn languages during the interaction with the outer world, goal-driven language models have been proposed. These approaches usually involve training multiple agents to accomplish certain tasks that cannot be solved without collaboration. Also, the form of collaboration is set as communication, which means that the agents can formulate a sequence of symbols from a given lexicon and broadcast this information to others. In this way, the agents are forced to invent their own system of languages. Natural languages with basic composition features that resembles human languages are observed to emerge from this collaboration process [13].

However, such discovery is not acknowledged by all the researchers. Satwik et al. [9] gave a diametrically opposite result that natural language dose not emerge from multi-language dialogues.

According to their experiments, though the agents can complete the task via exchange of symbols, such invented language is decidedly neither interpretable nor compositional. They argued that though it is possible to coax the invented languages to resemble human languages by imposing stricter restrictions on how agents talk, such approach deviates from the goal of generating 'natural' languages. Therefore, a better paradigm of producing natural languages from the interaction with the environment remains to be designed.

To solve the problem that the emerged language system does not resemble human, we propose an alternation of the goal-driven paradigm described above. Notice that the natural acquisition of human languages relies on not only the interaction with the external world, but also on the linguistic instructions from experts, i.e. there are always some language instructors such as parents and teachers who tell the learner explicitly what is correct and what is wrong. Inspired by such facts, we propose to introduce an omniscient agent in the game that plays the roles of this instructor.

Specifically, we will divide the training into two stages that corresponds to the infancy (learn from others) and mature period (learn by oneself and use what have learned) of human beings respectively. In the first stage, we will introduce a hard-coded agent that comes with human wisdom to solve the task. A human-like speaking pattern will be assigned and other learning agents will interactive with this instructor agent. Then we will remove the omniscient agent and observe how the learning agents perform. Hopefully, the learning agents can solve the task while the 'instructors' are absent and learn better and faster than purely learning by themselves.

2 Related Works

Recent years have witnessed tremendous growth in language modelling researches, especially in terms of neural network based approaches. As early as in 2003, neural networks were utilized to tackle the language modelling problem[2]. Then the neural networks evolve from recurrent neural networks [11, 12], to long short-term memory networks [19], and now transformers with attention mechanism [20]. The researches on language modeling related tasks also emerge in large numbers. Neural networks are widely used in machine translation [1, 23, 5], sentiment analysis [10, 4], dialogue systems [21, 17] and many other disciplines.

However, as was mentioned above, the statistical models trained by reading corpora cannot truly comprehend human language due to the lack of grounding, compositional and pragmatics knowledge. Besides, the huge network size that is growing much faster than computer hardware upgrade is gradually corrupting the accessibility of these language models. Some popular models such as Pre-training of deep bidirectional transformers(BERT) [7] and Transformer-XL[6] have a parameter size of around 2M, which is already prohibitive for a wide range of applications. Nevertheless, the arms race has just been escalated to a new level by General Pre-Training (GPT-2) [14] and Megatron-LM [18], which have 1.5B and 8.3B parameters respectively.

In order to tackle the problems above, and to imitate the natural language acquisition process, goal-driven language learning was proposed. These works formulate the language acquisition process as a reinforcement learning problem where multiple agents communicate with each other to complete a given task. During this prcedure, they will invent their own system of language. Jeshua et al. is the first one to introduce reinforcement learning into language modelling and suggested that linguistic systems are constrained optimal policies in multi-agent control problems [3]. Foerster et al.[8] realized end-to-end learning of communication protocols in environments with partial observability. [13] reported the emergence of the languages that resembles human beings during the collaborations amongst agents. On the contrary, Satwik et al. argued that though the agents can accomplish the task via some kind of language, it is decidedly not interpretable or compositional [9]. In this project, we will follow the settings of Satwik et al. and propose an imitation learning approach to solve human language resemblance issue.

3 Problem Formulation

3.1 Environment

In this work, we base our work on reinforcement learning and use the same testbed as [9]. The environment, which is a simple game, is called *Talk & Talk*. In this game, there are two agents, Q-bot and A-bot. The environment has an objects with 3 attributes: shape, color and style. Each attribute has 4 options, formulating $4^3 = 64$ possibilities in total. *Talk & Talk* is conducted in an iterative manner. At the beginning, an instance will be sampled from all possibilities and given to A-bot (say, a purple solid star). In the meanwhile, a task that asks two of the attributes will be assigned to Q-bot (say, color & shape). Then A-bot and Q-bot will conduct two rounds of communication starting from Q-bot asking first. All utterances of agents comes from a given word list. After that, Q-bot will give its answer. If the answer is correct, both agents will receive a reward $R = 1$, otherwise both agents will be penalized with $R = -10$. The environment is illustrated in Figure 1.

Notice that in *Talk & Talk*, there are two levels of information asymmetry between A-bot and Q-bot. One is that A knows what is the object is while B does not. Another is that B knows the question while A does not. This necessitates the collaboration between two agents to complete the task.

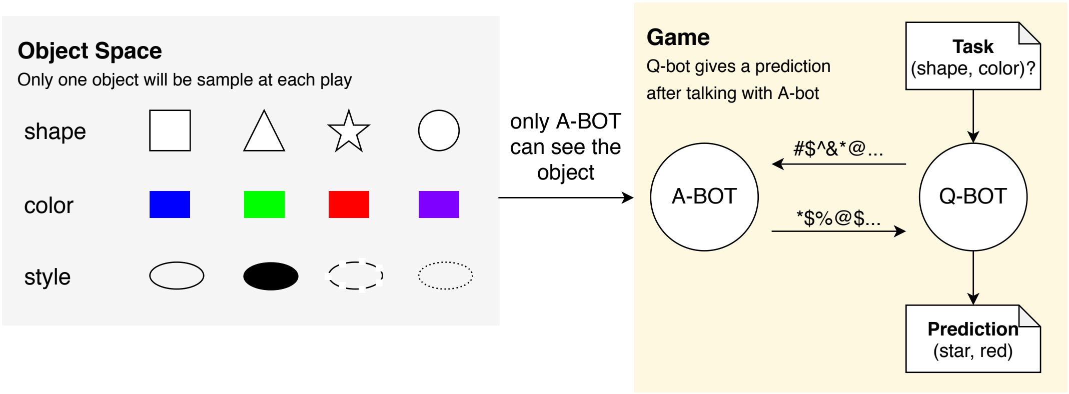
Object Space
Only one object will be sample at each play
shape
color
style
only A-BOT can see the object
Game
Q-bot gives a prediction after talking with A-bot
Task
(shape, color)?
#$^&*@...
*$%@$...
A-BOT
Q-BOT
Prediction
(star, red)

Figure 1: The world in *Talk & Talk*

3.2 Learning Goals

In this work, we hope to achieve three goals:

1. Completeness
   We hope our agents can give the correct answer via proper communication with or without instructor agent.

2. Human-Like
   We hope that the utterances of the both agents resemble human language.

3. Speed
   We hope to learn as fast as possible, and faster than learning without instructor agents.

4 Methods

4.1 Agent and Environment

In this work, both A-bot and Q-bot are designed as agents that interact with the partially observable environment. At each step, the observation of an agent includes either the task G or the object I and the full history of communication. At step t, Q-bot will observe the state $s_Q^t = [G, q_1, a_1, ...q_{t-1}, a_{t-1}]$ and utters a sequence of tokens selected from its own lexicon $q_t = [w_1, w_2, ...w_{d_Q^t}] \in V_Q^{d_Q^t}$. Then, A-bot will observe the state $s_A^t = [I, q_1, a_1, ...q_{t-1}, a_{t-1}, q_t]$ and utters its answer formed by its word list $a_t = [w_1, w_2, ...w_{d_A^t}] \in V_A^{d_A^t}$. At the final step, Q-Bot will give its answer to the question $W^G = (w_1^G, w_2^G)$.

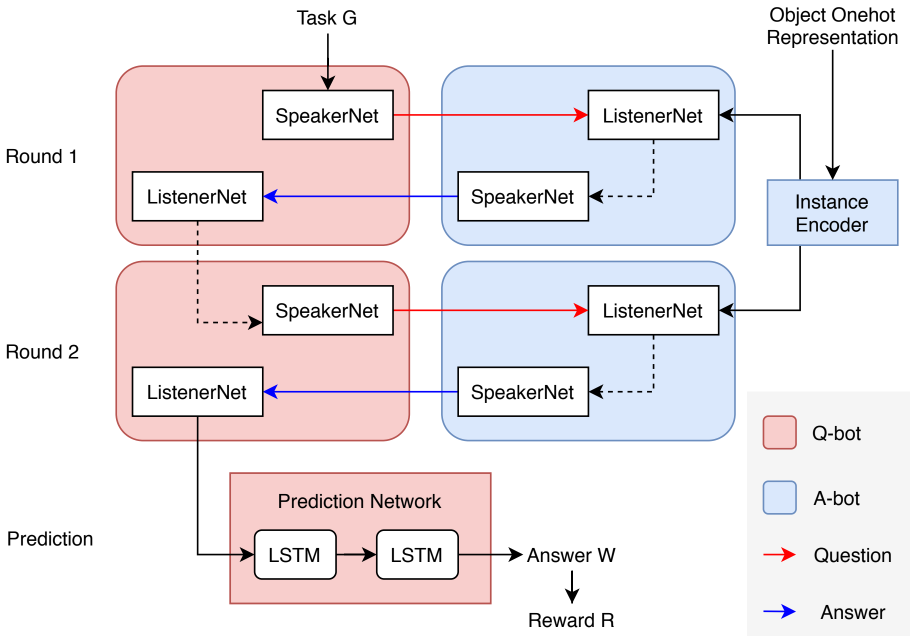

Figure 2: The training workflow and policy network structure for Q-bot and A-bot

4.2 Policy Networks

Our policy network will follow the design in [9]. The Q-bot consists of three parts: a speaker network, a listener network and a prediction network. When the game starts, the task G will be represented as a one hot vector V_G of length 6 because there are in total 6 possible questions. At step t, the speaker network will take state s_Q^t as input and calculate the distribution of utterances q_t via softmax. After A-bot replies, the state is updated by the listener network. At the final step, the prediction LSTM network is invoked twice to generate the final answer.

Similarly, A-bot consists of a speaker network, a listener network and an instance encoder. The speaker network and listener network will function exactly the same as in Q-bot. The instance encoder is a one fully connected layer. It will transform the one hot embedding of the object with size [3,4] into a 6-dimension vector, which unifies with the setting of the Q-bot. The network structure and the training workflow can be represented by Figure 2.

4.3 REINFORCE Algorithm

In the orginal work[9], the policy is trained by REINFORCE algorithm[22]. The loss is defined as the negated reward:

$$\begin{aligned} J(\theta) &= -R \\ &= -\mathbb{1}[W_i^G = W_i^*] + 10 \times \mathbb{1}[W_i^G \neq W_i^*] \end{aligned} \tag{1}$$

where $\mathbb{1}[W_i^G = W_i^*]$ is the number of correct predictions given by the Q-bot and $\mathbb{1}[W_i^G \neq W_i^*]$ is the count of incorrect predictions. The policy gradient is defined as

$$\nabla_\theta J(\theta) = \sum_{i=1}^{m} \sum_{t=0}^{T-1} G_t^{(i)} \nabla_\theta \log(\pi_\theta(a_t^{(i)} | s_t^{(i)})) \tag{2}$$

where $G_t^{(i)} = \sum_{t=0}^{T-1}$ is the return for the i^{th} trajectory. The term $\nabla_\theta \log(\pi_\theta(a_t^{(i)} | s_t^{(i)}))$ is calculated by backpropagating the LSTM networks. The parameters are updated by

$$\theta_{new} = \theta_{old} - \alpha \nabla_\theta J(\theta) \tag{3}$$

4.4 Proximal Policy Optimization

We propose to improve the learning processing by Proximal Policy Optimization (PPO)[16]. In this algorithm, the parameters are optimized to maximize the reward function

$$L(\theta) = \min\left(\frac{\pi(a_t|s_t)}{\pi_{old}(a_t|s_t)}A_t, \text{clip}(\frac{\pi(a_t|s_t)}{\pi_{old}(a_t|s_t)}A_t, 1-\epsilon, 1+\epsilon)A_t\right) \tag{4}$$

Here we use Generalized Advantage Estimation (GAE)[15] to estimate the advantage:

$$A_t^{GAE} = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l} \tag{5}$$

where $\delta_t = r_t + \gamma V(s_t + 1) - V(s_t)$ is the temporal difference.

4.5 Learn with Instructor Agent

According to [9], the system can complete the task upon training, but the language emerged from the process dose not resemble human. To tackle this issue, we propose to introduce an instructor agent. The intuition is that apart from interacting with the external environment, instructions from experts are also highly important in natural language acquisition. During the growth of an child, they are directly told by the his parents and teachers about what is correct and what is wrong, which may be a big factor that enables better and faster learning. Therefore, we would like to see whether introducing omniscient agents that always act correctly and speak like human could yield better results.

In our case, the training are divided into two phases. we will firstly train a A-bot with a Q-bot that always asks that right question and train a Q-bot with a A-bot that always gives the right answer. Considering the simplicity of this *Talk & Talk* game, instructor bots will be implement by some hard-coded condition statements. Then we will train the learning bots together. The intuition is that when human is young, they usually learn from others. After they grow up, they start to learn by themselves and apply his knowledge into practice. And the two stages of training corresponds to this two periods of human development.

5 Experiments

5.1 Experiment 1: Learn without instructor bot

In this experiment, we test the performance of the chat bots without the instructor bot. The performance will be measured in two aspects: accuracy and human language resemblance. Accuracy means that ratio of Q-bot's correct predictions of the object attribute that matches the ground truth. The resemblance to human language, however, is very hard to quantify. Therefore, we will visualize the conversation between the two bots and conduct a case study.

In Standard *Talk & Talk*, we use a minimal word list for the chat bots. For A-bot, we provide the abstract words {0,1,2,3}. During training, the agent is expected to learn a mapping from these symbols to the four possibilities for each attribute. Similarly, the available symbols for Q-bot are {X, Y, Z}. Hopefully, the Q-bot can learn a mapping from each character to an attribute name.

We firstly tried the algorithm used in [9]. After learning, the Q-bot obtained the accuracy of 64.1%. With our PPO improvement, the testing accuracy increased to 84.7%. The training process is shown by Figure 3.

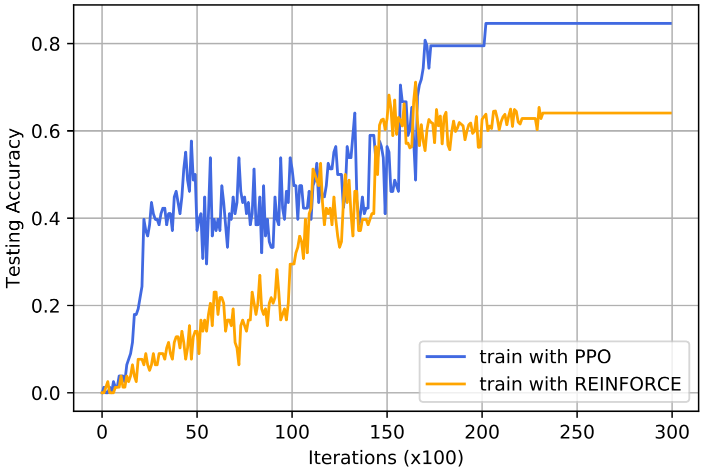

Figure 3: The training result of the original settings in [9]

Table 1: Part of the conversation records in experiment 1

Object	Task	Conversation	Truth	Predication
blue, square, dotted	colors, shapes	Q1:X A1:2 Q2:Y A2:1	blue, square	blue, square
blue, square, dotted	shapes, colors	Q1:X A1:2 Q2:Y A2:1	square, blue	square, blue
blue, square, dotted	colors, styles	Q1:Z A1:1 Q2:Y A2:1	blue, dotted	blue, dotted
blue, square, dotted	styles, colors	Q1:Z A1:1 Q2:Y A2:1	dotted, blue	dotted, blue
blue, square, dotted	shapes, styles	Q1:Z A1:1 Q2:X A2:2	square, dotted	square, dotted
blue, square, dotted	styles, shapes	Q1:Z A1:1 Q2:X A2:2	dotted, square	dotted, square

Table 1 shows a part of the conversation records of the trained bots during testing[1]. In our expectation, the Q-bot should ask one attribute at each round and A-bot should answer accordingly. However, the records shows that a completely different communication strategy was evolved. Notice that under the same object and the same task but in reversed order, the communication between the chat bots are exactly the same. Nevertheless, the Q-bot can give the correct prediction in both scenarios. It indicates that instead of learning to represent each attribute with one symbol, Q-bot develops the strategy to map a whole observation sequence into some prediction $s_Q^t = [G, q1, a1, ...q_2, a_2] \rightarrow W^G = (w_1^G, w_2^G)$. That is to say, instead of asking one attribute at a time and merge the two answers, Q-bot concatenates all the information, including task and replies from A-bot, into one representation, which is then mapped to some final prediction. Though such strategy can do the job, it greatly deviates from how human speaks. Specially, it fails to learn compositionality because it cannot combine the information in interpretable manner. It also lacks a good grounding as the chat bot does not learn a mapping from one symbol to one concept. This result matches with the discovery in [9] that language systems invented by computers are decidedly neither interpretable nor compositional.

5.2 Experiment 2: Learn with instructor bots

In this section, we introduce the instructor bots. We first train a Q-bot with an instructor A-bot, then train an A-bot with an instructor Q-bot. Then the learner A-bot and Q-bot are fine-tuned together. In particular, the actions of the instructor bots are specified by Table 2. Figure 4 shows the learning curve of these three training processes. The final testing accuracy of the bots reaches 94.8%. Comparing Figure 3 and Figure 4, we can see that the bots indeed learn faster and better under this setting.

Table 3 is the same excerpt of the conversation record in this experiment 2[2]. We can see that the utterances of bots completely match the behaviours of instructors. Given the grounding in Table 2, it can be regarded as a human-like conversation.

[1] For the full conversation record, please refer to Figure 5 in Appendix

[2] For the full conversation record, please refer to Figure 6 in Appendix

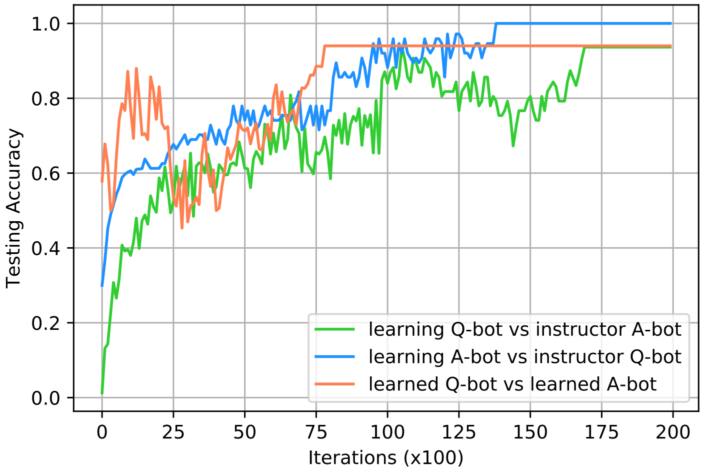

Figure 4: The training result curve under the presence of instructor bots

Table 2: Behaviour assigned to instructor bots

	A-bot				Q-bot		
Utterance	0	1	2	3	X	Y	Z
Meaning	square/ blue/ hollow	triangle/ green/ solid	star/ red/ dashed	circle/ purple/ dotted	shape	color	style

Table 3: Part of the conversation records in experiment 2

Object	**Task**	**Conversation**	**Truth**	**Predication**
blue, square, dotted	colors, shapes	Q1:Y A1:0 Q2:X A2:0	blue, square	blue, square
blue, square, dotted	shapes, colors	Q1:X A1:0 Q2:Y A2:0	square, blue	square, blue
blue, square, dotted	colors, styles	Q1:Y A1:0 Q2:Z A2:3	blue, dotted	blue, dotted
blue, square, dotted	styles, colors	Q1:Z A1:3 Q2:Y A2:0	dotted, blue	dotted, blue
blue, square, dotted	shapes, styles	Q1:X A1:0 Q2:Z A2:3	square, dotted	square, dotted
blue, square, dotted	styles, shapes	Q1:Z A1:3 Q2:X A2:0	dotted, square	dotted, square

6 Conclusion

In this report, we explored the natural language emergence from multi-agent collaborations. We setup the *Talk & Talk* game as the testbed. In this environment, we designed two chat bots to communicate with each other to exchange information. But the languages invented by computers does not resemble human beings. To solve this problem, we proposed an imitation learning mechanism. Inspired by the fact that human learners acquire language skills from experts, we propose to introduce instructor bots who act and speak exactly like human into the problem. The agents are trained against those instructors instead of learning by themselves. Experiments shows that agents learns faster and obtains better performance with instructors. More importantly, it enables the computer to catch the composition and grounding features of human languages.

7 Future Works

This report only discussed a tiny fraction of the natural language emergence problem. As far as we know, the following problems have not yet been explored:

1. Learning with extended word space
   If the agent lexicon contains real words, or even be extend to the total English word space,

dose the imitation learning still work? If yes, will it enjoy a larger advantage than self-learning bots than in this report?

2. Beating the instructor
   In this report, we only managed to make the agents as intelligent as their instructors, which are hard coded by human. However, a brute fact is that this method is not useful unless the trained agents can beat their instructors. Therefore, whether and how to train agents to surpass the their teachers in terms of linguistic wisdom becomes a key problem.

3. Combining data-driven and task-driven model
   Despite the limitations of data-driven models, they have achieved remarkable success in NLP while task-driven models are still at their infancy. Therefore, it is promising to combine these two methodologies to learn the statistics features and language grounding at the same time.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[3] J. Bratman, M. Shvartsman, R. L. Lewis, and S. Singh. A new approach to exploring language emergence as boundedly optimal control in the face of environmental and cognitive constraints. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 7–12. Citeseer, 2010.

[4] E. Cambria. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107, 2016.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, pages 2137–2145, 2016.

[9] S. Kottur, J. Moura, S. Lee, and D. Batra. Natural language does not emerge ‘naturally’ in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[10] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[11] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[12] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011.

[13] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[15] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[17] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[19] M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[21] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

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

[23] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

8 Appendix

A Full Conversation between Bots in Experiment 1

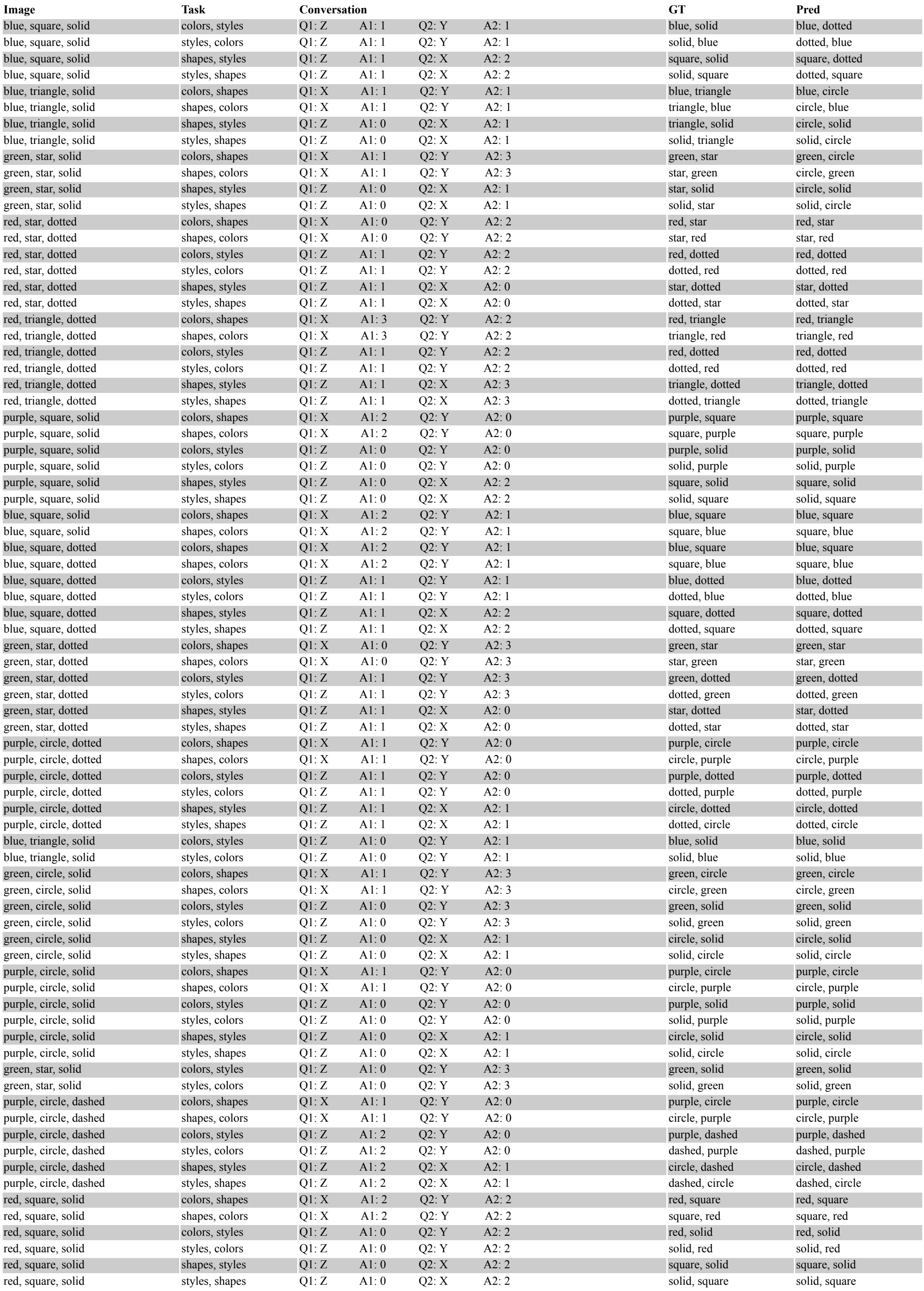

Image	Task	Conversation				GT	Pred
blue, square, solid	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 1	blue, solid	blue, dotted
blue, square, solid	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 1	solid, blue	dotted, blue
blue, square, solid	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 2	square, solid	square, dotted
blue, square, solid	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 2	solid, square	dotted, square
blue, triangle, solid	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 1	blue, triangle	blue, circle
blue, triangle, solid	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 1	triangle, blue	circle, blue
blue, triangle, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 1	triangle, solid	circle, solid
blue, triangle, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 1	solid, triangle	solid, circle
green, star, solid	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 3	green, star	green, circle
green, star, solid	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 3	star, green	circle, green
green, star, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 1	star, solid	circle, solid
green, star, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 1	solid, star	solid, circle
red, star, dotted	colors, shapes	Q1: X	A1: 0	Q2: Y	A2: 2	red, star	red, star
red, star, dotted	shapes, colors	Q1: X	A1: 0	Q2: Y	A2: 2	star, red	star, red
red, star, dotted	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 2	red, dotted	red, dotted
red, star, dotted	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 2	dotted, red	dotted, red
red, star, dotted	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 0	star, dotted	star, dotted
red, star, dotted	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 0	dotted, star	dotted, star
red, triangle, dotted	colors, shapes	Q1: X	A1: 3	Q2: Y	A2: 2	red, triangle	red, triangle
red, triangle, dotted	shapes, colors	Q1: X	A1: 3	Q2: Y	A2: 2	triangle, red	triangle, red
red, triangle, dotted	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 2	red, dotted	red, dotted
red, triangle, dotted	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 2	dotted, red	dotted, red
red, triangle, dotted	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 3	triangle, dotted	triangle, dotted
red, triangle, dotted	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 3	dotted, triangle	dotted, triangle
purple, square, solid	colors, shapes	Q1: X	A1: 2	Q2: Y	A2: 0	purple, square	purple, square
purple, square, solid	shapes, colors	Q1: X	A1: 2	Q2: Y	A2: 0	square, purple	square, purple
purple, square, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 0	purple, solid	purple, solid
purple, square, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 0	solid, purple	solid, purple
purple, square, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 2	square, solid	square, solid
purple, square, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 2	solid, square	solid, square
blue, square, solid	colors, shapes	Q1: X	A1: 2	Q2: Y	A2: 1	blue, square	blue, square
blue, square, solid	shapes, colors	Q1: X	A1: 2	Q2: Y	A2: 1	square, blue	square, blue
blue, square, dotted	colors, shapes	Q1: X	A1: 2	Q2: Y	A2: 1	blue, square	blue, square
blue, square, dotted	shapes, colors	Q1: X	A1: 2	Q2: Y	A2: 1	square, blue	square, blue
blue, square, dotted	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 1	blue, dotted	blue, dotted
blue, square, dotted	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 1	dotted, blue	dotted, blue
blue, square, dotted	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 2	square, dotted	square, dotted
blue, square, dotted	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 2	dotted, square	dotted, square
green, star, dotted	colors, shapes	Q1: X	A1: 0	Q2: Y	A2: 3	green, star	green, star
green, star, dotted	shapes, colors	Q1: X	A1: 0	Q2: Y	A2: 3	star, green	star, green
green, star, dotted	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 3	green, dotted	green, dotted
green, star, dotted	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 3	dotted, green	dotted, green
green, star, dotted	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 0	star, dotted	star, dotted
green, star, dotted	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 0	dotted, star	dotted, star
purple, circle, dotted	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 0	purple, circle	purple, circle
purple, circle, dotted	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 0	circle, purple	circle, purple
purple, circle, dotted	colors, styles	Q1: Z	A1: 1	Q2: Y	A2: 0	purple, dotted	purple, dotted
purple, circle, dotted	styles, colors	Q1: Z	A1: 1	Q2: Y	A2: 0	dotted, purple	dotted, purple
purple, circle, dotted	shapes, styles	Q1: Z	A1: 1	Q2: X	A2: 1	circle, dotted	circle, dotted
purple, circle, dotted	styles, shapes	Q1: Z	A1: 1	Q2: X	A2: 1	dotted, circle	dotted, circle
blue, triangle, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 1	blue, solid	blue, solid
blue, triangle, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 1	solid, blue	solid, blue
green, circle, solid	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 3	green, circle	green, circle
green, circle, solid	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 3	circle, green	circle, green
green, circle, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 3	green, solid	green, solid
green, circle, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 3	solid, green	solid, green
green, circle, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 1	circle, solid	circle, solid
green, circle, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 1	solid, circle	solid, circle
purple, circle, solid	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 0	purple, circle	purple, circle
purple, circle, solid	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 0	circle, purple	circle, purple
purple, circle, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 0	purple, solid	purple, solid
purple, circle, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 0	solid, purple	solid, purple
purple, circle, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 1	circle, solid	circle, solid
purple, circle, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 1	solid, circle	solid, circle
green, star, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 3	green, solid	green, solid
green, star, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 3	solid, green	solid, green
purple, circle, dashed	colors, shapes	Q1: X	A1: 1	Q2: Y	A2: 0	purple, circle	purple, circle
purple, circle, dashed	shapes, colors	Q1: X	A1: 1	Q2: Y	A2: 0	circle, purple	circle, purple
purple, circle, dashed	colors, styles	Q1: Z	A1: 2	Q2: Y	A2: 0	purple, dashed	purple, dashed
purple, circle, dashed	styles, colors	Q1: Z	A1: 2	Q2: Y	A2: 0	dashed, purple	dashed, purple
purple, circle, dashed	shapes, styles	Q1: Z	A1: 2	Q2: X	A2: 1	circle, dashed	circle, dashed
purple, circle, dashed	styles, shapes	Q1: Z	A1: 2	Q2: X	A2: 1	dashed, circle	dashed, circle
red, square, solid	colors, shapes	Q1: X	A1: 2	Q2: Y	A2: 2	red, square	red, square
red, square, solid	shapes, colors	Q1: X	A1: 2	Q2: Y	A2: 2	square, red	square, red
red, square, solid	colors, styles	Q1: Z	A1: 0	Q2: Y	A2: 2	red, solid	red, solid
red, square, solid	styles, colors	Q1: Z	A1: 0	Q2: Y	A2: 2	solid, red	solid, red
red, square, solid	shapes, styles	Q1: Z	A1: 0	Q2: X	A2: 2	square, solid	square, solid
red, square, solid	styles, shapes	Q1: Z	A1: 0	Q2: X	A2: 2	solid, square	solid, square

Figure 5: The full conversation record of bots on testing set in experiment 1

B Full Conversation between Bots in Experiment 2

Image	Task	Conversation	GT	Pred
blue, square, solid	colors, styles	Q1: Y A1: 0 Q2: Z A2: 1	blue, solid	blue, solid
blue, square, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 0	solid, blue	solid, blue
blue, square, solid	shapes, styles	Q1: X A1: 0 Q2: Z A2: 1	square, solid	square, solid
blue, square, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 0	solid, square	solid, square
blue, triangle, solid	colors, shapes	Q1: Y A1: 0 Q2: X A2: 1	blue, triangle	blue, triangle
blue, triangle, solid	shapes, colors	Q1: X A1: 1 Q2: Y A2: 0	triangle, blue	triangle, blue
blue, triangle, solid	shapes, styles	Q1: X A1: 1 Q2: Z A2: 1	triangle, solid	triangle, solid
blue, triangle, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 1	solid, triangle	solid, triangle
green, star, solid	colors, shapes	Q1: Y A1: 1 Q2: X A2: 2	green, star	green, star
green, star, solid	shapes, colors	Q1: X A1: 2 Q2: Y A2: 1	star, green	star, green
green, star, solid	shapes, styles	Q1: X A1: 2 Q2: Z A2: 1	star, solid	star, solid
green, star, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 2	solid, star	solid, star
red, star, dotted	colors, shapes	Q1: Y A1: 2 Q2: X A2: 2	red, star	red, star
red, star, dotted	shapes, colors	Q1: X A1: 2 Q2: Y A2: 2	star, red	star, red
red, star, dotted	colors, styles	Q1: Y A1: 2 Q2: Z A2: 3	red, dotted	red, dotted
red, star, dotted	styles, colors	Q1: Z A1: 3 Q2: Y A2: 2	dotted, red	dotted, red
red, star, dotted	shapes, styles	Q1: X A1: 2 Q2: Z A2: 3	star, dotted	star, dotted
red, star, dotted	styles, shapes	Q1: Z A1: 3 Q2: X A2: 2	dotted, star	dotted, star
red, triangle, dotted	colors, shapes	Q1: Y A1: 2 Q2: X A2: 1	red, triangle	red, triangle
red, triangle, dotted	shapes, colors	Q1: X A1: 1 Q2: Y A2: 2	triangle, red	triangle, red
red, triangle, dotted	colors, styles	Q1: Y A1: 2 Q2: Z A2: 3	red, dotted	red, dotted
red, triangle, dotted	styles, colors	Q1: Z A1: 3 Q2: Y A2: 2	dotted, red	dotted, red
red, triangle, dotted	shapes, styles	Q1: X A1: 1 Q2: Z A2: 3	triangle, dotted	triangle, dotted
red, triangle, dotted	styles, shapes	Q1: Z A1: 3 Q2: X A2: 1	dotted, triangle	dotted, triangle
purple, square, solid	colors, shapes	Q1: Y A1: 3 Q2: X A2: 0	purple, square	purple, square
purple, square, solid	shapes, colors	Q1: X A1: 0 Q2: Y A2: 3	square, purple	square, purple
purple, square, solid	colors, styles	Q1: Y A1: 3 Q2: Z A2: 1	purple, solid	purple, solid
purple, square, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 3	solid, purple	solid, purple
purple, square, solid	shapes, styles	Q1: X A1: 0 Q2: Z A2: 1	square, solid	square, solid
purple, square, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 0	solid, square	solid, square
blue, square, solid	colors, shapes	Q1: Y A1: 0 Q2: X A2: 0	blue, square	blue, square
blue, square, solid	shapes, colors	Q1: X A1: 0 Q2: Y A2: 0	square, blue	square, blue
blue, square, dotted	colors, shapes	Q1: Y A1: 0 Q2: X A2: 0	blue, square	blue, square
blue, square, dotted	shapes, colors	Q1: X A1: 0 Q2: Y A2: 0	square, blue	square, blue
blue, square, dotted	colors, styles	Q1: Y A1: 0 Q2: Z A2: 3	blue, dotted	blue, dotted
blue, square, dotted	styles, colors	Q1: Z A1: 3 Q2: Y A2: 0	dotted, blue	dotted, blue
blue, square, dotted	shapes, styles	Q1: X A1: 0 Q2: Z A2: 3	square, dotted	square, dotted
blue, square, dotted	styles, shapes	Q1: Z A1: 3 Q2: X A2: 0	dotted, square	dotted, square
green, star, dotted	colors, shapes	Q1: Y A1: 1 Q2: X A2: 2	green, star	green, star
green, star, dotted	shapes, colors	Q1: X A1: 2 Q2: Y A2: 1	star, green	star, green
green, star, dotted	colors, styles	Q1: Y A1: 1 Q2: Z A2: 3	green, dotted	green, dotted
green, star, dotted	styles, colors	Q1: Z A1: 3 Q2: Y A2: 1	dotted, green	dotted, green
green, star, dotted	shapes, styles	Q1: X A1: 2 Q2: Z A2: 3	star, dotted	star, dotted
green, star, dotted	styles, shapes	Q1: Z A1: 3 Q2: X A2: 2	dotted, star	dotted, star
purple, circle, dotted	colors, shapes	Q1: Y A1: 3 Q2: X A2: 3	purple, circle	purple, circle
purple, circle, dotted	shapes, colors	Q1: X A1: 3 Q2: Y A2: 3	circle, purple	circle, purple
purple, circle, dotted	colors, styles	Q1: Y A1: 3 Q2: Z A2: 3	purple, dotted	purple, dotted
purple, circle, dotted	styles, colors	Q1: Z A1: 3 Q2: Y A2: 3	dotted, purple	dotted, purple
purple, circle, dotted	shapes, styles	Q1: X A1: 3 Q2: Z A2: 3	circle, dotted	circle, dotted
purple, circle, dotted	styles, shapes	Q1: Z A1: 3 Q2: X A2: 3	dotted, circle	dotted, circle
blue, triangle, solid	colors, styles	Q1: Y A1: 0 Q2: Z A2: 1	blue, solid	blue, solid
blue, triangle, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 0	solid, blue	solid, blue
green, circle, solid	colors, shapes	Q1: Y A1: 1 Q2: X A2: 3	green, circle	green, circle
green, circle, solid	shapes, colors	Q1: X A1: 3 Q2: Y A2: 1	circle, green	circle, green
green, circle, solid	colors, styles	Q1: Y A1: 1 Q2: Z A2: 1	green, solid	green, solid
green, circle, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 1	solid, green	solid, green
green, circle, solid	shapes, styles	Q1: X A1: 3 Q2: Z A2: 1	circle, solid	circle, solid
green, circle, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 3	solid, circle	solid, circle
purple, circle, solid	colors, shapes	Q1: Y A1: 3 Q2: X A2: 3	purple, circle	purple, circle
purple, circle, solid	shapes, colors	Q1: X A1: 3 Q2: Y A2: 3	circle, purple	circle, purple
purple, circle, solid	colors, styles	Q1: Y A1: 3 Q2: Z A2: 1	purple, solid	purple, solid
purple, circle, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 3	solid, purple	solid, purple
purple, circle, solid	shapes, styles	Q1: X A1: 3 Q2: Z A2: 1	circle, solid	circle, solid
purple, circle, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 3	solid, circle	solid, circle
green, star, solid	colors, styles	Q1: Y A1: 1 Q2: Z A2: 1	green, solid	green, solid
green, star, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 1	solid, green	solid, green
purple, circle, dashed	colors, shapes	Q1: Y A1: 3 Q2: X A2: 3	purple, circle	purple, circle
purple, circle, dashed	shapes, colors	Q1: X A1: 3 Q2: Y A2: 3	circle, purple	circle, purple
purple, circle, dashed	colors, styles	Q1: Y A1: 3 Q2: Z A2: 2	purple, dashed	purple, dashed
purple, circle, dashed	styles, colors	Q1: Z A1: 2 Q2: Y A2: 3	dashed, purple	dashed, purple
purple, circle, dashed	shapes, styles	Q1: X A1: 3 Q2: Z A2: 2	circle, dashed	circle, dashed
purple, circle, dashed	styles, shapes	Q1: Z A1: 2 Q2: X A2: 3	dashed, circle	dashed, circle
red, square, solid	colors, shapes	Q1: Y A1: 2 Q2: X A2: 0	red, square	red, circle
red, square, solid	shapes, colors	Q1: X A1: 0 Q2: Y A2: 2	square, red	circle, red
red, square, solid	colors, styles	Q1: Y A1: 2 Q2: Z A2: 1	red, solid	red, solid
red, square, solid	styles, colors	Q1: Z A1: 1 Q2: Y A2: 2	solid, red	solid, red
red, square, solid	shapes, styles	Q1: X A1: 0 Q2: Z A2: 1	square, solid	circle, solid
red, square, solid	styles, shapes	Q1: Z A1: 1 Q2: X A2: 0	solid, square	solid, circle

Figure 6: The full conversation record of bots on testing set in experiment 2