Emergence of Numerals in Multi-Agent Autonomous Communication System

Shangmin Guo

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

Abstract

This project aims to propose a new computational simulation method for the emergence of numerals based on multi-agent autonomous communication system following deep reinforcement learning methodology.

Acknowledgements

Any acknowledgements go here.

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 statae-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 descibe their experiences/perceptions (E/P). Under such a perspective, symbols in natural languages are actually feature representations of the original E/P, whereas most current NLP systems directly take these symbols as original features.

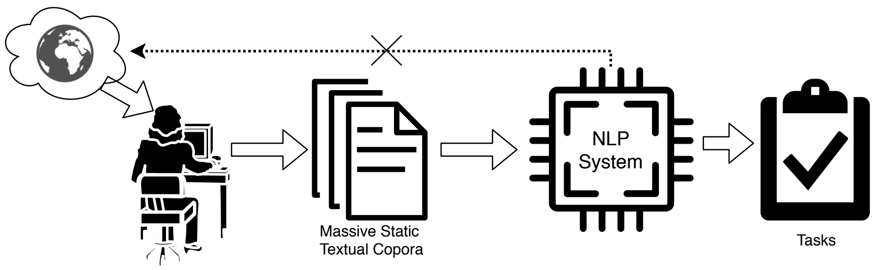

Figure 1.1: An overview of information flow in current NLP systems.

Considering the missing original E/P, 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 correspongding E/P and can complete the specified tasks, it is necessary to facilitate 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. [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 colors, were considered and to which discrete symbols were grounded to.

This project, on the other hand, aims to explore and analyse the grounding of abstractions which are non-referential (?) in the original experiences/perceptions of human beings. However, as it is too huge a topic to tackle, our project is limited to cardinal numerals 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 define numerals as symbols indicating numbers of replicating tokens when generating outputs.

2. Based on the language game, we successfully train agents to transmit numerical concepts which corresponds to function words in natural languages.

3. We transformed iterated learning proposed by [Smith et al., 2003] to train DL models and find that it does help to improve the compoitionality of the emergent language.

4. We further analyse and discuss the compositionality of the emergent communication protocal and HERE!.

Chapter 2

Background

As we demonstrate in Chapter 1, 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 GLL. 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

The emergence and evolution of natural language have always been critical questions to the field of evolutionary linguistics [MacWhinney, 2013] and one important issue is to use quantitative methods to overcome the time limit on unpreserved pre-historic linguistic behaviors[Lieberman, 2006, Evans and Levinson, 2009]. Since it was first introduced by [Hurford, 1989], computer simulation methods have 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 we introduced in Chapter 1, one of our objectives is to facilitate computational agents to discover and develop various kinds of natural langauge phenonmana, which shares a same objective and motivation of computer simulation methods in evolutionary linguistics.

To imply and verify linguistics theories, 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 protocal in cooperation between individuals as a prototype of language.

- *Genetic evolution* introduced by [Briscoe, 1998] which aims at simluating 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 diffrent kinds of resulting linguistic phenonmana, this simulation procedure can be broadly categorised intto 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 techiniques learn semantics from only large-scaled static textual materials. Thus, GLL argues that computational also need to learn and understand languages by interacting with environments and grounding language into their E/P. With the recent rapid development of deep reinforcement learning (DRL), it is proven that computational agents can master a variaty of complex cognitive activities, e.g. [Mnih et al., 2015, Silver et al., 2017]]. Thus, a bunch of works in GLL apply DRL techiniques to facilitate 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 competence instead of cooperation such as [Cao et al., 2018].

From another perspetive, 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 participator, in which the main concern is to explore how could computational agents learn the compositionality of semantics.

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

However, like we mentioned in Chapter 1, whichever kind of langauge 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, numerals) during the game.

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

Chapter 3

Game, Models and Evaluation Methods

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 obejcts.

- *Dynamics*: In this project, the dynamics mean not only the manually designed reward function for agents but also the training mechanism, 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 all the linguistic hypotheses are purely implied by dynamics.

3.1.1 Game Procedure

The overall view of the proposed Set-Forward game is illustrated in Figure 3.1 given as follow.

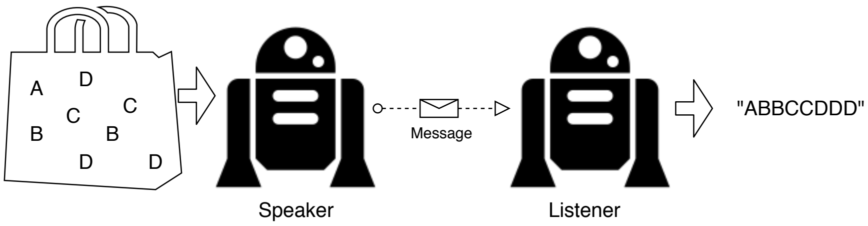

Figure 3.1: Diagram of Game Playing Procedure.

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 reproduction: 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 diffrence. Further, we argue that the only important feature that need to be transmitted is actually the numbers of different objects which correponds to the function of numerals in natural language.

[1]Different ways of updating parameters are instroduced 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 correpond 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 reproduce 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. Also, in our models, there is no specific mechanism to separate the meanings of symbols from their positions.

From the side of speakers S, a numeral preferred to be defined as a symbol s^n at **position** i that represents the numbers of specific object o_i. 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

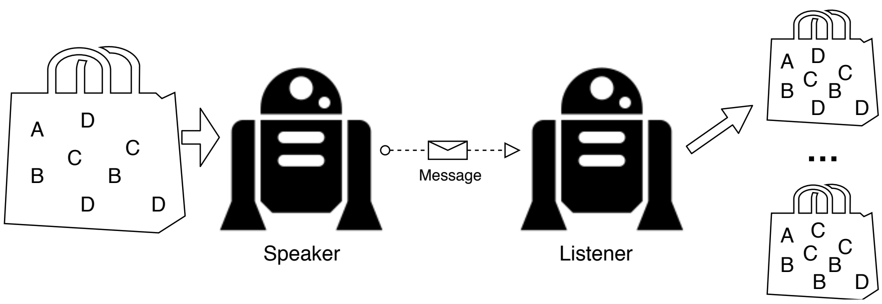

Figure 3.2: Diagram of Referential Game Playing Procedure.

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

3.2 Proposed Models

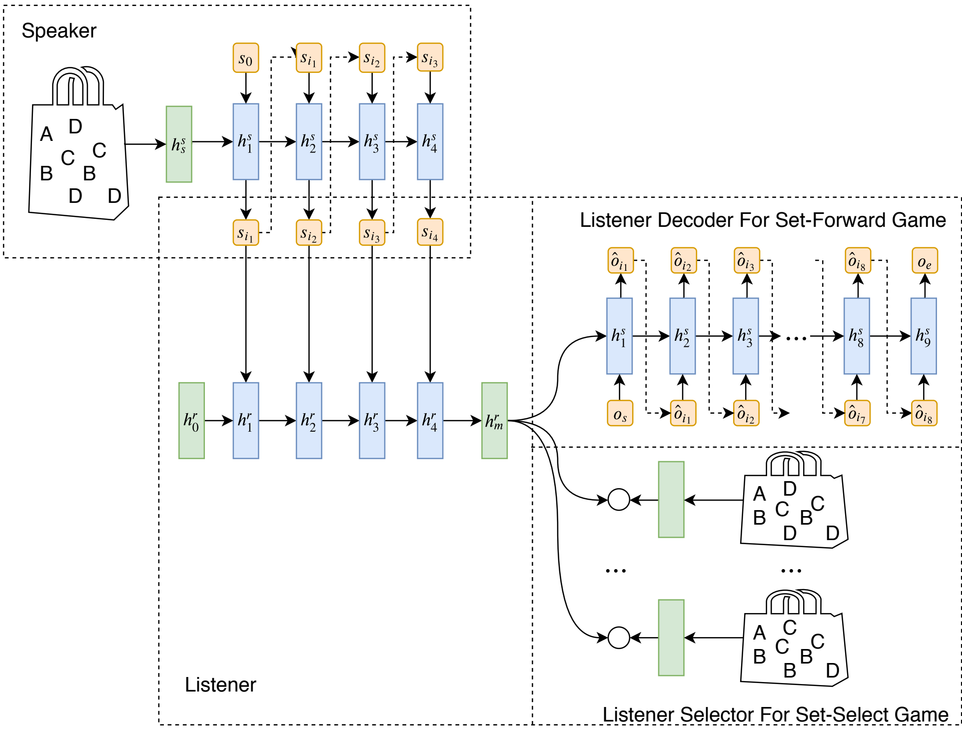

Figure 3.3: Overall Diagram of Model Architectures for Playing Games.

We illustrate the overall architecture of our models in Figure 3.3 given above, in which it is straightforward to see that a speaker S consists of a set encoder and a standard LSTM sequence decoder that can generate messages. As for a listeners L, it would first encode messages with a LSTM sequence encoder and get the feature vector h^l_m. Then, in the Set-Forward game, L would take h^l_m 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^l_m with a bunch of sets which are encoded by set encoders of L and select the one shown to S based on the dot product between h^l_m 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 would be introduced in the following subsubsection. As Seq-to-Seq model is quite popular nowadays, we skip details about how to generate sequences which correpond 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.

In our implementation, the number of attention operations 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 correponding 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 Listeners

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

Listener in Set-Forward Game: The listener in Set-Forward 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 as "Set2Seq2Seq".

Listener in Set-Select Game: The listener in Set-Select game would also first encoder messages with a LSTM like it is in standard Seq-to-Seq model. However, as it needs to select among a bunch of candidates, it also needs to encoder 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-Forward game, as the predictions of listeners are a sequence of objects $\hat{s}_i = \{\hat{o}_{i_1}, \ldots, \hat{o}_{i_n}\}$, we use cross-entropy between the original set and the predicted sequence as the objective function that needs to be minimised. Formally,

$$\mathcal{L}_{\theta^S,\theta^L}(o_{i_1}, \ldots, o_{i_n}) = \mathbb{E}_{m_i \sim p_{\theta^S}(\cdot|s_i)} \left[-\sum_{k=1}^{n} o_{i_k} \log(p(\hat{o}_{i_k}|m_i, \hat{o}_{-i_k})) \right] \tag{3.3}$$

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.

$$\mathcal{L}_{\theta^S,\theta^L}(s_i) = \mathbb{E}_{m_i \sim p_{\theta^S}(\cdot|s_i)} \left[-\sum_{k=1}^{C} s_i log(p(c_k)) \right] \tag{3.4}$$

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 cultual 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 transform the Bayesian iterated learning to so called Neural Iterated Learning(NIL).

Following the overall architecture of iterated learning, we also train agents generation by generation. In the begging 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 numbe of learning rounds 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_t^S and θ_t^L 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 correponding 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-Forward 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 tranmit representation vector h_s^s of set s_i to listener. And, listener compare h_s^s with all candiadate sets to make a selection.

3.3 Evaluation Methods

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. In order to calculate the topological compositionality, we need define the disctance 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 also use edit disctance following [Brighton and Kirby, 2006].

Meaningwhile, as we could perfectly ecode 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 langauge 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.

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

Chapter 4

Experiment Results and Discussion

4.1 Emergence of Language & Iterated Learning

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%	$\lvert M\rvert = 8$, $\lvert V\rvert = 10$, $\lvert O\rvert = 6$, $\lvert N_o\rvert = 10$
	REINFORCE	89.89%	
	SCST	98.67%	
Set2Seq2Choice	GUMBEL	100%	$\lvert M\rvert = 6$, $\lvert V\rvert = 10$, $\lvert O\rvert = 4$, $\lvert N_o\rvert = 10$
	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

4.3 Sample Complexity of Languages

4.4 Effects of Different Representations

4.5 Discussion

Chapter 5

Conclusions

5.1 Numeric Representations

The body of your dissertation, before the references and any appendices, *must* finish by page 40. The introduction, after preliminary material, should have started on page 1.

You may not change the dissertation format (e.g., reduce the font size, change the margins, or reduce the line spacing from the default 1.5 spacing). Over length or incorrectly-formatted dissertations will not be accepted and you would have to modify your dissertation and resubmit. You cannot assume we will check your submission before the final deadline and if it requires resubmission after the deadline to conform to the page and style requirements you will be subject to the usual late penalties based on your final submission time.

5.2 Future Works

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.

[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.

[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.

[MacWhinney, 2013] MacWhinney, B. (2013). *The emergence of language*. Psychology Press.

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

[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.

[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*.