# Giving Agents a Voice: Emergence of Language in Auditory Multi-Agent Games

Juri Morisse (s3737764)

December 19, 2022

**Abstract**

Language is a key part of intelligence. To better understand its emergence, use, and development language formation can be simulated in multi-agent games. Previous work has explored many such setups but communication between agents was always implemented as exchange of symbolic sequences. Considering that natural languages emerged not as textual but as gestural and auditory communication, I argue this to be a severe limitation. I propose an extension to an existing framework for studies on language emergence in multi-agent games that allows agents to communicate via audio signals. Performing an initial experiment, I find agents to be benefiting from auditory communication. Agents learn faster, achieve higher success in playing their game, and do so more robustly across several random seeds. Considering these results, I encourage further work that investigates language emergence in auditory multi-agent games.

## 1 Introduction

The way we communicate is often argued to be the biggest achievement of human cognitive development. It allows humans not only to coordinate and cooperate but also to exchange immaterial concepts like morality and religion. Consequently, understanding the development and use of human language is of great interest to researchers that try to understand human intelligence or try to develop artificial intelligence resembling it. However, studying how language emerges from scratch is practically impossible and studying language development across generations requires studies that span over centuries. A field of study that tries to circumvent these restrictions simulates language emergence between artificial agents that need to communicate in order to successfully play a game. Insights gained in these studies of artificial language emergence can potentially be transferred to the domain of human languages or can be used to improve communication between artificial agents. In recent years, there has been an increase in studies aiming to do so. However, all of these studies let agents communicate either via a fixed length vector or symbol sequences.

Considering that human symbolic language developed much later than gestural and auditory communication, this significantly limits the comparability between language emergence in these artificial setups and they way human language emerged.

In this study, I propose an extension to a well established framework for language emergence studies that allows the implementation of multi-agent communicative games in which the communication between agents consists of audio sequences. Using such a setup for auditory communication, I show that agents can play a discriminative game more successfully as they were able to with symbolic communication. Performance is also more stable across random seeds than it is when agents use symbol sequences.

# 2 Related Work

The first studies simulating language emergence in multi-agent settings were conducted already around 30 years ago [1]. These first works followed a number of different approaches including the use of finite-state machines [1] and evolutionary optimization of RNN-based agents [2]. The research field experienced a shift after a study proposed to use multi-agent reinforcement learning without requiring pre-defined communication protocols [3]. Today, the use of multi-agent reinforcement learning is the dominant approach in language emergence studies.

Using games as an environment in which agents develop communication has been a popular approach already in early studies [4] [5]. The reason for this is two-fold. The game success can be used as an indirect measure of the quality of communication learned by the agents. In addition, it is theoretically founded in the popular view that the meaning of a language is founded in its use, first put forward by Ludwig Wittgenstein [6].

Following the breakthrough of deep learning, studying language emergence in multi-agent games experienced renewed interest. Many studies were performed demonstrating how deep learning allows agents to learn communication in more complex settings and in ways that more closely resemble natural language [7] [8] [9] [10].

Since then, studies have focused on investigating the emergent languages [11], [12], the factors influencing them [13], transmission of language between generations of agents ( [14], [15]), and effects of social structure in agent communities ( [16]). However, no previous study has forced agents to communicate with auditory messages. Like many studies (e.g. [7]) provided proof, that deep neural agents can successfully learn to play cooperative communication games by communicating with symbol sequences, I aim to show that they are also able to do so with the use of audio sequences.

# 3 Methods

Given that no previous study has implemented multi-agent communication learning with the use of audio signals, I propose a new method to do so. In the following, I outline how I extend an existing framework for symbol sequence multi-agent communication learning to allow the exchange of audio signals between agents. Afterwards, I explain the game played by the agents. Following that, I describe the architecture used for implementing my agents.

## 3.1 Framework

Motivated by the increase in popularity of studies investigating language emergence in multi-agent games, a team from Facebook's research department developed a toolkit for implementing such studies called Emergence of lanGuage in Games (EGG) [17] [1]. This toolkit lowers the entry barrier for researches as it provides pre-implemented building blocks that can be used to design and evaluate new setups quickly. To design a new experiment, one has to define only a few components: the data, the agents' core architecture, and the loss function. Given these components, one can implement a complete study setup by utilizing pre-implemented functions provided in the EGG toolkit. The data component needs to be a PyTorch compatible dataset that has at least three elements per sample: the sender input, the target, and the receiver input. The agent's core architectures must conform to the EGG logic. That is, the sender's core architecture must must map the sender input to an embedding that can be used as an input to a recurrent neural network (RNN). The RNN does not have to be implemented from scratch. Instead, EGG provides a wrapper function that can be applied to a sender's core architecture. The output of this wrapper function is then the message produced by a sender. The receiver works similarly. A wrapper function implements a RNN that processes the received message and provides an embedding of the message that is further processed by the receiver's core architecture. The core architecture's output is then the output of the receiver. The last thing one has to implement is the loss function that computes a loss score that can be affected by many factors. However, one can also reduce the number of factors and compute the loss score only based on the target and the receiver's output. Given on has defined the dataset, the agents' core architectures and the loss function, the rest can be implemented using functions provided in EGG. With these functions, one can embed agents in a number of different games, use different RNN architectures, use different optimization techniques and customize the setup further. For a detailed description of the EGG toolkit, I refer the reader to [17].

To extend the EGG toolkit to allow communication via audio signals, I cloned existing EGG functions and adjusted their functionality. In particular, I extended an existing wrapper function for sender agents. My extended version still works in the same, i.e. it generates a sequence of integers that serve as

[1] `https://github.com/facebookresearch/EGG`

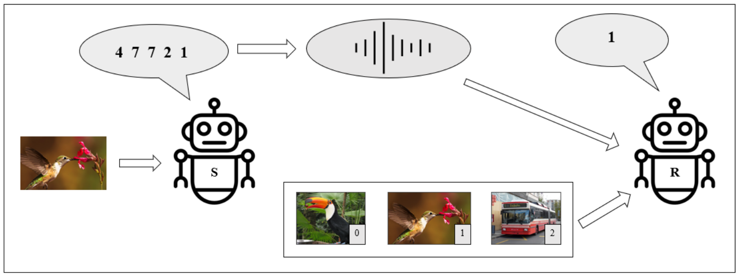

Figure 1: Visualization of a successful game run. The sender (left) receives an image, produces a symbolic message, and encodes this message as an audio signal. The receiver gets this audio signal with a set of images and needs to decide which one was described by the sender.

a message. However, my extended function receives and additional argument that contains audio signals, one for each possible symbol. Using these audio signals, I transform the symbol sequence to an audio sequence. This is done by one-hot encoding the integers and performing matrix multiplication with the tensor containing the audio signals. The result is a tensor where an audio signal has replaced each of the integers. Only this audio signal is then transmitted to the receiver. The receiver embeds the audio signals to a representation that can be sequentially processed by a RNN. From this point onward, the procedure remains unchanged compared to the symbol sequence approach.

## 3.2 Game

I let the agents play a discrimination game which is a common choice in modern studies [7] [13] [15]. It is played by two agents, the sender and the receiver. The sender gets an image and describes the image in a message. This message is then sent to the receiver who along with it gets a set of images. The receiver then has to determine based on the message it received, which of the images was described by the receiver. The game was formalized first by [7] and consists of three steps:

1. A target image $t$ and a set of $K$ distractor images $k_{k=1}^{K}$ are sampled from a set of $N$ images.
2. The sender receives $t$ and produces a message $m_t$ that contains symbols from vocabulary $V$ of size $|V|$. Messages can maximally be of length $L$.
3. The receiver gets the message $m_t$ along with a randomly ordered set of images that contains the target and distractor images. The receiver predicts which image was described by the sender.

My setup changes the third step such that the receiver does not get message $m_t$ but an audio signal based on $m_t$. Further, in my setup, $V$ consists of the integers between 0 and 9 with 0 serving as a stop symbol. Therefore, $|V|$ is nine. The number of images fed in to the receiver is set to 3, i.e. it receives the target image and two distractor images. Figure 1 shows how an interaction between sender and receiver in this type of game looks like.

### 3.3 Agent Architectures

As the focus of my paper is on providing setup in which agents can learn successfully play a cooperative game through the use of communication via audio signals, I keep the agent's core architecture simple. The sender's core architecture consists only of a single fully connected layer. This layer maps the image embedding (see Section 4.1.1) to an initial hidden state for its RNN implemented in the wrapper function. The sender's RNN is a GRU with a hidden state size of 10. Before the hidden state produced by the sender's core architecture is passed to the RNN, a tanh activation function is applied. The receiver gets two inputs: the last hidden state of its wrapper RNN (a GRU with a hidden state of size 35) that processes the received message and the set of target image and distractor images (see Section 3.2). To decide which of the received images is the one described by the sender, it embeds the set of target and distractor images to the same dimension as the message embedding it receives from its wrapper RNN. The output of the receiver is then a list of dot products between the message embedding and the embeddings of the target and distractor image. A softmax is applied to the dot products and its outputs are the receivers predictions on which of the received images is the target image.

## 4 Experiments

To evaluate how agents learn to communicate using audio messages compared to symbolic messages, I train agents in both settings. In the following, I provide details about the dataset used and the training procedure. To allow for inspection and reproduction of my study, the code and all results are publicly available online [2].

### 4.1 Data

In this section, I describe the datasets used. One dataset contains the images the agents communicate about and the other contains audio files used to encode symbolic messages into audio messages.

#### 4.1.1 Image Data

The images used in my setup are from the ILSVRC2012 dataset [18]. ILSVRC2012 contains over 1.2 million high resolution images that each belong to one of 1,000

[2] https://github.com/jumorisse/Giving_Agents_a_Voice-Emergence_of_Language_in_Auditory_Multi-Agent_Games

classes. It is a popular dataset to benchmark image classification systems. I decided to use ILSVRC2012 for two main reasons. First, it is more complex than comparable datasets like CIFAR10 and CIFAR100 and, thus, poses a greater challenge to the agents. Showing that agents can successfully play using images from ILSVRC2012 would most likely mean that they would also be able to do so with images from other datasets. The second reason is that I do not feed the images to the agents directly. Instead, I use a pre-trained ResNet18 [19] model to embed the images and feed in the embeddings instead of the raw images. This technique reduces the complexity of my agents and the time needed for training. The embedding of an image is the activation in the second to last layer of the ResNet18 which results in embeddings of size 512. Since ResNet18 was designed to classify ImageNet images, it is well suited for embedding the ILSVRC2012 dataset.

#### 4.1.2 Audio Data

Since the symbolic messages are composed of integers between 0 and 9, I decided to use recordings of these numbers to produce semantically equal audio encodings of the symbol sequences. To do so, I use a free spoken digits dataset (FSDD) obtained from GitHub [3]. It contains over 3,000 recordings of 6 different speakers for each digit between 0 and 9 in English. I extracted one recording for each digit between 0 and 9 from the same speaker to construct the audio dataset that is passed to the sender in order for it to encode the symbolic messages as audio sequences as described in 3.1. To have a set of equally long audio sequences, I padded all sequences with zeros until they met the length of the longest sequence. Further, I zeroed the values of the sequence representing the integer zero. This is due to the fact that in my implementation, integer zero acts as a stop symbol for the message producing and processing RNNs and, thus, should not contain any information.

### 4.2 Training and Hyperparameter

Given the receiver's prediction of the target image position and its actual position, a cross entropy loss is computed. Given this loss, an EGG provided trainer optimizes the agents' following the REINFORCE algorithm [20] and using the ADAM optimizer. Training is performed with batches of 128 samples and a learning rate of 0.0001. Agents are trained for 200 epochs. After each training, the agents performance on the validation data is recorded. To account for random effects, I repeat training for five different random seeds.

## 5 Results

The results obtained from the experiments described in the previous section clearly show that agents are able to learn to communicate successfully in both

[3] `https://github.com/Jakobovski/free-spoken-digit-dataset`

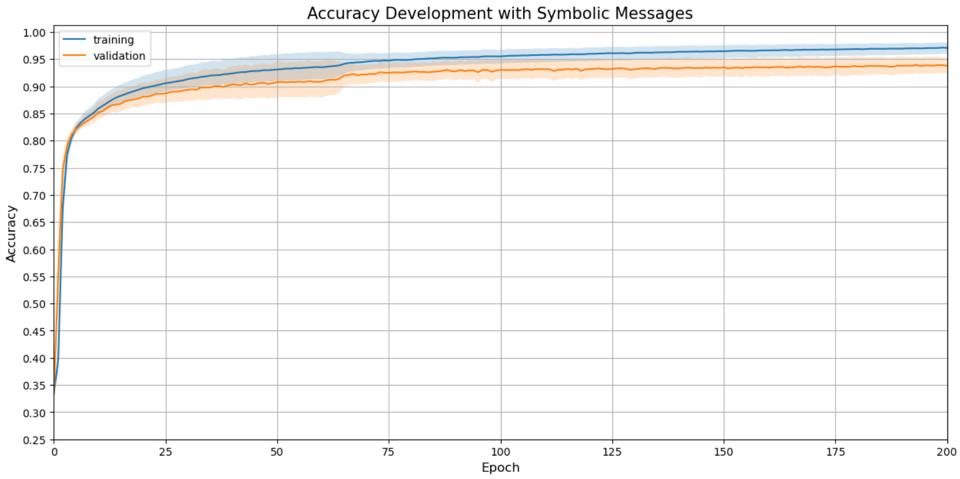

Figure 2: Development of training and validation accuracy in the symbol sequence setup. Curves are the averages across five random seeds with the confidence intervals representing standard deviations across the random seeds.

| Setup | Test Accuracy | Test Loss |
|---|---|---|
| Symbol Messages | 93.64% ± 1.59% | 0.618 ± 0.015 |
| Audio Messages | 97.31% ± 0.4% | 0.582 ± 0.004 |

Table 1: Comparison of both setups' performance on the test dataset. Reported metrics are averages across random seeds with their standard deviations across random seeds.

setups. In fact, agents perform better when communicating via audio messages instead of symbolic messages. Using audio messages, they reach higher accuracies on the train, validation and test set. In addition, using audio messages lets agents make faster progress. Training is also much more stable across random seeds when agents use audio messages: the standard deviation in Figure 3 is barely visible. This is confirmed by the performance on the test dataset (see Table 1). The standard deviation across random seeds is reduced by about more than two thirds for agents using audio messages compared to those using symbol messages.

# 6 Conclusion

In this paper, I proposed a method to extend previous works on language emergence in multi-agent games to the audio domain. To do so, I added new functionalities to the established toolkit EGG. These additions allow the implementation of a sender agent who produces a symbolic message like in previous approaches but encodes this symbolic message using given audio data into an audio message.

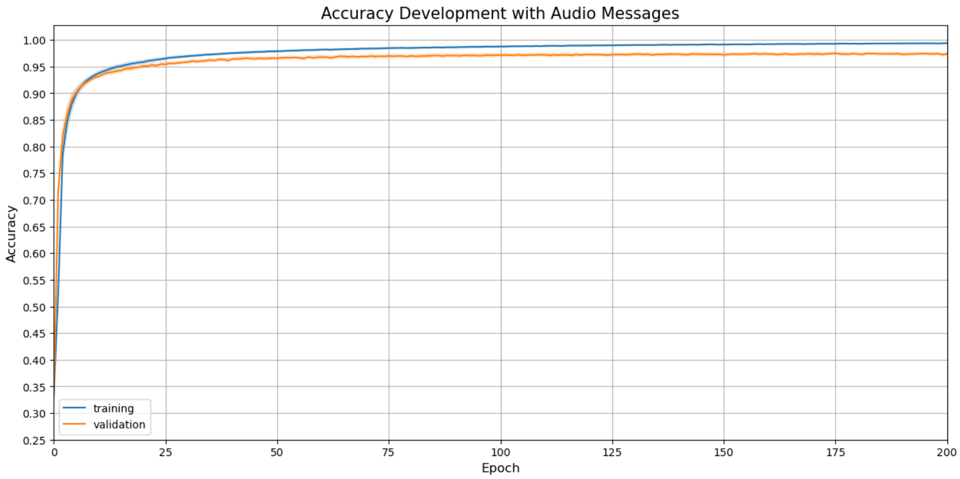

Figure 3: Development of training and validation accuracy in the audio sequence setup. Curves are the averages across five random seeds with the confidence intervals representing standard deviations across the random seeds.

Further, my extensions allow a setup in which the receiver only receives this audio message for discriminating a target image. Comparing the performance of agents in a setup using symbolic messages to one using audio messages, I found agents that communicate via audio messages to be more successful and learn faster. Further, performance was more stable across random seeds compared to the setup with symbol messages. These results are surprising and encouraging at the same time. It might be that the performance gains are rooted solely in the larger dimensionality of the messages transmitted. However, this larger dimensionality does not cause increased overfitting and also does not increase the overall message space. Since the sender still produces symbol based messages before encoding them as audio signals, the number of distinct audio signals is equally large to that of distinct symbol sequences. It requires further analysis to determine what exactly causes the advantage of audio based communication over test based communication. Another interesting avenue for future work would be to investigate the effect of non-deterministic encoding of symbol to audio sequences by sampling one of many audio sequences that resemble the same symbol. This would be closer to natural languages where speakers sound differently when pronouncing the same word.

For now, I have shown that multi-agent communication learning is possible using audio sequences instead of symbol sequences.

# References

[1] Bruce J. MacLennan and Gordon M. Burghardt. Synthetic ethology and the evolution of cooperative communication. *Adapt. Behav.*, 2(2):161–188, 1993.

[2] John Batali. Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax. In *Artificial life IV*, pages 160–171. Cambridge, MA, 1994.

[3] Tatsuya Kasai, Hiroshi Tenmoto, and Akimoto Kamiya. Learning of communication codes in multi-agent reinforcement learning problem. In *2008 IEEE conference on soft computing in industrial applications*, pages 1–6. IEEE, 2008.

[4] C. Lee Giles and Kam-Chuen Jim. Learning communication for multi-agent systems. In Walt Truszkowski, Christopher A. Rouff, and Michael G. Hinchey, editors, *Innovative Concepts for Agent-Based Systems, First International Workshop on Radical Agent Concepts, WRAC 2002, McLean, VA, USA, January 16-18, 2002, Revised Papers*, volume 2564 of *Lecture Notes in Computer Science*, pages 377–392. Springer, 2002.

[5] Luc Steels. Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. *Evolution of Human Language. Edinburgh University Press, Edinburgh*, 1997.

[6] Ludwig Wittgenstein. *Philosophical Investigations.* Basil Blackwell, Oxford, 1953.

[7] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols, 2017.

[8] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language, 2016.

[9] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input, 2018.

[10] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation, 2016.

[11] Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2962–2967. Association for Computational Linguistics, 2017.

[12] Bence Keresztury and Elia Bruni. Compositional properties of emergent languages in deep learning. *CoRR*, abs/2001.08618, 2020.

[13] Diana Rodríguez Luna, Edoardo Maria Ponti, Dieuwke Hupkes, and Elia Bruni. Internal and external pressures on language emergence: Least effort, object constancy and frequency. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4428–4437. Association for Computational Linguistics, 2020.

[14] Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. Emergence of compositional language with deep generational transmission. *CoRR*, abs/1904.09067, 2019.

[15] Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. Co-evolution of language and agents in referential games. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2993–3004. Association for Computational Linguistics, 2021.

[16] Laura Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3698–3708. Association for Computational Linguistics, 2019.

[17] Eugene Kharitonov, Roberto Dessì, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. EGG: a toolkit for research on Emergence of lanGuage in Games. `https://github.com/facebookresearch/EGG`, 2021.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. 2014. cite arxiv:1409.0575Comment: 43 pages, 16 figures. v3 includes additional comparisons with PASCAL VOC (per-category comparisons in Table 3, distribution of localization difficulty in Fig 16), a list of queries used for obtaining object detection images (Appendix C), and some additional references.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[20] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992.