Emergence of Numeric Concepts in Multi-Agent Autonomous Communication

Shangmin Guo

Master of Science
Data Science
School of Informatics
University of Edinburgh
2019

Abstract

Natural language understanding is a long-standing topic in artificial intelligence. With the rapid development of deep learning, most of current state-of-the-art techniques in natural langauge processing are based on deep learning models trained with large-scaled static textual corpora. However, we human beings learn and understand in a different way. Thus, grounded language learning argues that models need to learn and understand language by the experience and perceptions obtained by interacting with enviroments, like how humans do.

With the help of deep reinforcement learning techniques, there are already lots of works focusing on facilitating the emergence of communication protocols that have compositionalities like natural languages among computational agents population. Unlike these works, we, on the other hand, focus on the numeric concepts which correspond to abstractions in cognition and function words in natural language.

Based on a specifically designed language game, we verify that computational agents are capable of transmitting numeric concepts during autonomous communication, and the emergent communication protocols can reflect the underlying structure of meaning space. Although their encodeing method is not compositional as natural languages from a perspective of human, the emergent languages can be generalised to unseen inputs and, more importantly, are easier for models to learn. Besides, iterated learning can help further improving the compositionality of the emergent languages, under the measurement of topological similarity. Furthermore, we experiment another representation method, i.e. directly encode numerals into concatenations of one-hot vectors, and find that the emergent languages would become compositional like human natural languages. Thus, we argue that there are 2 important factors for the emergence of compositional languages: i) input feature representations are inherently disentangled; ii) effective methods to amplify compositional inductive bias, e.g. iterated learning.

Acknowledgements

Throughout the writing of this dissertation, I have received a great deal of support and assistance from many people.

First of all, I would like to express my deepest appreciation to my supervisors, Dr. Ivan Titov and Prof. Kenny Smith, for their support for the proposal of this project and also their patient guidance, encouragement and advices. I am extremely lucky to have supervisors who cared so much about this work and proposed so much insights about the research topic.

I would also like to extend my deepest gratitude to my tutors, Mr. Serhii Havrylov and Dr. Stella Frank, whose expertise were invaluable in the formulating of the research topic and methodologies in particular. Without their insights and sagacities, completing this work would be much more difficult.

In particular, I greatly appreciate the support received through the collaborative work undertaken with Joshua Ren for the iterated learning part in this work as well as his insights for discussions about explaining the phenomena in experiments.

I am deeply indebted to all my friends in Edinburgh who opened their homes to me during my time at University of Edinburgh and who were always so helpful in numerous ways. Special thanks to Yan Yang, Yiyun Jin, Ruochun Jin, Muyang Liu, Wenbin Hu, Yuanhao Li and Jie Zhou.

I am extremely grateful to my beloved girlfriend, Siting Lu. Thank you for supporting my for everything and especially I cannot thank you enough for being with me throughout this experience. Thank you for all your companionship and care to me.

I cannot begin to express my heartfelt thank you to my Mum and Dad for always believing in me and encouraging me to follow my dreams as well as for helping in whatever way they could during this challenging period.

Table of Contents

Chapter 1

Introduction

Natural language processing (NLP) is an important and long-standing topic in artificial intelligence (AI), in which a core question is natural language understanding (NLU). With the rapid development of deep learning (DL), most current state-of-the-art methods in NLP, e.g. [Socher et al., 2013, Mikolov et al., 2013, Kim, 2014], are based on DL models trained on massive static textual corpora. From an information processing perspective, the information flow of NLP-based human-computer interaction systems is illustrated in Figure 1.1 given as follow. As the diagram shows, the input of NLP systems are various kinds of textual materials generated by human beings to describe their experiences and perceptions. Under such a perspective, symbols in natural languages are actually feature representations of the original experiences and perceptions, whereas most current NLP systems directly take these symbols as original features.

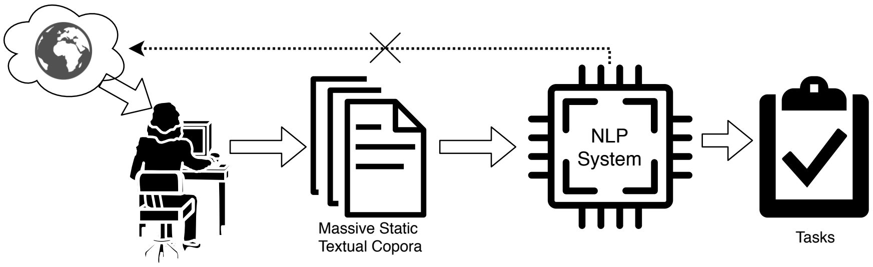


Figure 1.1: An overview of information flow in current NLP systems.

Considering the missing original experiences and perceptions, grounded language learning (GLL) argues that models need a grounded environment to learn and understand language[Matuszek, 2018]. However, natural languages of the time have been developed for at least tens of thousands of years[Berwick and Chomsky, 2016] and already became very sophisticated. Thus, to verify that computational agents can truly ground symbols in natural languages to corresponding experiences and perceptions, as well as be able to complete the specified tasks, it is necessary to help them to discover and develop various kinds of characteristics of natural language during autonomous communication of agents. There are already lots of works, e.g. [Batali, 1998, Christiansen and Kirby, 2003, Smith et al., 2003a, Hill et al., 2017, Havrylov and Titov, 2017, Yu et al., 2018, Kottur et al., 2017], aiming to facilitate the emergence of "natural language" in multi-agent autonomous communication systems. However, one significant limitation of previous works is that, only referential objects/attributes in environments, e.g. shapes and colours, were considered and to which discrete symbols were grounded.

This project, on the other hand, aims to explore and analyse the grounding of abstractions which are non-referential[1] in the original experiences and perceptions of human beings. However, as it is too huge a topic to tackle, our project is limited to numeric concepts which correspond to cardinal numerals in natural languages for the following reasons: i) numeral systems are relatively simple and self-contained[Hurford, 1999]; ii) concepts related to cardinal numerals are more straightforward to model with numeric representations; iii) functions of emergent cardinal numerals can be formalised and verified more reliably in simulation.

In this work, our main contributions are given as follows:

1. We propose a language game in which we can verify whether computational agents can communicate numeric concepts with each other, and successfully train agents to "invent" communication protocols that can autonomously transmit numeric concepts.

2. We further analyse and discuss the structure of the emergent communication protocols, and improve the compositionality by transforming iterated leaning proposed by [Smith et al., 2003b] to train the DL models.

3. We compare learning speeds of various kinds of languages as well as different representations, and propose an alternative hypothesis for explaining the emergence of words with different types and functions.

[1] By non-referential, we mean that "there is no concrete entity in the world (real or virtual) can be referred as".

Chapter 2

Background

There are 2 almost disjointly developed research topics that motivates this project, i.e. computer simulation methods in evolutionary linguistics and multi-agent games in grounded language learning. Thus, in the following 2 sections, we will introduce works which are highly related to our project from these 2 different areas.

2.1 Computer Simulation Methods in Evolutionary Linguistics

One important issue in the field of evolutionary linguistics is to use quantitative methods to overcome the time limit on unpreserved pre-historic linguistic behaviours [Lieberman, 2006, Evans and Levinson, 2009]. Since it was first introduced by [Hurford, 1989], computer simulation method has attracted a rapidly growing attention, e.g. [Hurford et al., 1998, Knight et al., 2000, Briscoe, 2002, Christiansen and Kirby, 2003, Bickerton and Szathmáry, 2009, Cangelosi and Parisi, 2012]. As introduced in Chapter 1, one of our objectives is to facilitate the discovery and development of various kinds of natural language phenomena of computational agents, which shares a same objective and motivation of computer simulation methods in evolutionary linguistics.

To imply and verify a linguistic theory, there are 2 necessary component: i) environments, in which agents can execute actions and communicate with each other; ii) pre-defined elementary linguistic knowledge that can be manipulated and altered by agents. Further, we could categorise the environments into the following 3 different types according to their simulation objectives:

- *Iterated learning* introduced by [Kirby, 1999] which aims at simulating cultural transitions from generation to generation.
- *Language games* introduced by [Wittgenstein, 1953] which takes the emergent communication protocols in cooperation between individuals as a prototype of language.
- *Genetic evolution* introduced by [Briscoe, 1998] which aims at simulating evolution of languages as a kind of natural selection procedure[Darwin, 1859].

With environments and pre-defined elementary linguistic knowledge, computational agents can then learn bi-directional meaningutterance mapping functions[Gong and Shuai, 2013]. With different kinds of resulting linguistic phenomena, this simulation procedure can be broadly categorised into 2 classes:

- lexical models, e.g. [Steels, 2005, Baronchelli et al., 2006, Puglisi et al., 2008], whose main concern is whether a common lexicon can form during the communication in agent community;
- syntactic and grammatical models, e.g. [Kirby, 1999, Vogt, 2005], in which agents mainly aim to map meanings (represented in various ways) to utterances (either structured or unstructured).

However, no matter how these mapping functions are learnt, e.g. by neural network models [Munroe and Cangelosi, 2002] or by mathematical equations [Minett and Wang, 2008, Ke et al., 2008], the most basic elements of linguistics, e.g. meanings to communicate about and a signalling channel to employ, are all pre-defined.

In contract, although we also follow the framework of language games and train agents in an iterated learning fashion, the basic linguistics elements in our project are not provided from the outset any more and computational agents can specify meanings of symbols/utterances by themselves.

2.2 Multi-agent Games in Grounded Language Learning

Unlike how we human beings learn and understand language, the current DL-based NLP techniques learn semantics from only large-scaled static textual materials. Thus, grounded language learning argues that computational agents also need to learn and understand languages by interacting with environments and grounding language into their experience and perceptions. With the recent rapid development of deep reinforcement learning (DRL), it has been shown that computational agents can master a variety of complex cognitive activities, e.g. [Mnih et al., 2015, Silver et al., 2017]]. Thus, several papers in grounded language learning apply DRL techniques to allow agents to learn or invent natural languages[1], such as [Hermann et al., 2017, Mordatch and Abbeel, 2018, Havrylov and Titov, 2017, Hill et al., 2017].

To verify language abilities of computational agents, previous works in GLL usually follow the framework of language games, of which are mainly variants of referential games introduced by [Lewis, 1969], e.g. [Hermann et al., 2017, Havrylov and Titov, 2017]. Also, some works are more motivated by the competition instead of cooperation such as [Cao et al., 2018].

From another perspective, based on the number of participated agents, we can broadly categorise language games in GLL into the following 2 types:

- *Single-agent games* usually need to be done by one agent and a human particiator, in which the main concern is to explore how computational agents could learn the compositionality of semantics.

- *Multi-agent games* are usually completed by an agent population, in which the main concern is to explore how various kinds of natural language phenomena emerge and evolve during communicating among agents.

However, like we mentioned in Chapter 1, whichever kind of language game they follow in previous works of GLL, the objects/attributes the symbols grounded to are all referential. We, on the other hand, aim to explore the feasibility of grounding symbols to non-referential objects (specifically, numeric concepts) during the game.

[1] Strictly speaking, "invent natural language" should be called as "invent communication protocols sharing compositionality like natural languages". However, as our project is to facilitate compositionality in multi-agent communication protocols, we thus call these emergent communication protocols a kind of "language" invented by agents

Chapter 3

Game, Models and Measurements

In this chapter, we first describe the proposed language game and the definition of numerals in our game. We then introduce the architecture of models we used and also the iterated learning for training models.

3.1 Game Description

Unlike traditional simulation methods in evolutionary linguistics introduced in Section 2.1, there are 3 necessary components in our architecture and they are given as follows:

- *Environment*: To imply our linguistic assumption as well as make the size of environment limited and thus analysable, all perceptions in the established environment are sequences of objects represented by one-hot vectors. For ease of demonstration, we denote these objects as $o \in O$ where $O = \{A, B, C, \dots\}$ is the universal set of all kinds of objects in the following sections.

- *Agents*: There are 2 kinds of agents in our project: i) *speakers* S that can observe objects in the environment and emit messages m_i; ii) *listeners* L that can receive the messages and generate a sequence of objects.

- *Dynamics*: In this project, the dynamics mean not only the manually designed reward function for agents but also all elements related to training them, e.g. iterated learning and blank vocabulary. The details will be introduced in Subsection 3.2.3 and Subsection 3.2.4.

It worth mentioning that one premise of our project is that we do not have any assumption about the architecture of computational agents, and we focus more on the representations from environments as well as how agents are trained.

3.1.1 Game Procedure

The first proposed game is to let listeners reconstruct sets of objects based on the messages transmitted by speakers, thus we call it “Set-Reconstruct” game. The overall view of the proposed Set-Reconstruct game is illustrated in Figure 3.1 given as follow.

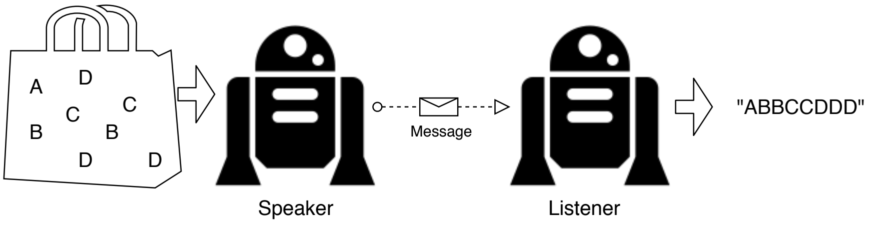


Figure 3.1: Diagram of Game Playing Procedure.

According to the steps of playing games at iteration i, the components of our games are illustrated as follows:

1. Perceptions: the perception from environments is a **set** of objects, i.e. $s_i = \{o_{i_1}, o_{i_2}, \dots\ o_{i_n}\}$ where n is the number of elements.

2. Speaker observation and message generation: after observing and encoding the perception, speaker S would generate a message $m_i = \{s_{i_1}, s_{i_2}, \dots, s_{|M|}\}$ where $|M|$ is the maximum length of messages and $s_k, k \in 1, \dots, |V|$ are selected from a randomly initialised vocabulary such that the symbols in the initially meaningless vocabulary whose size is $|V|$;

3. Listener receiving message and perception reconstruction: after receiving and encoding the message m_i, the listener would generate a **sequence** $\hat{s}_i = \{\hat{o}_{i_1}, \hat{o}_{i_2}, \dots\ \hat{o}_{i_n}\}$ whose symbols are identical to those in the original perception s_i;

4. Reward and parameter update: by comparing s_i and $\hat{s}_i$, we take the cross-entropy between them as the reward for both listener and speaker and update parameters of both speaker and listener with respect to it.[1]

One thing that needs to be highlighted is that the perceptions s_i are sets and thus order of objects would not make any difference. Further, we argue that the only important feature that need to be transmitted is actually the numbers of different objects which corresponds to the function of numerals in natural language.

[1] Different ways of updating parameters are introduced in Section 3.2.

3.1.2 Functions of Numerals in the Game

Broadly speaking, numerals are words that can describe the numerical quantities and usually act as determiners to specify the quantities of nouns, e.g. ”two dogs” and ”three people”. Also, under most scenarios, numerals correspond to non-referential concepts[Da Costa and Bond, 2016]. Considering the objective of listeners L in our language game, we define a numeral as a symbol s^n at **position** i indicating a function that reconstructs some object o_i exactly n times:

$$s^n : o_i \rightarrow \{ \overbrace{o_i, \ldots, o_i}^{n \text{ elements}} \} \tag{3.1}$$

Note that, the meaning of a symbol is not only decided by itself but also its position in message, as L would encode meanings of symbols according to their appearance in messages.

From the side of speakers S, a numeral is defined as a symbol s^n at **position** i that represents the numbers of specific object o_i, as we cannot whether agents realise the meanings of symbols are not related to their positions in the messages, without specifically designed model architecture. Thus, we expect S would first learn to count the number of different objects and then encode them into a sequence of discrete symbols. As [Siegelmann and Sontag, 1992] shows that Recurrent Neural Networks (RNNs) are Turing-complete and Long-short Term Memory (LSTM) model proposed by [Hochreiter and Schmidhuber, 1997] is a super set of RNN, it is safe to claim that LSTM is also Turing-complete and thus capable of counting numbers of objects.

3.1.3 A Variant: Set-Select Game

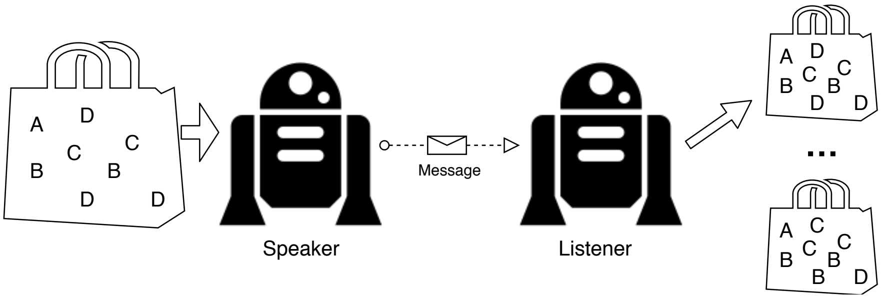


Figure 3.2: Diagram of Referential Game Playing Procedure.

We illustrate the Set-Select game, a referential variant of Set-Reconstruct game, in Figure 3.2 given above. The only difference is that listeners need to select the correct set of objects among several distractors [2] instead of reconstructing it.

3.2 Proposed Models

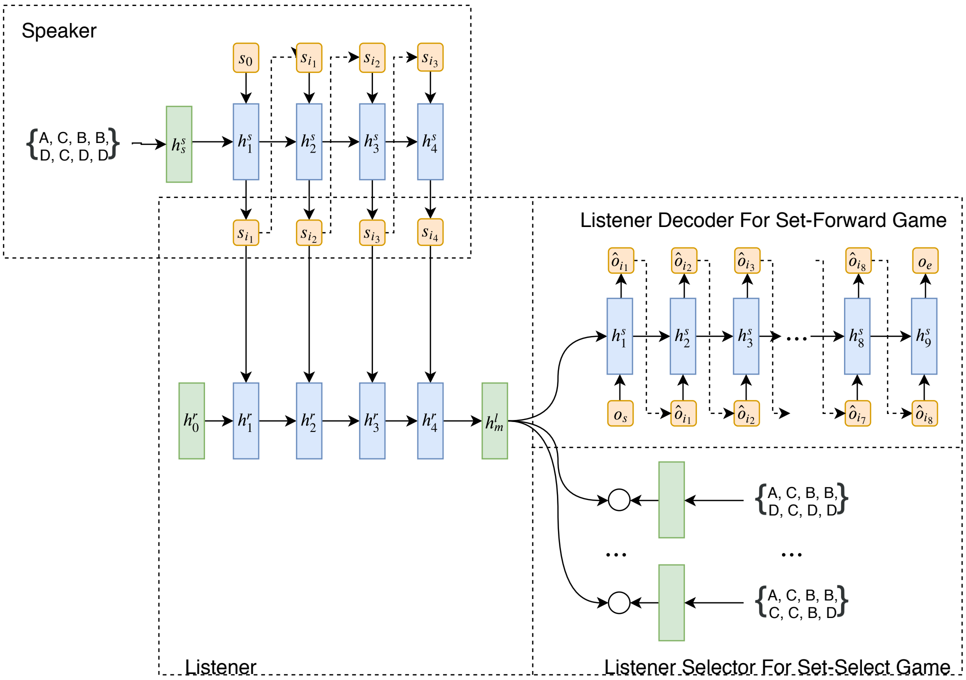


Figure 3.3: Overall Diagram of Model Architectures for Playing Games.

We illustrate the overall architecture of our models in Figure 3.3 given above. A speaker S consists of 2 components: i) a set encoder that takes a set of objects as input and outputs its vector representation h_s^s; ii) a standard LSTM sequence decoder that can generate a message $s_{i_1}, s_{i_2}, s_{i_3}, \ldots$ based on h_s^s. As for a listeners L, it would first encode messages with a LSTM sequence encoder and get the feature vector h_m^l. Then, in the Set-Reconstruct game, L would take h_m^l as the initial hidden state and predict a sequence of objects with a LSTM sequence decoder. As for in Set-Select game, L would compare h_m^l with several sets which are encoded by set encoders of L and select the one shown to S based on the dot product between h_m^l and feature vectors of each set.

Further details are shown in the following subsections.

[2] A distractor is a set that contains different numbers of objects as the correct one.

3.2.1 Speaker

The architecture of our speaking agents is very similar to the Seq-to-Seq model proposed by [Sutskever et al., 2014] except that replace the encoder for input sequences with a set encoder whose details are introduced in the following subsubsection. As Seq-to-Seq model is quite popular nowadays, we skip details about how to generate sequences which correspond to the messages in our games, and focus on how to encode sets of objects.

3.2.1.1 Set Encoder

Our set encoder shares an almost same architecture of inputting sets proposed by [Vinyals et al., 2015]. However, as there is an addition in $softmax$ function and it would introduce counting bias into the feature representation of sets, we replace equation (5) in [Vinyals et al., 2015] with the following operation in order to avoid exposing counting system to models:

$$a_{i,t} = \sigma(e_{i,t}) \tag{3.2}$$

where σ is sigmoid function.

Thus, assume the input for speaker S is a set $s_i = \{o_{i_1}, o_{i_2}, \dots\ o_{i_n}\}$. The first step is to read the set s_i as a sequence and project all objects to dense vectors by an embedding layer. Based on the sequence $\{w^s_{i_1}, w^s_{i_2}, \dots\ w^s_{i_n}\}$ (where $w^s_{i_k}$ is the embedding vector of o_{i_k} for speaker where $k \in \{1, \dots, n\}$), the calculation of h^s_s can be given as follows:

$$\begin{aligned} e^s_{i,t} &= f(q^s_{t-1}, w^s_i) \\ a^s_{i,t} &= \sigma(e^s_{i,t}) \\ r^s_t &= \sum_i a^s_{i,t} w^s_i \\ q^s_t &= LSTM(r_t, q^s_{t-1}, c^s_{t-1}) \end{aligned} \tag{3.3}$$

where $t \in \{1, \dots, T\}$ is the number of attention times, f is an affine layer, q^s_t and c^s_t are hidden and cell states in LSTM respectively.

Besides, in our implementation, T is set to be the same as the number of all types of objects, as we want to help models to represent number of each kind of objects as features in the vector representation of input set.

3.2.1.2 Message Generator

To generate the message m_i, we follow [Havrylov and Titov, 2017] and adopt a LSTM-based sequence decoder with 2 different kinds of sampling mechanisms: i)direct sampling that directly sample from the corresponding categorical distribution specified by $softmax(Wh_k + b)$; ii) GUMBEL-softmax estimator proposed by [Jang et al., 2016] with straight-through trick introduced in [Bengio et al., 2013]. Beside, the learning mechanisms also vary for these 2 different sampling methods, which is further discussed in Subsection 3.2.3.

Note that the length of each message m_i is fixed to and symbols $s_{i_1}, \dots, s_{i_{|M|}}$ are all one-hot vectors that represent different discrete symbols. The effect of number of all discrete message symbols $|V|$ and length of messages $|M|$ on the emergent language is further discussed in Chapter 4.

3.2.2 Listener

The architectures of listening agents are specifically designed for handling different kinds of tasks/games and thus vary from Set-Reconstruct game to Set-Select game.

Listener in Set-Reconstruct Game: The listener in Set-Reconstruct game has exactly the same architecture as Seq-to-Seq model proposed by [Sutskever et al., 2014]. And, when combined with speaker model, the overall model is called “Set2Seq2Seq”.

Listener in Set-Select Game: The listener in Set-Select game would also first encode messages with a LSTM like it is in standard Seq-to-Seq model. However, as it needs to select among several candidates, it also needs to encode all these sets with Set Encoder introduced in Subsection 3.2.1.1. Then, the listener would make predictions based on the dot-products between embedding of message h_m^r and embeddings of each set of objects. Similarly, when combined with speaker model, the overall model is called as “Set2Seq2Choice”.

3.2.3 Loss/Reward and Learning

In Set-Reconstruct game, as the predictions of listeners are a sequence of objects $\hat{s}_i = \{\hat{o}_{i_1}, \dots, \hat{o}_{i_n}\}$, we use cross-entropy between the original set and the predicted sequence as the objective function that needs to be minimised. Formally,

$$\mathcal{L}_{\theta^S, \theta^L}(o_{i_1}, \dots, o_{i_n}) = \mathbb{E}_{m_i \sim p_{\theta^S}(\cdot|s_i)} \left[-\sum_{k=1}^{n} o_{i_k} \log(p(\hat{o}_{i_k}|m_i, \hat{o}_{-i_k})) \right] \tag{3.4}$$

where $\hat{o}_{-i_k}$ represent all predicted objects preceding $\hat{o}_{i_k}$.

In Set-Select game, we still use the cross entropy between the correct candidate and as the loss to minimise, i.e.

$$\mathcal{L}_{\theta^S,\theta^L}(s_i) = \mathbb{E}_{m_i \sim p_{\theta^S}(\cdot|s_i)}\left[-\sum_{k=1}^{C} s_i log(p(c_k))\right] \tag{3.5}$$

where c_k is the predicted logit score for candidate k among C candidates.

In the case that we use GUMBEL-softmax for sampling messages from speaker S, parameters θ^S and θ^L are learnt by back-propogation. In the case that we use direct sampling, θ^L is still learnt by back-propogation, where as θ^S is learnt by REINFORCE estimator [Williams, 1992] with cross-entropy scores as rewards.

3.2.4 Neural Iterated Learning

The evolutionary linguistic community has already studied the origins and measurements of language compositionality since [Kirby and Hurford, 2002] which points out a cultural evolutionary account of the origins of compositionality and proposes iterated learning to model this procedure. Thus, to facilitate the emergence of compositionality among the autonomous communication between agents, we also trained our agents in a iterated learning fashion. In the original iterated learning, an agent can both speak and listen. However, in this project, agent can be either a speaker or a listener, not both at the same time. Thus, we slightly transform the iterated learning framework and call the following one "neural iterated learning" (NIL).

Following the overall architecture of iterated learning, we also train agents generation by generation. In the beginning of each generation t, we would instantiate a new speaker S_t and a new listener L_t and then execute the following 3 phases:

1. **Speaker Learning phase**: During this phase, we would train S_t with the set-message pairs generated by S_{t-1}, and the number of epochs is set to be fixed. Note that there is no such phase in the initial generation, as there is no set-message pair for training S_t.

2. **Game Playing phase**: During this phase, we would let S_t and L_t cooperate to complete the game and update θ^S_t and θ^L_t with loss/reward illustrated in previous section, and use early-stopping to avoid overfitting.

3. **Knowledge Generation phase**: During this phase, we would feed all s_i in training set into S_t and get corresponding messages m_i. Then, we would keep the sampled "language" for S_{t+1} to learn.

3.2.5 Baseline Models

To get the upper bounds of our multi-agent communication systems, we remove the communication between speaker and listener to be the baseline models.

In Set-Reconstruct game, our baseline is Set-to-Seq model which first encodes the input set s_i with the set encoder introduced in subsection 3.2.1.1 and then directly generate the predicted sequence $\hat{s}_i$ following the sequence generation in standard seq-to-seq model.

As for in Set-Select game, our baseline is Set-to-Choose model, in which speaker directly transmit representation vector h_s^s of set s_i to listener. And, listener compare h_s^s with all candidate sets to make a selection.

3.3 Measurements

With the recent rapid development of GLL, measuring the compositionality of emergent communication protocol attracts more and more attention nowadays, e.g. [Andreas, 2019], [Lowe et al., 2019].

However, as the setting of our game is simple and the space size is limited, we follow [Brighton and Kirby, 2006] and take the topological similarity between meaning space (space of all sets of objects) and message space as the measurement of compositionality. Briefly speaking, as much of language is neighbourhood related, i.e. nearby meanings tend to be mapped to nearby messages, the compositionality of language can be measured as the correlation degree between distances of meanings and distances of corresponding messages. For example, the meaning of set $\{A,A,A,B,B\}$ is closer to $\{A,A,B,B\}$ than $\{A,A,A,A,B,B,B\}$. In natural language (which is perfectly compositional), messages for $\{A,A,A,B,B\},\{A,A,B,B\},\{A,A,A,A,B,B,B\}$ are "3A2B", "2A2B" and "4A3B" respectively. However, in a non-compositional language, the messages may be "5B5A", "1C2E" and "3A4C".

In order to calculate the topological compositionality, we need define the distance measurement for meaning space and message space respectively. Thus, for an input set s_i, we could first count the number of each kind of object and then concatenate the

Arabic numerals as the meaning sequence. Take a set $s_i = \{A, A, A, B, B\}$ for example, the corresponding meaning sequence would be "32" as there are 3 A and 2 B in s_i.[3] As for the message space, we have several different settings which are further illustrated in subsection 4.2.2, and edit distance as in [Brighton and Kirby, 2006] is also included.

Meanwhile, as we could perfectly encode the meaning of a set into natural language, we could take the speaker as a machine translation model that translates a meaning represented in natural language into emergent language invented by computational themselves. Based on this point of view, we could also use BLEU score proposed by [Papineni et al., 2002] as a measurement of semantic similarities between messages. For the sets that share more similar meanings, we expect their corresponding messages to share more uni-grams, bi-grams and so on. Following the above example, in a perfectly compositional language, as $\{A, A, A, B, B\}$ locates very close to $\{A, A, B, B\}$, their messages ("3A2B" and "2A2B") share 3 uni-grams ("A", "B" and "2") and 2 bi-grams ("A2" and "2B") in common. However, in a non-compositional language, in which the messages for $\{A, A, A, B, B\}$ and $\{A, A, B, B\}$ are "5B5A" and "1C2E" respectively, the messages share no uni-gram and bi-gram in common.

In our case, the BLEU score between m_i and m_j is calculated as follow:

$$BLEU(m_i, m_j) = 1 - \sum_{n=1}^{N} \omega_n \cdot \frac{\text{Number of common } n\text{-grams}}{\text{Number of total different } n\text{-grams}} \tag{3.6}$$

where n is the size of n-grams and ω_n is the weight for similarity based on n-grams.

[3] Again, the appearing order of objects would not effect the meaning sequence of a set.

Chapter 4

Experiment Results and Discussion

4.1 Emergence of Language

First of all, we have to verify that the agents can successfully address the problems by communicating with discrete symbols, so that they communicate meaningful things with each other. Thus, we train both “Set2Seq2Seq” and “Set2Seq2Choice” on different game settings, and the performance of models are given in Table 4.1.

Model	Sampling Method	Performance	Game Setting
Set2Seq2Seq	GUMBEL	99.89%	$\|M\| = 8$, $\|V\| = 10$, $\|O\| = 6$, $\|N_o\| = 10$
	REINFORCE	89.89%	
	SCST	98.67%	
Set2Seq2Choice	GUMBEL	100%	$\|M\| = 6$, $\|V\| = 10$, $\|O\| = 4$, $\|N_o\| = 10$
	REINFORCE	76.45%	
	SCST	83.26%	

Table 4.1: Performance of Models and Corresponding Game Settings.

In the above table, $|M|$ is the length of messages, $|V|$ is the size of vocabulary[1] for message, $|O|$ is the number of all kinds of objects and $|N_o|$ is the maximum number of a single kind of object.

Besides the “REINFORCE” and “GUMBEL” sampling methods introduced in subsection 3.2.1.2, we also tried the self-critic sequence training proposed by [Rennie et al., 2017] as a baseline for REINFORCE algorithm, which is denoted by “SCST”.

[1] Note that the meaning of “vocabulary” is not like it is in traditional NLP, but refers to the set of initially meaningless symbols that can be used for commmunication.

Based on the performance shown in Table 4.1, it is clear that GUMBEL is the most stable training mechanism on all different settings. Thus, unless specifically stated, the following experiments and discussions are all based on training with GUMBEL method.

4.2 Structure of Emergent Language

4.2.1 Emergent Languages in Various Games

After verifying that computational agents could always complete games through commmunication, we are curious about the messages produced during their communication. However, unlike what was shown by the previous works in GLL, e.g. [Hill et al., 2017] and [Mordatch and Abbeel, 2018], the emergent language during both Set-Reconstruct and Set-Select games are not "perfectly" compositional. One reason of this phenomenon is that $|M| > |O|$ in our game settings, as we want to avoid the effects brought by fine-tuning hyperparameters.

To have give an intuitional demonstration of the emergent language, we list all messages transmitted in a Set-Reconstruct game where $|O| = 2, |N_o| = 5, |M| = 4, |V| = 10$ in Table 4.2 given as follow. In the table, the first row and first column are the basic elements of meanings and each cell is the corresponding message for that meaning. Take cell "1A2B" for example, the original input set is $s_i = \{A, A, B, B, B\}$ and the corresponding message m_i is "7751". Note that the numbers in the message do not correspond to the numerals in natural language.

	0A	1A	2A	3A	4A	5A
0B		7377	7317	3711	3111	3353
1B	7737	7731	7111	1715	1151	5135
2B	7773	7751	7181	7515	1585	5551
3B	7775	7754	7815	7545	8515	4515
4B	7787	7874	7841	8851	8455	4455
5B	7788	7888	7844	8848	8444	4444

Table 4.2: An emergent language in a Set-Reconstruct game.

As we can see from Table 4.2, there is no clear compositional structure in it. To better define the compositional elements, we argue that if a language is said to be perfect compositional, then it should satisify the following 2 properties:

- **Mutual Exclusivity**: Symbols describing different values of a same property should be mutually exclusive to each other. For example, “green” and “red” are both used to describe color of an object and they should not appear at the same time as an object can not be green and red at the same time.
- **Orthogonality**: Appearance of symbols for describing a property should be independent from the appearance of symbols used to describe another property. For example, the appearance of symbols used for describing colors of objects should be independent from the appearance of symbols used for describing shapes of objects.

Based on the above assumptions, we could see that the emergent language shown in Table 4.2 satisifies neither of mutual exclusivity nor orthogonality. Thus, the emergent language is not a perfectly compositional one as we expect.

However, as Set-Reconstruct game is a generation task, the agents may transmit more than numeric concepts in order that listeners could generate the original input. Thus, to verify whether this is the case, we illustrate an emergent language in a Set-Select game whose settings are exactly the same as the Set-Reconstruct game illustrated above. The meanings and corresponding messages are shown in Table 4.3 given as follow.

	0A	1A	2A	3A	4A	5A
0B		3335	3352	3522	5232	5222
1B	3333	3313	3435	3555	5533	5522
2B	3323	3033	3131	3445	5453	5555
3B	3232	3023	3003	1311	4443	4545
4B	2323	2302	0302	0130	1131	1441
5B	2232	2202	0222	0022	1000	1111

Table 4.3: An emergent language in a Set-Select game.

Based on the message contents in Table 4.3, we could find that the referential game does not make the emergent language perfectly compositional. According to [Kottur et al., 2017], another alternative probability is that the message space is much larger in the previous game settings and thus it is overcomplete for agents to encode the sets of objects in a compositional fashion. Thus, we re-train that agents with $|O| = 2, |N_o| = 5, |M| = 2, |V| = 10$, and the emergent language is shown in Table 4.4. As we can see, the smaller meaning space does not neccessarily facilitate the emergence of compositional language in Set-Select game.

	0A	1A	2A	3A	4A	5A
0B		15	51	55	55	35
1B	12	16	14	41	45	34
2B	21	66	11	17	44	43
3B	21	62	60	01	71	74
4B	23	22	69	00	07	77
5B	32	29	92	93	30	37

Table 4.4: Another emergent language in a Set-Select game.

4.2.2 Topological Similarities

As introduced in subsection 3.3, we measure the topological similarity between meaning space and message space under a language as the compositionality of it. We list compositionality scores under different kinds of measurements in Talbe 4.5 given as follow.

	Ham+Edit	Ham+BLEU	Euclid+Edit	Euclid+BLEU
Compositional	1.00	0.61	0.38	0.24
Set-Reconstruct	0.32	0.27	0.60	0.65
Set-Select	0.13	0.16	0.45	0.52
Holistic	-0.04	-0.04	0.01	0.00

Table 4.5: Topological similarity scores of different languages.

Ham+Edit: We first follow the distance measurements in [Brighton and Kirby, 2006]: i) use hamming distances between meaning sequences as the similarity measurement for meaning space; ii) use edit distances between corresponding messages as the similarity measurement for message space.

Ham+BLEU: In this setting, we use: i) hamming for meaning space too; ii) BLEU score illustrated in Section 3.3 as the the similarity measurement for message space.

Euclid+Edit: In this setting, we use: i) Euclidean distance as the measurement for meaning space, e.g. Euclidean distance between "4A2B" and "1A3B" is $\sqrt{(4-1)^2+(2-3)^2} = \sqrt{10}$; ii) edit distance for message space.

Euclid+BLEU: In this setting, we use: i) Euclidean distance for meaning space; ii) BLEU score illustrated in Section 3.3 for message space.

To get the upper bound and lower bound of compositionalities, we specifically designed: i) a perfectly compositional language, in which the message is exactly the same as meaning sequence, e.g. "4A2B" is represented as "4829" ($A \to 8, B \to 9$); ii) a holistic language, in which messages are randomly generated.

Then, from the above results, we could see that although the emergent languages in Set-Reconstruct and Set-Select games gain low topological similarity scores under Hamming distance for meaning space, they obtain much higher similarity scores under Euclidean distances for meaning space. Therefore, we argue that, although the emergent languages do not look like compositional, they can reflect the underlying structure of the meaning space. Based on the above results, we could also say that the meaning of symbols in the emergent language **is not numerals** defined in Subsection 3.1.2.

4.2.3 Significance Test of Same Numeric Concepts

Although the compositionality of emergent language is not like our natural language, we could also see that it reflects the underlying structure of meaning space. Thus, we further verify whether messages for meaning pairs that share same numeric concepts are more similar. To do this, we established 2 different datasets: i) meaning pairs sharing exactly same numeric concepts, e.g. "4A3B" and "3A4B", and corresponding BLEU similarity scores for their messages; ii) pairs of meaning sequences that share no numeric concept, e.g. "4A3B" and "5A1B", and corresponding BLEU similarity for their messages.

Then, we establish the following hypotheses for significance test:

- **Null hypothesis**: The BLEU scores between messages are independent from whether meaning pairs share same numeric concepts.
- **Alternative hypothesis**: The BLEU scores between messages are **not** independent from whether meaning pairs share same numeric concepts.

The the p-values got on different languages are given in Table 4.6 as follow.

Language	Compositional	Set-Reconstruct	Set-Select	Holistic
p-value	1.93×10^{-69}	7.01×10^{-3}	2.87×10^{-3}	0.57

Table 4.6: p-values of different languages.

The p-values for compositional language as well as emergent languages in both Set-Reconstruct and Set-Select games are smaller than 0.01. Thus, it is safe to reject null hypothesis and accept the alternative hypothesis. That is, The BLEU scores between messages depend on whether their meaning pairs share same numeric concepts. To be more precise, the messages of meaning pairs share same numeric concepts have more unigrams and bi-grams in common.

4.2.4 Generalisation of Emergent Language

To verify whether the emergent language can be generalised to unseen sets, we train the randomly initialised listeners with several kinds of languages: i) compositional language; ii) an emergent language invented by other agents; iii) holistic language. And, the game settings are $|M| = 8$, $|V| = 10$, $|O| = 4$, $|N_o| = 10$. Learning and performance curves of these languages on Set-Reconstruct and Set-Select games are given in Figure 4.1 and Figure 4.2 respectively.

It is quite suprising that, although we cannot see any significant pattern in the emergent language, it actually can be generalised to unseen sets of objects by listeners, as illustrated by the performance of listeners on evaluation dataset. Also, listener trained with emergent language converges faster on evaluation performance as well as training loss, although length of emergent messages ($|M| = 8$) are longer than that of compositional language ($|M| = 4$).

Besides, we also train listeners in Set-Reconstruct and Set-Select with languages emerged in the other game, i.e. “emergent-select” in Figure 4.1 and “emergent-forward” in Figure 4.2. From the evaluation accuracy in Figure 4.2, we could see that the emergent language in Set-Reconstruct game can be well generalised to unseen samples by listeners in Set-Select game. However, listeners in Set-Reconstruct game cannot generalise emergent language from Set-Select game, which is illustrated by the evaluation accuracies in Figure 4.1. This phenomenon demonstrates that the information encoded by speakers in Set-Reconstruct games are richer than the information encoded by speakers in Set-Select games, or to say this demonstrates that speakers in Set-Reconstruct games may encode more than only numeric concepts.

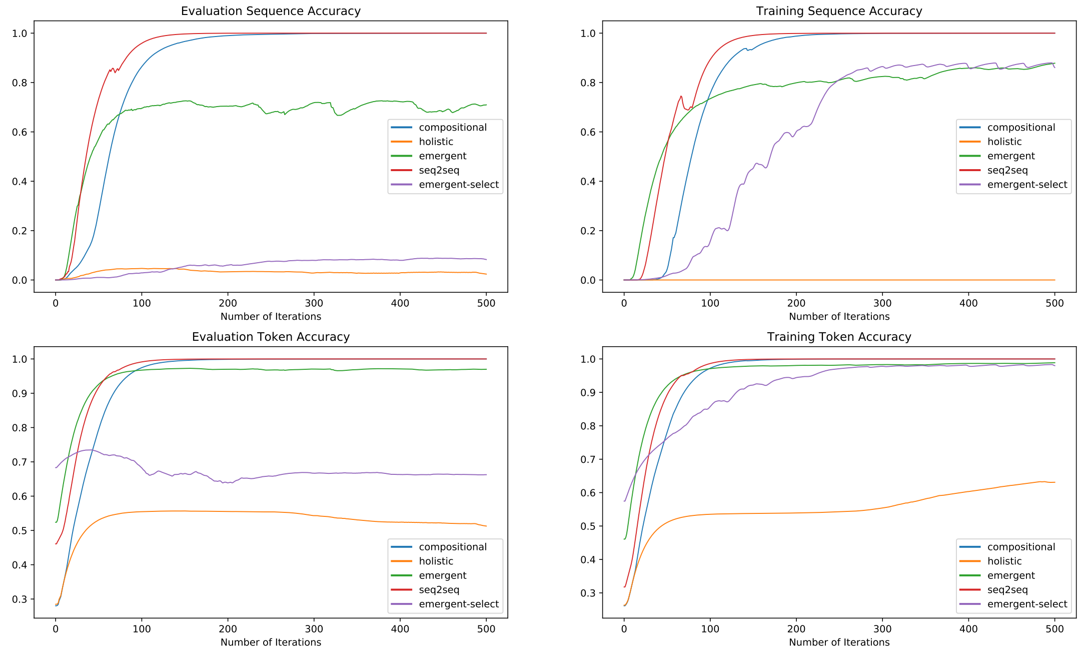


Figure 4.1: Learning and performance curves of different languages in Set-Reconstruct game.

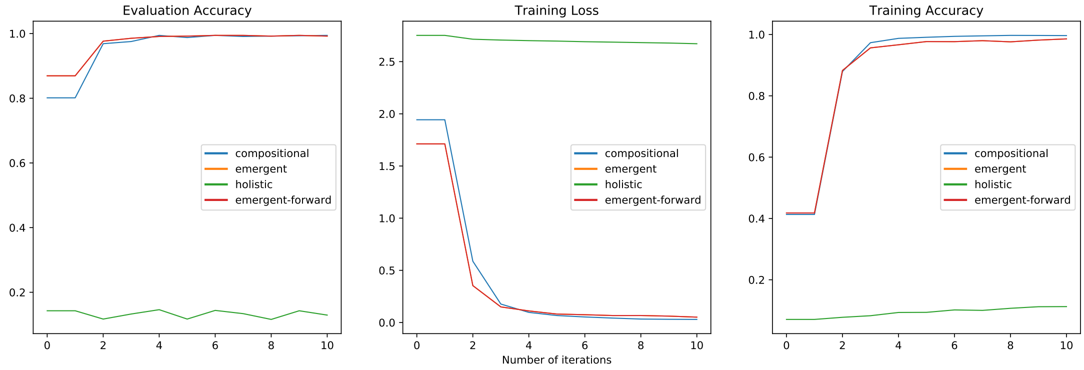


Figure 4.2: Learning and performance curves of different languages in Set-Select game. Lines of “emergent” and “emergent-forward” overlap with each other.

Sum up from all above, although we cannot find observable patterns in emergent languages under various game settings, the emergent languages are actually easier for agents to learn and also can be generlaised to unseen sets of objects. Thus, based on the previous topological similarity measurements and significance test, we claim that the emergent languages do capture the underlying structure of meanins space and encode them into sequences consisting of discrete variables.

4.3 Learning Speed & Iterated Learning

From the previous sections, we could see that the emergent languages can reflect the underlying structure of meaning spaces, although they may not be as compositional as our natural languages. Thus, we are further curious about the motivation of the emergent language. Or, to say, the reasons why computational agents prefer to communicate in such a “non-natural”[2] way.

4.3.1 For Listener

The first thing we are curious is whether the emergent language is the most easy one for listeners to understand. To verify this, we test the learning speeds of all kinds of languages with randomly initialised new listeners in both Set-Reconstruct game and Set-Select Game, and the results are shown in Figure 4.3 and Figure 4.4 respectively.

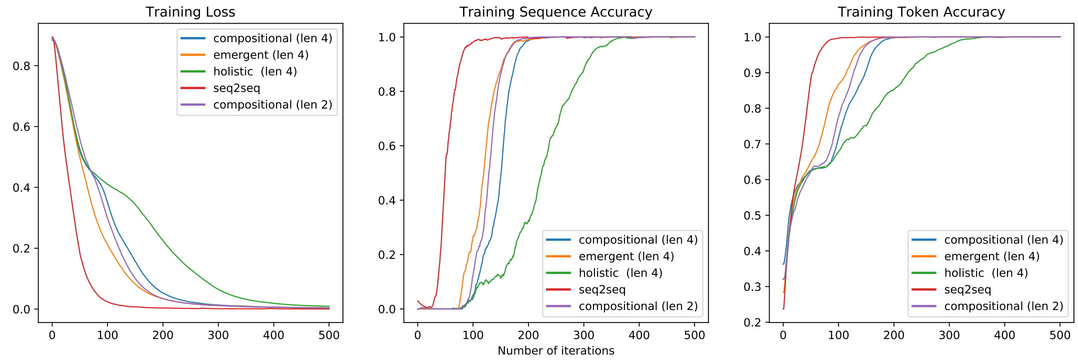


Figure 4.3: Learning and performance curves of different languages in Set-Reconstruct game.

From the above figures, we could easily see that emergent languages are learnt faster than compositional and holistic language in whichever the game, which implies that the emergent language has a lower sample complexity[Vapnik, 2013].

4.3.2 For Speaker

We then test the learning speed of different languages on speaker, i.e. we randomly intialise new speakers and let it learn to produce messages of input sets under different languages. Note that the architecture of speaker is identical in Set2Seq2Seq and Set2Seq2Choice model, thus all the curves are shown in Figure 4.5 given as follow.

[2] From a human perspective, it is not like how we communicate numeric concepts through natural language.

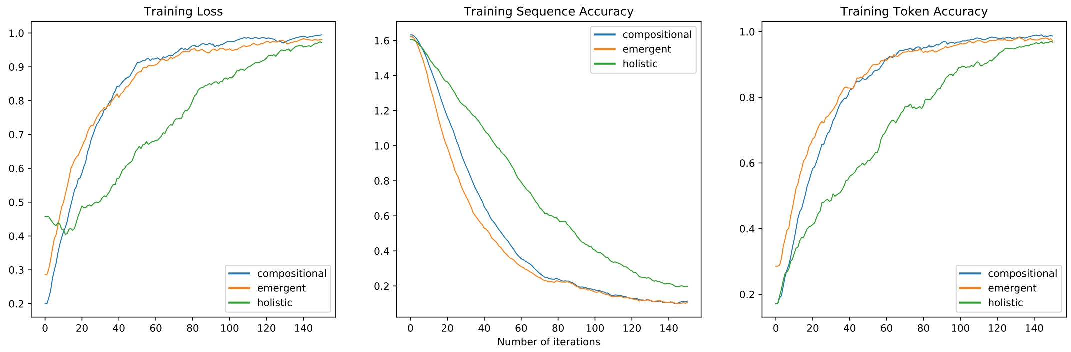


Figure 4.4: Learning and performance curves of different languages in Set-Select game.

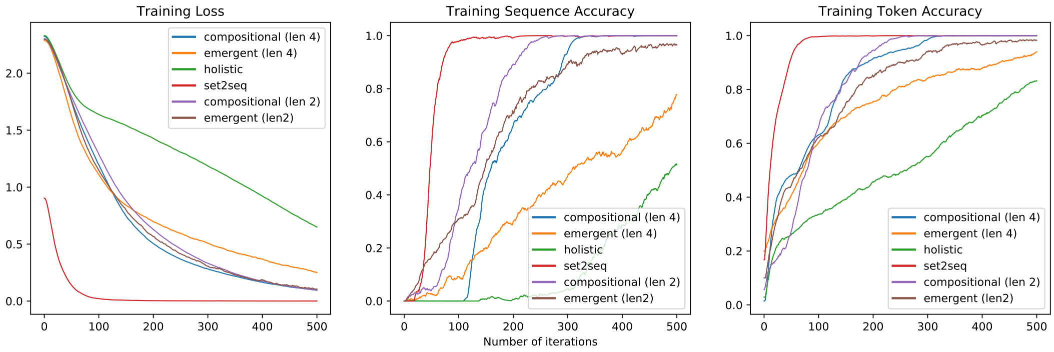


Figure 4.5: Speaker learning curves of different languages.

One thing we need to mention is that we also test an emergent language in Set-Select game with $|M| = 2, |V| = 10, |O| = 2, |N_o| = 5$ which is denoted as “emergent (len 2)”. Compared learning messages with length 2 and 4, we could easily see that smaller message spaces are always easier to learn than the larger ones.

Meanwhile, it is clear compositional languages are always easier for speaker to learn than the same sized emergent languages, which is contradictory to the situation on listener side. Our hypothesis is that compositional language is smoother function for speark to learn and thus it is easier to be optimised. However, as time is limited, this phenomenon is not further discussed in this work but will be explored in the future works.

4.3.3 Improvement by Iterated Learning

Although the iterated learning framework [Kirby and Hurford, 2002] is proven to be effective in experiments with both Bayesian agents and humans, there are several obstacles for directly applying it into our neural agents:

1. we cannot feed prior probability that favors high compositional languages to neural netowrks;

2. the pre-training procedure in learning phase of original iterated learning need to be re-designed, as spearkers and listeners in our game are not inverse functions to each other.

Thus, we adapt iterated learning into our project, which is illustrated in Subsection 3.2.4, and train agent population with respect to normal training mechanism and iterated learning. The results are shown in Figure 4.6. It needs to be pointed out that the disctance measurement for meaning space is Euclidean disctance for topological similarity, and measurement for message space is edit distance.

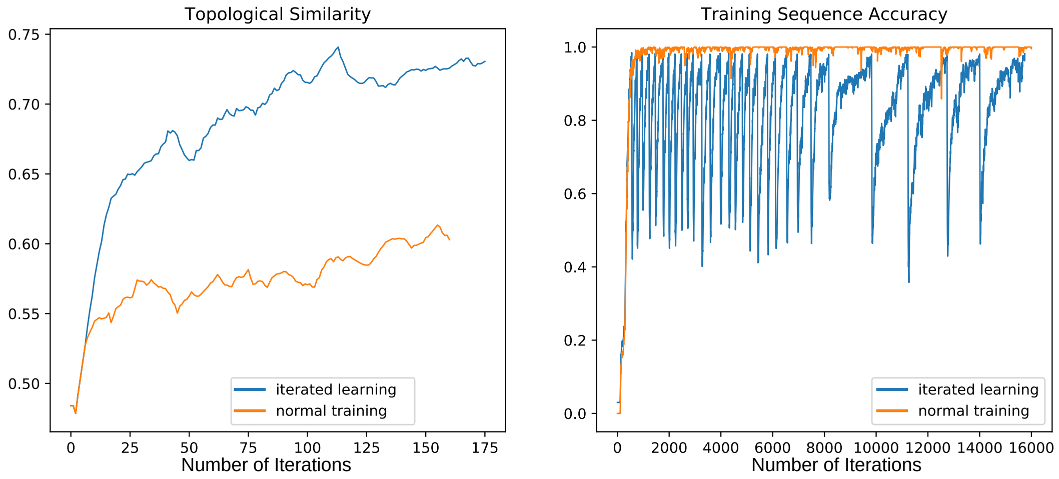


Figure 4.6: Topological similarity curves of iterated learning and normal training in Set-Select game.

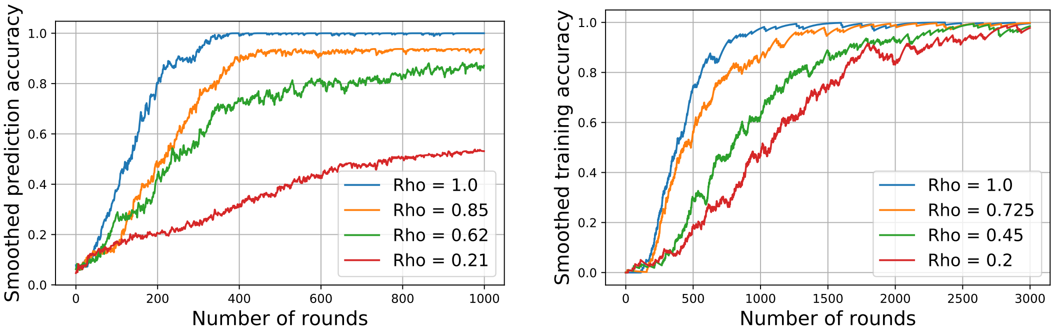
By comparing the curves of iterated learning and normal training, we can see a significant improvement of topological similarity in iterated leanring, about 0.1. However, althouth the messages emerged in iterate learning becomes more correlated with Euclidean distances between meanings, the numerica conceptes in them are still not represented like numerals in natural languages.

4.4 Effects of Different Representations

Compared our results in Section 4.1 to 4.3 with previous works in GLL, we argue that the different characteristics of emergents languages in our works are due to the feature representations of meanings.

To be specific, in our games, listeners need to generate object sequences or select the correct object sequence according to features representing each kind of objects. For example, the feature representation of set $\{A,B,A\}$ would be a sequence $\{[10],[01],[10]\}$ (assume that $|O| = 2, |N_o| = 8$), and the corresponding message would be $\{2,1\} = \{[001000000],[010000000]\}$ (assume that $|M| = |O| = 2, |V| = |N_o| = 8$). Thus, to understand the message, the listener needs to correctly count the numbers of each kind of obects in the set and ground symbols to the counting results. During this procedure, there are 2 gaps between meanings (or perceptions) and messages: i) from meaning to numeric concpets; ii) from numeric concepts.

To verfy which step imports bias towards emergent language, we slightly change the representation of sets in Set-Select game, i.e. we directly encode the numbers of each kind of objects as one-hot vectors and concatenate them to be the representation of the whole set. Take set $\{A,B,A\}$ as example, its representation would be **vector** $[001000000;010000000]$, whereas its message is still the **sequence** $\{[001000000],[010000000]\}$. Then, it is straightforward that mapping from messages to meanings is a linear transformation and thus it should be easy for neural networks to fit.



(a) Listener learning speed

(b) Speaker learning speed

Figure 4.7: Learning speed of languages with different compositionalities with linear feature representations.

First of all, we test the learning speed of manually designed languages with different topological similialrty scores on both speaker and listener side, and the results are shown in Figure 4.7. Note that the measurement for meaning distance is Hamming distance and thus languages with higher ρ-values would "look" more like our natural language. As we can see in Figure 4.7, language with higher ρ-values are much more easier to learn for both speaker and listener, under the current scenario.

Then, we track the probabilities with different ρ-values during the iterated leaning procedure and the results are shown in Figure 4.8. It is straightforward to see that high compositional languages gradually dominate among all kinds of languages generation by generation.

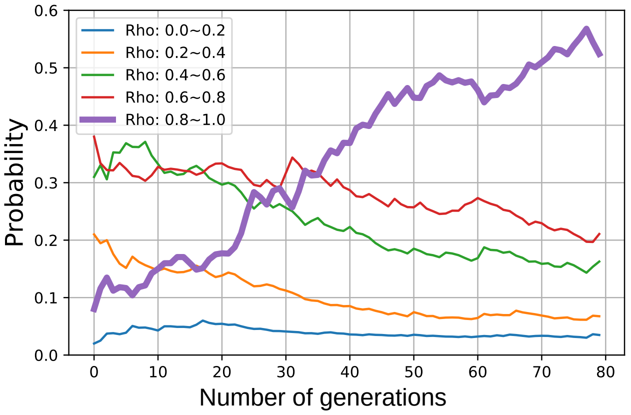


Figure 4.8: Changes of probabilities of languages with different ρ-values during iterated learning.

To have an intuitive feeling about the final emergent language with iterated learning and current feature representations, we illustrate it in Table 4.7.

	0A	1A	2A	3A	4A	5A	6A	7A	8A
0B		20	60	30	30	10	50	70	00
1B	42	22	62	82	32	12	52	72	02
2B	48	28	68	88	38	18	58	78	08
3B	47	27	67	87	37	17	57	77	07
4B	46	26	66	86	36	16	56	76	06
5B	45	25	65	85	35	15	55	75	05
6B	41	21	61	85	31	11	51	71	01
7B	44	24	64	84	34	14	54	74	04
8B	33	23	63	83	33	13	53	73	03

Table 4.7: Final emergent language in liear feature representation and iterated learning.

Then, it is straightforward to decipher the emergent language shown in Table 4.7. Basically, the symbols appeared in the first digit represent the numbers of "A" and the symbols appeared in second digit represent the numbers of "B". Of course, the language is still not perfect compositional, as there are some repetitive messages for different meanings, such as “3A5B” (85) and “3A6B” (85). Besides, it worth mentioning that the same symbol still represents different meanings if it appears at different positions.

Overall, we could say the the obstacle for the emergence of compositional languages in our Set-Reconstruct and Set-Select games is that symbols in messages do not directly correspond to any feature in the original meaning spaces. As long as the features we want the emergent language to represent is established, the agents could invent almost perfect compositional language by iterated learning.

4.5 Further Discussion

Comparing the experimental results in this chapter with previous work in GLL, e.g. [Kottur et al., 2017, Hermann et al., 2017, Havrylov and Titov, 2017, Mordatch and Abbeel, 2018], we propose an alternative hypothesis to explain the emergence of compositional language (some previous works call it natural language) during the autonomous communication among agents population.

First of all, we argue that the feature vectors of input experience and perceptions should be inherently disentangled, i.e. the feature vectors of these inputs should satisify mutual exclusivity and orthogonality defined in Subsection 4.2.1, in order to facilitate the emergence of compositional language. Then, it could be an optimal method to use a single symbol as a feature representation of a disentangled element in feature vectors. By comparing the emergent languages in Section 4.1 to 4.3 with that in Section 4.4, it is straightforward to see that linear transformed feature representations would be much more optimal for the emergence of highly compositional languages. However, as lots of previous use images as the perceptions for speakers, there is still a gap between our 2 representing methods. Without further experiment, we are not sure about whether the emergence of compositional languages of those works are caused by that convolutional neural networks (CNN) can spontaneously encode images into independent features.

Secondly, we argue that iterated learning is an effective method to introduce inductive bias into multi-agent autonomous communication systems, and thus improve the compositionalities of emergent languages. Considering the discoveries in [Locatello et al., 2018], we claim that the compositional languages are highly correlated with the appearance of disentangled representations. Further, inductive bias towards compositionalities of different kinds of symbols (which correpsond to words in natural languages) should be introduced to different spaces. For example, inductive bias towards the compositionality of symbols corresponding to objects/attributes that physically exists in real/virtual world can be introduced by iterated learning, as the feature values of these objects/attributes are mutually exclusive and independent (or to say, they are inherently disentangled). On the other hand, compositionality of function words, such as numerals in our project, requires the agents to first encode the input features in some specific ways and obtain disentangled representations. Thus, without specially designed training mechanism or data samples that could introduce such pressure, it is natural for agents to invent effective but non-natural "languages" during their autonomous communication.

Chapter 5

Conclusions

5.1 Express Numeric Concepts

With all experimental results shown in Chapter 4, we could conclude that the models illustrated in Chapter 3 can successfully transmit numeric concepts in whichever Set-Forward or Set-Select game proposed in Chapter 3. Although the emergent languages are not compositional from the perspective of humans, the agents do capture the underlying structure of meaning space and reflect it by the emergent language, which is measured by the Euclidean distances between meaning pairs. Further, the messages expressing same numeric concepts have higher similarities to each other, which is measured the BLEU score defined in Section 3.3. More importantly, the emergent languagess can be successfully generalised to unseen meanings and they are not only effective but also efficient, as they have lower sample complexity for models to fit.

Therefore, we claim that the agents capture the numeric concepts during cooperating to complete the game and successfully transmit these numeric concepts with a non-natural language.

5.2 Role of Iterated Learning

By transforming iterated learning to train our DL-based agents, it successfully improves the compositionalality of emergent languages, which is measured by Euclidean distances in meaning space and BLEU score in message space, in our original set representations of objects. Then, by taking vectors that directly encode quantities of different kinds of objects as the input for spearkers, the emergent languages become almost perfectly compositional under iterated learning.

Therefore, we claim that iterated learning is an effective method to improve the compositionalality of emergent languages, w/o inherently disentangled feature representation of inputs. Even thought the emergent compositionalality may not correspond to what it is in human natural languages.

5.3 Future Works

With the current exploration, there are still several open questions in our work and thus serveral interesting and meaningful future works:

1. **Generalisation and meta-learning**: [Smith et al., 2013] claims that language structure is an evolutionary trade-off between simplicity and expressivity. We assume that generalisation is another form of this trade-off. Further, emergence of numerals is a good candidate for discovering the role of generalising pressure in language evolution, as numerals can be used for whatever kind of objects and such pressure can be formalised by meta-learning.

2. **Feature representation**: As discussed in Section 4.4, different kinds of representations have a strong effect on the compositional form of emergent languages. Argued by [Locatello et al., 2018], representation learnt without supervision are not disentangled. We further assume that inherently disentangled elements are not only important in the input feature space but also in the parameter feature space. Or, to say, some words in our natural languages directly correspond to elements in input feature representations, while others may correspond the features of specific functions, e.g. agents need to learn counting (a function) in our games.

Bibliography

[Andreas, 2019] Andreas, J. (2019). Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*.

[Baronchelli et al., 2006] Baronchelli, A., Felici, M., Loreto, V., Caglioti, E., and Steels, L. (2006). Sharp transition towards shared vocabularies in multi-agent systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06014.

[Batali, 1998] Batali, J. (1998). Computational simulations of the emergence of grammar. *Approach to the Evolution of Language*, pages 405–426.

[Bengio et al., 2013] Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

[Berwick and Chomsky, 2016] Berwick, R. C. and Chomsky, N. (2016). *Why only us: Language and evolution*. MIT press.

[Bickerton and Szathmáry, 2009] Bickerton, D. and Szathmáry, E. (2009). *Biological foundations and origin of syntax*, volume 3. Mit Press.

[Brighton and Kirby, 2006] Brighton, H. and Kirby, S. (2006). Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242.

[Briscoe, 1998] Briscoe, E. J. (1998). Language as a complex adaptive system: co-evolution of language and of the language acquisition device. In *Proceedings of eighth computational linguistics in the Netherlands Conference*, pages 3–40. Citeseer.

[Briscoe, 2002] Briscoe, T. (2002). *Linguistic Evolution through Language Acquisition: Formal and Computational Models*. Cambridge University Press.

[Cangelosi and Parisi, 2012] Cangelosi, A. and Parisi, D. (2012). *Simulating the evolution of language*. Springer Science & Business Media.

[Cao et al., 2018] Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. (2018). Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*.

[Christiansen and Kirby, 2003] Christiansen, M. H. and Kirby, S. (2003). *Language evolution*. OUP Oxford.

[Da Costa and Bond, 2016] Da Costa, L. M. and Bond, F. (2016). Wow! what a useful extension! introducing non-referential concepts to wordnet. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4323–4328.

[Darwin, 1859] Darwin, C. (1859). On the origin of species by means of natural selection (j. murray, london).

[Evans and Levinson, 2009] Evans, N. and Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and brain sciences*, 32(5):429–448.

[Gong and Shuai, 2013] Gong, T. and Shuai, L. (2013). Computer simulation as a scientific approach in evolutionary linguistics. *Language Sciences*, 40:12–23.

[Havrylov and Titov, 2017] Havrylov, S. and Titov, I. (2017). Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in neural information processing systems*, pages 2149–2159.

[Hermann et al., 2017] Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., et al. (2017). Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.

[Hill et al., 2017] Hill, F., Hermann, K. M., Blunsom, P., and Clark, S. (2017). Understanding grounded language learning agents. *arXiv preprint arXiv:1710.09867*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hurford, 1989] Hurford, J. R. (1989). Biological evolution of the saussurean sign as a component of the language acquisition device. *Lingua*, 77(2):187–222.

[Hurford, 1999] Hurford, J. R. (1999). Artificially growing a numeral system. `http://www.lel.ed.ac.uk/~jim/grownum.html`. Accessed March 12, 2019.

[Hurford et al., 1998] Hurford, J. R., Studdert-Kennedy, M., and Knight, C. (1998). *Approaches to the evolution of language: Social and cognitive bases*. Cambridge University Press.

[Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

[Ke et al., 2008] Ke, J., Gong, T., and Wang, W. S. (2008). Language change and social networks. *Communications in Computational Physics*, 3(4):935–949.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[Kirby, 1999] Kirby, S. (1999). *Function, selection, and innateness: The emergence of language universals*. OUP Oxford.

[Kirby and Hurford, 2002] Kirby, S. and Hurford, J. R. (2002). The emergence of linguistic structure: An overview of the iterated learning model. In *Simulating the evolution of language*, pages 121–147. Springer.

[Knight et al., 2000] Knight, C., Studdert-Kennedy, M., and Hurford, J. (2000). *The evolutionary emergence of language: social function and the origins of linguistic form*. Cambridge University Press.

[Kottur et al., 2017] Kottur, S., Moura, J. M., Lee, S., and Batra, D. (2017). Natural language does not emerge'naturally'in multi-agent dialog. *arXiv preprint arXiv:1706.08502*.

[Lewis, 1969] Lewis, D. (1969). *Convention: A philosophical study*. John Wiley & Sons.

[Lieberman, 2006] Lieberman, P. (2006). *Toward an evolutionary biology of language*. Harvard University Press.

[Locatello et al., 2018] Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schölkopf, B., and Bachem, O. (2018). Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*.

[Lowe et al., 2019] Lowe, R., Foerster, J., Boureau, Y.-L., Pineau, J., and Dauphin, Y. (2019). On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 693–701. International Foundation for Autonomous Agents and Multiagent Systems.

[Matuszek, 2018] Matuszek, C. (2018). Grounded language learning: Where robotics and nlp meet. In *IJCAI*, pages 5687–5691.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

[Minett and Wang, 2008] Minett, J. W. and Wang, W. S. (2008). Modelling endangered languages: The effects of bilingualism and social structure. *Lingua*, 118(1):19–45.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

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

[Munroe and Cangelosi, 2002] Munroe, S. and Cangelosi, A. (2002). Learning and the evolution of language: The role of cultural variation and learning costs in the baldwin effect. *Artificial life*, 8(4):311–339.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of* *the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Puglisi et al., 2008] Puglisi, A., Baronchelli, A., and Loreto, V. (2008). Cultural route to the emergence of linguistic categories. *Proceedings of the National Academy of Sciences*, 105(23):7936–7940.

[Rennie et al., 2017] Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

[Siegelmann and Sontag, 1992] Siegelmann, H. T. and Sontag, E. D. (1992). On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 440–449, New York, NY, USA. ACM.

[Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

[Smith et al., 2003a] Smith, K., Brighton, H., and Kirby, S. (2003a). Complex systems in language evolution: the cultural emergence of compositional structure. *Advances in Complex Systems*, 6(04):537–558.

[Smith et al., 2003b] Smith, K., Kirby, S., and Brighton, H. (2003b). Iterated learning: A framework for the emergence of language. *Artificial life*, 9(4):371–386.

[Smith et al., 2013] Smith, K., Tamariz, M., and Kirby, S. (2013). Linguistic structure is an evolutionary trade-off between simplicity and expressivity. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

[Steels, 2005] Steels, L. (2005). The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. *Connection science*, 17(3-4):213–230.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Vapnik, 2013] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

[Vinyals et al., 2015] Vinyals, O., Bengio, S., and Kudlur, M. (2015). Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*.

[Vogt, 2005] Vogt, P. (2005). On the acquisition and evolution of compositional languages: Sparse input and the productive creativity of children. *Adaptive Behavior*, 13(4):325–346.

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

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical investigations*. John Wiley & Sons.

[Yu et al., 2018] Yu, H., Zhang, H., and Xu, W. (2018). Interactive grounded language acquisition and generalization in a 2d world. *arXiv preprint arXiv:1802.01433*.