The Relaying Game

Maarten Burger
University of Amsterdam

Rico Mossinkoff
University of Amsterdam

Abstract

In this report we extend a referential game by introducing new agents that relay information. We analyse the effect on the emerged languages using statistical message evaluation and unsupervised grammar induction techniques. We show that the change in emerged language is significant when information is relayed. Our experiments show that even with very little grammar structure and many agents, some information is still retained. We argue that further research should be done in this area.

1 Introduction

As we use languages everyday to communicate with each other, it is useful to learn more about the structure and rules that make a language. With the rise of new Natural Language Processing (NLP) techniques, it is important to understand how computers use and understand languages (Kirby, 2002). With the use of referential games, we can find out how languages evolve and what characteristics a language has. We will study a referential game similar to (Lazaridou et al., 2016) and (Havrylov and Titov, 2017) where a sender and receiver try to transmit information via a message. They are jointly trained with some given task. This task could be the sender encoding an image into a message and the receiver decoding this message to retrieve the correct image. After a while, the sender and the receiver will develop their own language to communicate the necessary information in the image. In this research, we will extend this setup with the use of 'relayers'. These relayers will act as intermediate players that receive an embedding, decode it into a message, reconstruct the embedding using this message and send it to the next agent. This extended setup can simulate the human 'pass the message' game. In this game, a message is passed from person to person with the final person trying to reconstruct the original message. This game shows that message information is often lost when there are multiple actors involved. We aim to investigate if computer actors encounter this same rate of information loss when multiple actors are involved. Therefore our first research question is as follows: "What is the rate of information decay in a referential game with various intermediate actors?". As we will use identical intermediate actors, we hypothesise that the information decay should be linear with the amount of intermediate actors. We are also interested if the generated language has some kind of a grammar. Therefore, our second research question will be: "Do the generated languages have a linguistic structure and how does this change for games with various intermediate actors?". We hypothesise that the linguistic structure will decrease with a higher amount of intermediate actors.

2 Previous work

There has been quite some interesting work in the area of referential games. We will outline some of the most prominent ones in this section.

The research from (Dagan et al., 2020) investigates referential games with a transmission bottleneck. This bottleneck is enforced by introducing new agents that learn to communicate with more experienced agents. They show that this cultural transmission results in an improvement in language efficiency and communicative success. Other research from (Luna et al., 2020) shows that emerging languages will be better at generalisation with less redundancy if sources of pressure are applied on the agent communication.

In our research we enforce a different type of bottleneck. We use intermediate actors that try to reconstruct an image from a message and sending their reconstructed image to the next agent. This bottleneck forces the agents to only include necessary information, as faulty information has a higher chance to corrupt the message along the way.

Another study (Kirby et al., 2014) investigated

how agents can learn behaviour by exposure to another individual (iterated learning). Furthermore, (Ren et al., 2020) investigate compositionality in emerged languages with iterated learning. They propose an algorithm to obtain a more compositional language, which results in better generalizing power and faster training. Yet another study (Smith et al., 2003) shows that iterated learning can lead to the emergence of linguistic structure. In our research, we try to measure the formation of emergent linguistic structure by using the CCL-BMM (Seginer, 2007; Stolcke and Omohundro, 1994) algorithm, as proposed by (van der Wal et al., 2020). With this we can measure the emergence of grammar, which can be seen as a strong indicator for compositionality as a language is compositional if it has a compositional grammar (Kracht, 2007).

3 Experimental setup

3.1 Experiments

During this research, we use several different agents to play the 'pass the message' game. These agents are the relayer, the sender and the receiver. In our version of the referential game, the sender creates a message from embedded images of the CIFAR-100 dataset (Krizhevsky et al., 2009). The receiver is supposed to receive this message and choose the correct image from a collection of images. However, in our version, before the sender sends this message to a regular receiver, it sends the message to a relaying receiver instead, such a pair forms a relayer. A relayer tries to reconstruct the original embedded image from a message. The relayer then sends this embedding to the next agent which is either a regular sender and receiver pair, or another relayer (which is consists of a sender and relaying receiver pair). There can be multiple relayers in a sequence. The last sender and receiver pair then try to choose the correct image. From now on, we will refer to this game as the 'Relaying Game', it is an extension of the referential game by (Havrylov and Titov, 2017). All agents use a vocabulary size of 100 and a maximum message length of 10. We decided to use embeddings instead of the full images for computational efficiency. Furthermore, the images are embedded once using a pre-trained vision module and are not changed thereafter.

The last (or first when no relayers are present) sender and receiver agents are trained using a cross entropy loss as the receiver outputs the probabilities for the images to choose from. The relayer is trained separately as it effectively functions as an Autoencoder (Liou et al., 2014). We train the relayer using an L1-loss function. The sender part of the relayer has the exact same architecture as the regular sender. The difference lies between the relaying receiver and the regular receiver, as the relaying receiver outputs an image embedding rather than probabilities. This way, relayers can be set up in a sequential fashion in order to simulate a 'pass the message' game.

For our experiments we used several configurations. For the agent setup, we will use a sender and a receiver with 0 (baseline), 1, 2, 3, 5 and 10 relayers, we will also compare our results to an untrained (random) agent. For the Game Size, which is the amount of images that the receiver will need to choose from, we will use a size of 2, 3, 5 and 10. We will test all possible combinations of 'Number of relayers' and 'Game Size'.

3.2 Evaluation metrics

To evaluate our results, we will use a variety of different metrics. We generate messages to evaluate by using our test set which consists of 10000 embedded images and pass them through the model, which generates 10000 messages at each appropriate point. To evaluate the generated messages, we use the metrics shown in Table 1. To evaluate the performance of different configurations, we will measure the receiver accuracy. This accuracy is defined as $\frac{N_{correct}}{N_{total}}$. The emerged language grammar will be evaluated using CCL-BMM as proposed by (van der Wal et al., 2020). The metrics we will report here can be found in Table 2.

We will also apply statistical tests to our results to ensure significance in our conclusions.

Metric	Description
|UM|	Total amount of unique messages
|UT|	Total amount of unique tokens
μ **msg length**	Average message length
σ **msg length**	Standard deviation of the message length
μ **unique tokens**	Average amount of unique tokens in a message
σ **unique tokens**	Standard deviation of the amount of unique tokens in a message

Table 1: Message evaluation metrics.

4 Results

In this section we will present the most interesting results we gathered from our experiments.

As can be seen in figure 1, the accuracy of the receiver drops when the amount of relayers is increased. This drops seems to be more extreme

Metric	Description
term.	Total amount of terminal symbols
preterm.	Total amount of preterminal symbols
recurs.	Total amount of recursive production rules
nominals	Total amount of nominal categories
pt groups	Total amount of unique pre-terminal groups
coverage	Ratio of messages that the grammar can parse

Table 2: Grammar evaluation metrics.

with higher game sizes, as the random chance is lower here. However, around 3 relayers, the accuracy seems to reach a point of convergence. This point of convergence still has an accuracy value that seems to be approximately twice as high as the expected accuracy of a receiver that chooses an image randomly.

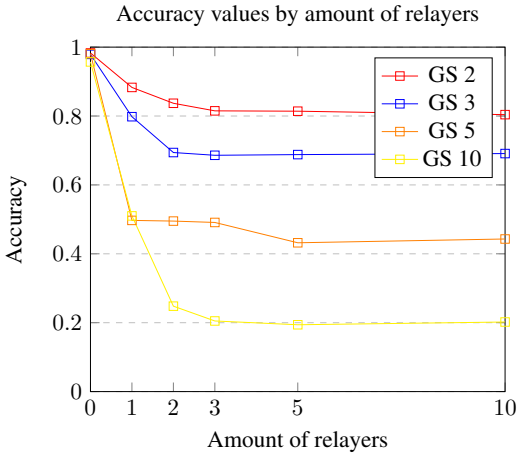

Figure 1: Accuracy values by amount of relayers for different game sizes (GS).

Another interesting result is shown in Table 3. This is a summary of the full table that is included in appendix B.2. We can see that the amount of unique messages, unique tokens, unique tokens per message and the variance of unique tokens per message all decrease for a higher number of relayers. This difference is significant for all of these variables, with the exception of the difference between 1 and 2 relayers. The full significance tests are included in appendix E.

Nr. of relayers	|UM|	|UT|	μ UT	σ UT
1	8225	37	5.203	1.946
2	7615	35	5.163	1.885
3	6957	35	5.05	1.827
4	5990	35	4.906	1.755
5	4780	35	4.792	1.713

Table 3: Metrics for the generated 10000 messages from the relayer emergent languages.

Apart from the general language analysis for the relayer, we also analysed the final messages that were sent to the last receiver by the sender. We have compiled interesting results in Table 4. These results are gathered from configurations with game size 2, the full table and other game sizes are included in appendix C.

Game Type	|UM|	|UT|	μ UT	σ UT
Random	2104	56	2.16	1.293
Baseline	7343	38	3.576	1.426
Relaying Game 1	17	12	1.095	0.611
Relaying Game 2	6	4	0.589	0.492
Relaying Game 3	5	4	0.925	0.495
Relaying Game 5	5	4	0.919	0.495
Relaying Game 10	5	5	0.916	0.812

Table 4: Metrics for 10000 generated messages of the final emergent language for game size 2, Relaying Game n refers to the Relaying Game played with n relayers.

We also evaluated the grammar from both the relayer and the sender-receiver using the CCL-BMM algorithm as proposed by (van der Wal et al., 2020). In Table 5 we show interesting results for the relayer language and in Table 6 we show a few interesting results for the final messages, the full tables can be found in appendix B.3 and D.1.

Language Type	term.	preterm.	recurs.	nom.	pt groups	coverage
Random	44	8	1	6	24	100
1 Relayer	35	30	6	47	964	100
2 Relayers	33	30	2	50	910	100
3 Relayers	31	27	8	68	803	100
5 Relayers	34	27	3	80	835	100
10 Relayers	2	2	10	20	20	100

Table 5: Various metrics for evaluating emergent languages of the relayer. In this table n Relayers refers to the language that emerges when n passes through a relayer occured.

Game Type	term.	preterm.	recurs.	nom.	pt groups	coverage
Random	44	8	1	6	24	100
Baseline	38	25	58	230	336	99.60
Relaying Game 1	6	6	0	2	5	29.26
Relaying Game 2	3	3	0	1	1	57.20
Relaying Game 3	2	2	0	1	1	58.43
Relaying Game 5	2	2	0	1	1	58.16
Relaying Game 10	2	2	0	1	1	94.31

Table 6: Various metrics for evaluating emergent languages of the final messages using game size 2. Relaying Game n refers to the Relaying Game played with n relayers.

5 Conclusion and Discussion

While we see the accuracy drop when using more relayers, we still see that the accuracy is far above the performance of a random agent. This could be an indication that, even through 10 relayers, some information is still retained. Similarly we still see

that grammar emerges from the relayer language (even after 10 relayers) indicating that once again, some information is retained. Our hypothesis that the information loss would drop linearly does not seem to hold in our results. Most of the information seems to be lost at the first relayers, as the accuracy converges around 2-3 relayers. From there, adding relayers does not seem to effect the accuracy. This result is quite interesting, because we would expect that an equal amount of information would be lost for every relayer reconstruction. However, this does not seem to be the case.

If we look at the generated language of the relayer, we can see that the amount of unique messages and message tokens decreases when we increase the number of relayers. This could be explained by information loss when passing a message through a sequence of agents. When more agents are involved, some very specific and rare words are more likely to get lost and the more general and frequent ones are more likely to be kept. The same could hold for tokens inside a message. An interesting result here is that the accuracy of games with 3 or more relayers stays roughly the same, while the uniqueness of messages and tokens decreases. This could indicate that the relayers are able to communicate a similar amount of information with a language that is less expressive. Which can be interpreted as a more efficient communication between relayers.

The generated language between the sender and the receiver shows a similar decrease as for the relayer language. Because the relayers are relaying information to the sender and the receiver, it makes sense that the final generated language suffers from some form of information decay.

We can conclude that the generated languages have a linguistic structure and therefore a form of compositionality. In our grammar analysis, we consider terminals, preterminals and recursives to be 'grammar rules' that are present in the language. In Table 5 we can see that these variables decrease when we increase the number of relayers. This would mean that the language becomes more structured. With less unique tokens, perhaps there is a greater need to find meaning in structure. Structure might also be harder to corrupt when relaying as opposed tokens.

Furthermore, the coverage is 100 for all relayer languages. This means that all messages in the test set can be explained with the generated language. The ability to generalize does not seem to vanish for a higher number of relayers.

In Table 5, we can see that there is a big decrease in all metrics except for coverage for 10 relayers, compared to the others. This phenomenon seems strange as it could to indicate that somehow between 5 and 10 relayers the language evolves to become highly structured. However, as this is the result of only a single experiment run, this needs to be investigated more thoroughly to derive any concrete conclusions.

The grammar of the language of the final sender and receiver seems to be quite consistent when we increase the number of relayers. Interesting to note is that the coverage is very low for a game with 1 relayer, while it increases when the amount of relayers increases. This is a notable result, as the language is better able to generalize when we increase relayers. This could be due to the language becoming more robust to noise.

A glaring shortcoming of this work is the fact that while we used a random seed, we did only perform a single run for each configuration. As such we have no indicator of variance between different runs and it is possible that some of our results are statistical anomalies. Furthermore, due to computational restrictions we only induced grammar on 10 percent of our available messages and only for Game Size 2. For the same reason we did not perform a hyperparameter search except for a very small manual one.

6 Future work

As stated previously, we see an indication that some information is retained, even through many relayers. Further research could explore exactly what information or information type is retained here.

Due to computational constraints we could not perform the grammar induction on our full test message set. Perhaps there are more insights to be gained into the structure of the emergent languages if the full test sets were to be used.

Due to time constraints we did not fully test whether the final sender receiver agents would be more robust to noise. Which could be the case due to the embedding outputted by the relayer being imperfect. If it is, training such agents using relayers could be a way to make such agents more robust to noise. Further research could explore this possibility.

References

Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2020. Co-evolution of language and agents in referential games. *CoRR*, abs/2001.03361.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30.

Simon Kirby. 2002. Natural language from artificial life. *Artificial life*, 8(2):185–215.

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

Marcus Kracht. 2007. Compositionality: The very idea. *Research on Language and Computation*, 5(3):287–308.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. 2014. Autoencoder for words. *Neurocomputing*, 139:84–96.

Diana Rodríguez Luna, Edoardo Maria Ponti, Dieuwke Hupkes, and Elia Bruni. 2020. Internal and external pressures on language emergence: least effort, object constancy and frequency. *CoRR*, abs/2004.03868.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. 2020. Compositional languages emerge in a neural iterated learning model. *CoRR*, abs/2002.01365.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic. Association for Computational Linguistics.

Kenny Smith, Simon Kirby, and Henry Brighton. 2003. Iterated learning: A framework for the emergence of language. *Artificial life*, 9:371–86.

Andreas Stolcke and Stephen Omohundro. 1994. Inducing probabilistic grammars by bayesian model merging. In *International Colloquium on Grammatical Inference*, pages 106–118. Springer.

Oskar van der Wal, Silvan de Boer, Elia Bruni, and Dieuwke Hupkes. 2020. The grammar of emergent languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3339–3359.

Appendix

A Accuracy Table

In this section we report the accuracy of the test set on the final epoch for each model. Relaying game n refers to the Relaying Game played with n passes through a relayer.

	Game Size 2	Game Size 3	Game Size 5	Game Size 10
Random	0.513	0.314	0.198	0.103
Baseline	0.983	0.979	0.981	0.957
Relaying Game 1	0.883	0.798	0.497	0.510
Relaying Game 2	0.837	0.694	0.495	0.248
Relaying Game 3	0.815	0.686	0.491	0.205
Relaying Game 5	0.814	0.688	0.432	0.194
Relaying Game 10	0.804	0.691	0.443	0.202

B Relayer

B.1 Relayer loss

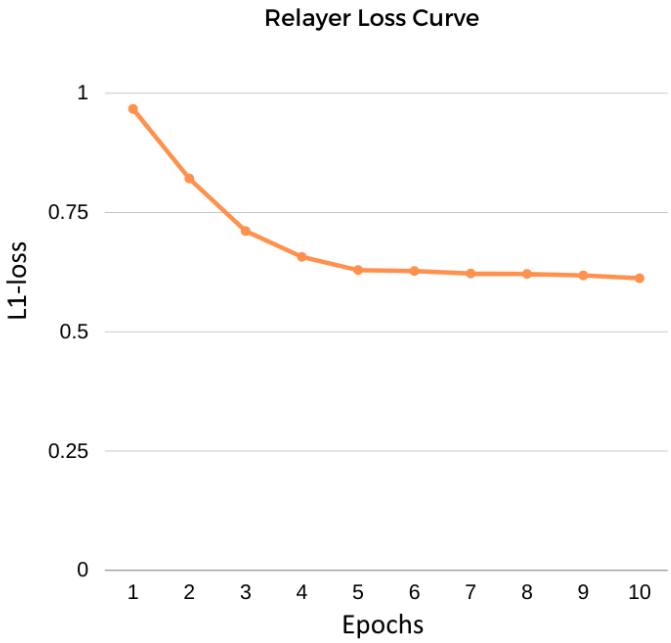

Figure 2: Embed-to-Embed Relayer L1-loss (same for all game sizes): 0.6183.

B.2 Relayer Language Statistics

In this section we report various language statistics for messages passed through various amounts of relayers. **|UM|** refers to the total amount of unique messages, **|UT|** refers to the total amount of unique tokens, μ **msg length** refers to the average message length, σ **msg length** refers to the standard deviation of the message length, μ **unique tokens** refers to the average amount of unique tokens in a message and σ **unique tokens** refers to the standard deviation of the amount of unique tokens in a message.

	|UM|	**|UT|**	μ **msg length**	σ **msg length**	μ **unique tokens**	σ **unique tokens**
1 Relayer	8225	37	8.015	2.99	5.203	1.946
2 Relayers	7615	35	8.098	2.912	5.163	1.885
3 Relayers	6957	35	8.12	2.901	5.05	1.827
5 Relayers	5990	35	8.122	2.891	4.906	1.755
10 Relayers	4780	35	8.113	2.902	4.792	1.713

B.3 Relayer Grammar

In this section we report the results of metrics used by (van der Wal et al., 2020) for grammar induction using CCL-BMM. Unfortunately due to computational restrictions we only induced grammar using 10 percent of our available messages.

Name	log prior	terminals	preterminals	recursives
Random	1215.32	44	8	1
1 Relayer	59787.93	35	30	6
2 Relayers	58510.83	33	30	2
3 Relayers	54164.99	31	27	8
5 Relayers	59436.63	34	27	3
10 Relayers	1927.67	32	2	10
	$\frac{terminals}{preterminals}$	induct μ log2likelihood	eval μ log likelihood	induct coverage
Random	5.5	-30.60	-30.60	100
1 Relayer	1.17	-15.40	-15.40	100
2 Relayers	1.10	-16.02	-16.02	100
3 Relayers	1.15	-14.71	-14.71	100
5 Relayers	1.26	-15.02	-15.02	100
10 Relayers	16.00	-50.20	-50.20	100
	eval coverage	num nominals	num preterminal groups	μ pre-terminal group by nominal
Random	100	6	24	4.67
1 Relayer	100	47	964	20.57
2 Relayers	100	50	910	18.26
3 Relayers	100	68	803	11.84
5 Relayers	100	80	835	10.48
10 Relayers	100	20	20	2.30

C Relaying Game Language Statistics

In this section we report various language statistics for the messages sent to the final receiver of various models for Game Size 2, 3, 5 and 10. **|UM|** refers to the total amount of unique messages, **|UT|** refers to the total amount of unique tokens, μ **msg length** refers to the average message length, σ **msg length** refers to the standard deviation of the message length, μ **unique tokens** refers to the average amount of unique tokens in a message and σ **unique tokens** refers to the standard deviation of the amount of unique tokens in a message.

C.1 Game Size 2

	|UM|	**|UT|**	μ **msg length**	σ **msg length**	μ **unique tokens**	σ **unique tokens**
Random	2104	56	4.293	3.797	2.168	1.293
Baseline	7343	38	8.844	2.764	3.576	1.426
Relaying Game 1	17	12	8.561	3.509	1.095	0.611
Relaying Game 2	6	4	5.898	4.918	0.589	0.492
Relaying Game 3	5	4	4.910	4.924	0.565	0.495
Relaying Game 5	5	4	5.653	4.957	0.565	0.495
Relaying Game 10	5	5	2.992	4.279	0.822	0.812

C.2 Game Size 3

	|UM|	**|UT|**	μ **msg length**	σ **msg length**	μ **unique tokens**	σ **unique tokens**
Random	2104	56	4.293	3.797	2.16	1.293
Baseline	5772	38	6.582	3.738	3.664	1.695
Relaying Game 1	15	8	8.561	3.509	1.171	0.655
Relaying Game 2	6	7	6.398	4.800	1.388	1.306
Relaying Game 3	4	6	6.398	4.800	1.775	1.779
Relaying Game 5	5	5	6.398	4.800	1.018	0.859
Relaying Game 10	6	5	1.868	3.265	0.639	0.480

C.3 Game Size 5

	|UM|	**|UT|**	μ **msg length**	σ **msg length**	μ **unique tokens**	σ **unique tokens**
Random	2104	56	4.293	3.797	2.16	1.293
Baseline	8421	45	7.895	2.679	4.969	1.531
Relaying Game 1	8	6	9.250	2.545	0.967	0.197
Relaying Game 2	8	8	5.862	4.530	1.817	1.043
Relaying Game 3	7	5	2.092	3.596	0.615	0.5709
Relaying Game 5	5	4	9.441	2.297	0.944	0.229
Relaying Game 10	7	7	4.088	3.669	1.627	0.587

C.4 Game Size 10

	|UM|	**|UT|**	μ **msg length**	σ **msg length**	μ **unique tokens**	σ **unique tokens**
Random	2104	56	4.293	3.797	2.16	1.293
Baseline	9143	39	9.333	1.926	5.224	1.559
Relaying Game 1	18	9	7.69	4.083	0.904	0.399
Relaying Game 2	4	3	5.395	4.984	0.539	0.498
Relaying Game 3	3	3	4.686	4.990	0.925	0.991
Relaying Game 5	3	4	4.575	4.975	0.919	0.996
Relaying Game 10	3	3	4.597	4.983	0.916	0.994

D Relaying Game Grammar Evaluation

In this section we report the results of metrics used by (van der Wal et al., 2020) for grammar induction using CCL-BMM. Unfortunately due to computational restrictions we only induced grammar using 10 percent of our available messages. For the same reason, we also only performed this operation on our languages which emergent during a game size of 2.

D.1 Game Size 2

Name	log prior	terminals	preterminals	recursives
Random	1215.32	44	8	1
Baseline	51753.67	38	25	58
Relaying Game 1	324.15	6	6	0
Relaying Game 2	56.45	3	3	0
Relaying Game 3	25.77	2	2	0
Relaying Game 5	25.77	2	2	0
Relaying Game 10	109.24	2	2	0
	$\frac{terminals}{preterminals}$	induct μ log2likelihood	eval μ log likelihood	induct coverage
Random	5.5	-30.60	-30.60	100
Baseline	1.52	-44.72	-44.72	99.60
Relaying Game 1	1	-1.53	-1.53	29.26
Relaying Game 2	1	0.00	0.00	57.20
Relaying Game 3	1	0.00	0.00	58.43
Relaying Game 5	1	0.00	0.00	58.16
Relaying Game 10	1	0.00	0.00	94.31
	eval coverage	num nominals	num preterminal groups	μ pre-terminal group by nominal
Random	100	6	24	4.67
Baseline	99.60	230	336	2.09
Relaying Game 1	29.26	2	5	2.5
Relaying Game 2	57.20	1	1	1
Relaying Game 3	58.43	1	1	1
Relaying Game 5	58.16	1	1	1
Relaying Game 10	94.31	1	1	1

E Significance Testing

E.1 Relayer Languages

In this section we will report various p-values which resulted from a two sample t-test using the Relayer Languages, so as to say the messages after they have been passed through n Relayers. The left number in a cell is the p-value calculated using distributions of message lengths and the right number in a cell is the p-value calculated using distributions of unique tokens. We considered a p-value of $p < 0.05$ an indicator of significance.

	10 Relayers	5 Relayers	3 Relayers	2 Relayers
1 Relayer	0.019 - <0.001	0.010 - <0.001	0.012 - <0.001	0.047 - 0.14
2 Relayers	0.715 - <0.001	0.593 - <0.001	0.565 - <0.001	
3 Relayers	0.865 - <0.001	0.962 - <0.001		
5 Relayers	0.826 - <0.001			

E.2 Relaying Game Languages

In this section we will report various p-values which resulted from a two sample t-test using the Relaying Game Languages, so as to say the messages that are sent to the final receiver in the emergent language reference game. We also compare the languages produced from the Relaying Game with the Baseline and Random Sender-Receiver agents. The left number in a cell is the p-value calculated using distributions of message lengths and the right number in a cell is the p-value calculated using distributions of unique tokens. We considered a p-value of $p < 0.05$ an indicator of significance.

E.2.1 Game Size 2

	Relaying Game 10	Relaying Game 5	Relaying Game 3	Relaying Game 2	Relaying Game 1	Baseline
Random	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001
Baseline	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	
Relaying Game 1	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001		
Relaying Game 2	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001			
Relaying Game 3	<0.001 - <0.001	<0.001 - 1				
Relaying Game 5	<0.001 - <0.001					

E.2.2 Game Size 3

	Relaying Game 10	Relaying Game 5	Relaying Game 3	Relaying Game 2	Relaying Game 1	Baseline
Random	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001
Baseline	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	
Relaying Game 1	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001		
Relaying Game 2	<0.001 - <0.001	1 - <0.001	1 - <0.001			
Relaying Game 3	<0.001 - <0.001	1 - <0.001				
Relaying Game 5	<0.001 - <0.001					

E.2.3 Game Size 5

	Relaying Game 10	Relaying Game 5	Relaying Game 3	Relaying Game 2	Relaying Game 1	Baseline
Random	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001
Baseline	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	
Relaying Game 1	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001		
Relaying Game 2	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001			
Relaying Game 3	<0.001 - <0.001	<0.001 - <0.001				
Relaying Game 5	<0.001 - <0.001					

E.2.4 Game Size 10

	Relaying Game 10	Relaying Game 5	Relaying Game 3	Relaying Game 2	Relaying Game 1	Baseline
Random	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001
Baseline	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	
Relaying Game 1	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001		
Relaying Game 2	<0.001 - <0.001	<0.001 - <0.001	<0.001 - <0.001			
Relaying Game 3	0.207 - 0.521	0.115 - 0.669				
Relaying Game 5	0.755 - 0.831					