ReferentialGym: A Nomenclature and Framework for Language Emergence & Grounding in (Visual) Referential Games

Kevin Denamganaï and James Alfred Walker
Department of Computer Science
University of York
York, UK
kyd500@york.ac.uk, james.walker@york.ac.uk

Abstract

Natural languages are powerful tools wielded by human beings to communicate information and co-operate towards common goals. Their values lie in some main properties like compositionality, hierarchy and recurrent syntax, which computational linguists have been researching the emergence of in artificial languages induced by language games. Only relatively recently, the AI community has started to investigate language emergence and grounding working towards better human-machine interfaces. For instance, interactive/conversational AI assistants that are able to relate their vision to the ongoing conversation.

This paper provides two contributions to this research field. Firstly, a nomenclature is proposed to understand the main initiatives in studying language emergence and grounding, accounting for the variations in assumptions and constraints. Secondly, a PyTorch based deep learning framework is introduced, entitled ReferentialGym, which is dedicated to furthering the exploration of language emergence and grounding. By providing baseline implementations of major algorithms and metrics, in addition to many different features and approaches, ReferentialGym attempts to ease the entry barrier to the field and provide the community with common implementations.

1 Introduction

Natural languages, whose evolution is supported by a culture of individuals that speak them, are a cornerstone of our ability to communicate and co-operate among each other, with different levels of expressivity and/or conciseness. They also act as a media through which we build our own learned representation of the world, as we experience it through our other modalities, e.g. vision. Although very little is known about how they came to be such a useful media, it is recognised that their values lie in some of their main properties: compositionality, hierarchical and/or recurrent syntax. They contribute to our ability to express an infinity of meanings while only using a finite amount of symbols, i.e. words and letters. Computational linguists have been researching the emergence of these properties in artificial languages induced by language games [20, 39, 9, 10, 40, 33, 61, 41] to better understand the evolution of natural languages. It is only relatively recently that it has also been investigated within the context of deep learning [46, 29, 45, 25, 47, 7, 19, 43, 21, 15, 16, 49, 60, 27, 50, 2, 17, 3], as the ability to ground into other modalities a natural-like language is thought to be a prerequisite for general AI [69, 54, 4, 18, 3].

In this paper, our first contribution is to propose a nomenclature to make sense of all the surveyed initiatives, which in spite of their great variations in assumptions and constraints, can be understood

4th NeurIPS Workshop on Emergent Communication (EmeCom @ NeurIPS 2020), Vancouver, Canada.

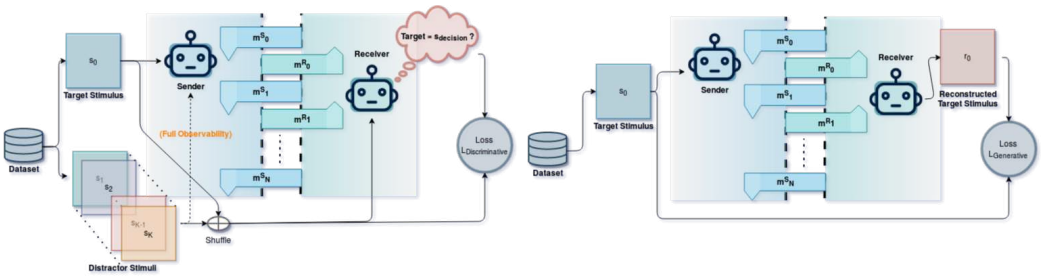

Figure 1: **Left:** Illustration of a partially-observable 2-players / $K+1$-stimuli / L-signal / N-round / uniformly-distributed-distractors / stimulus-focused variant of a referential game. **Right:** Illustration of a *generative* 2*-players* / L*-signal* / N*-round* variant of a *referential game*.

under the umbrella of *referential games* [48]. The nomenclature is presented in Section 2. Secondly, we introduce a deep learning framework entitled ReferentialGym, which is dedicated to further the exploration of language emergence and grounding in the visual modality, and their apparent properties and impact on learned representations, while providing a common implementation for all the features and constraints highlighted in the nomenclature. It is based on PyTorch [56] and is available under the MIT license[1].

The main features of ReferentialGym are:

1. It provides a *common implementation for many features and constraints of the main referential game variants* that can be found in the surveyed literature to date, thus allowing for a more systematic comparison between approaches.
2. Optimisation of the communication-channel via either *REINFORCE-like algorithms* [68], *(Straight-Through) Gumbel-Softmax estimator* [51, 34] following the work of Havrylov and Titov [29], or baseline and experimental variants of the *Obverter approach* [19, 6, 43].
3. It provides implementations of the main metrics in language emergence: *topographic similarity* [10] to evaluate language compositionality, *ambiguity* to evaluate language productivity, *instantaneous coordination* [35] to evaluate communication efficiency between the agents. Also, the main metrics in representation learning are provided with a primary focus on *disentanglement metrics*(e.g. [38]), in order to investigate the impact of language emergence and grounding onto the learned representations of the visual modality.
4. Modular graph-oriented design of the referential games allows the user to implement their own dedicated modules, for further inferences or metrics, and easily interface them with the base referential game graph.
5. It provides out-of-the-box data management strategies for classification datasets to be experimented with in a referential game. The current framework has implemented strategies to accommodate the design principles implemented by PyTorch with regards to data management and loading, thus doing a lot of the initial heavy-lifting for the user.
6. It provides a multi-task and transfer-learning-oriented framework where language emergence and grounding (via user-defined referential game variants) can be leveraged towards visual representation learning for other user-defined classification/regression tasks. Online training with multiple task losses are weighted via a *(homoscedastic) uncertainty learning module* [36].

2 Referential Games Nomenclature

In this section, a nomenclature is proposed to make sense of the main referential game variants that have spawned in the deep learning literature recently, and have explored the problem of efficiently communicating a meaning over a (limited) communication channel, without a prior grounding of the

[1]ReferentialGym can be downloaded at: `https://github.com/Near32/ReferentialGym`

symbols used in the communication channel.

The first instance of an environment that demonstrated a primary focus on the objective of communicating efficiently is the *signaling game* or *referential game* by Lewis [48], where a *sender/speaker* is asked to send a *signal/message* to the *receiver/listener*, based on the *state/stimulus* of the world that it observed. The *receiver/listener* then acts upon the observation of the *signal/message* by choosing one of the *actions* available to it. Both players goals are aligned (it features *pure coordination/common interests*), with the aim of performing the 'best' *action* given the observed *state*, where the notion of 'best' *action* is defined by the goal/interests common to both players.

Variants of this game have been driving a lot of research on language emergence and communication-based co-operation in the field of linguistics [11, 14, 62, 63], game theory[23, 26, 5, 22] (as acknowledged in [46]), and more recently, deep learning [46, 29, 45, 25, 47, 7, 19, 43, 21, 15, 16, 49, 27, 17, 59]. We will focus specifically on those variants that fit in the denomination of *referential games*. An instance of the kind of referential game that we are concerned with can be found in the work of Lazaridou et al. [46]. Under the nomenclature presented in this paper, Lazaridou et al. [46] would be featuring a *discriminative fully-observable /* 2*-players /* 1*-signal /* 0*-round / uniformly-distributed-distractors / stimulus-focused* variant, which they described as:

1. "There is a set of images represented by vectors $i_1, ..., i_N$, two images are drawn at random from this set, call them (i_L, i_R), one of them is chosen to be the 'target' $t \in L, R$
2. There are two players, a *sender* and a *receiver*, each seeing the images [–thus the adjective *fully-observable*, as opposed to when the *sender* would only see the 'target' stimulus–]- the *sender* receives input $\theta_S(i_L, i_R, t)$.
3. There is a vocabulary V of size [$|V|$] and the *sender* chooses one symbol to send to the receiver [–thus the adjective 1*-signalled*–], we call this the *sender*'s policy $s(\theta_S(i_L, i_R, t)) \in V$.
4. The *receiver* does not know the target, but sees the *sender*'s symbol and tries to guess the target image. We call this the *receiver*'s policy $r(i_L, i_R, s(\theta_S(i_L, i_R, t))) \in L, R$.
5. If $r(i_L, i_R, s(\theta_S(i_L, i_R, t)) = t$, that is, if the *receiver* guesses the target, both players receive a payoff of 1 (win), otherwise they receive a payoff of 0 (lose)."

Lazaridou et al. [46] shows, firstly, that fairly simple convolutional neural networks (CNNs) can learn to coordinate via a 1*-signal* communication protocol that is learned from scratch. Secondly, Lazaridou et al. [46] shows that "the meanings agents come to assign to symbols in this setup capture general conceptual properties of the objects depicted in the image, rather than low-level visual properties". In other words, the relationship between the meaning/stimulus space (i.e. the pixel space) and the signal/message space (similar in shape, here, to a finite set of integers of size $|V|$) is relying strongly on the fact that CNNs are known to "capture high-level visual properties of objects"([46], citing [70]).

Another variant can be seen in the work of Havrylov and Titov [29], which would be tackling a *discriminative partially-observable /* 2*-players /* L*-signal /* 0*-round / uniformly-distributed-distractors / stimulus-focused* variant, where the main results are presented with a number of distractors, $K = 127$, and a maximum sentence length, $L = 14$, and is described by Havrylov and Titov [29] as:

1. "There is a collection of images $\{i_n\}_{n=1}^N$ from which a target image t is sampled as well as K distracting images $\{d_k\}_{k=1}^K$.
2. There are two agents: a *sender* S_ϕ and a *receiver* R_θ.
3. After seeing the target image t, the *sender* has to come up with a message m_t, which is represented by a sequence of symbols from the vocabulary V of a size $|V|$. The maximum possible length of a sequence is L.
4. Given the message m_t and a set of images, which consists of distracting images and the target image, the goal of the *receiver* is to identify the target image correctly."

Figure 1 illustrates this main variant, which introduced the possibility of using a variable-length communication protocol, in addition to the reliance on the (Straight-Through) Gumbel-Softmax estimator approach (as opposed to the more common approach based on REINFORCE-like algorithms [68]).

Following this primer, outlining the kind of language game this paper focuses on, the following sections present our tentative nomenclature by describing the main features or dimensions of variations that have been investigated to date in the literature, in order to extend the most basic *referential game*.

2.1 Full vs. Partial Observability

This feature characterises whether the stimuli that are presented to the *sender/speaker* consist of all the stimuli experienced by the *receiver/listener* or solely of the target stimulus. The basic *referential game* has *full observability*, and thus it allows the *sender/speaker* to reason pragmatically, for instance. The orange arrow on Figure 1 highlights additional information available when the *sender/speaker* has full observability.

2.2 Multi-Players

The basic *referential game* consists of only 2 players, one *sender/speaker* and one *receiver/listener*. Yet, evolutionary and computational linguistics have shown that cultural transmission (i.e. from one generation of speaker/listener agents to another) plays a major role in the emergence of properties, such as compositionality [40, 61, 41] or recurrent syntax [39, 42]. The work of Cogswell et al. [21], Guo et al. [27], and, most recently, Ren et al. [59] illustrate implementations of implicit and/or explicit cultural transmission, where there exists at least one player/agent in each role and they may also be replaced (reset) according to different strategies. It is shown that even on the deep learning substrate, cultural transmission enhances the compositionality of the emerging language, as measured by a high performance accuracy in a referential game with novel combinations of known stimulus components, and an increase of topographic similarity between the meaning/stimulus space and the signal/message space throughout training.

2.3 Variable-length Communication

This feature characterises the ability from the *sender/speaker* to send/utter more than one *symbol/signal* to the *receiver/listener*, up to a maximal possible length, L, for the sequence of symbols. The basic *referential game* is 1-*signalled*. It is first introduced by Havrylov and Titov [29] and is quickly adopted by the research community as standard, independently of what approach is favoured to support the communication channel [47, 19].

2.4 Multi-Round Communication

This feature characterises whether the *receiver/listener* can send a *signal/message* back to the *sender/speaker* and how many communication rounds can be expected before the *receiver/listener* is finally tasked to act (this is described further in Section 2.6). The basic *referential game* is 0-*rounded*, which means the *receiver/listener* is not allowed to send *any signal/message*, which could probably be understood as queries, back to the *sender/speaker*. It only observes one *signal/message* and thus disambiguation is not possible, and the *receiver/listener* must make a correct decision on this one and only *signal/message*.

2.5 Multi-Modality

The basic *referential game* is not multi-modal in the sense that both the *sender/speaker* and *receiver/listener* experience stimuli through the same modality (usually either raw pixel inputs or symbolic (one-hot encoded) inputs). On the other hand, Evtimova et al. [25] featured different modalities, one for each role, such that the *receiver/listener* (referred to as the *questioner*) is experiencing raw pixel inputs while the *sender/speaker* (referred to as the *answerer*) is experiencing textual descriptions (in natural language). This work instantiates a *discriminative multi-modal / partially-observable /* 2*-player / L-signal / R-round / all-distractors / stimulus-focused* variant of a

referential game.

2.6 Discriminative vs. Generative

As mentioned earlier in Section 2.4, after all the communication rounds have been performed, the *receiver/listener* is tasked with performing an action. Depending on the type of referential game, the action can take two main forms:

- **Discriminative** - In this form, the agent has to discriminate between a set of stimuli, comprised of the target stimulus observed by the *sender/speaker* and some additional distractor stimuli, and find the target.
- **Generative** - In this form, the agent has to generate an output, which can for instance, be the task of reconstructing the target stimulus itself or some of its (symbolic) attributes [45, 17].

While the generative form is rather simplistic, in terms of the number of moving parts, as illustrated in Figure 1, it is not the case for the discriminative form. The following subsections describe the different refinements that can be found for the discriminative form in the literature.

2.6.1 Distractor Stimuli Distribution

This feature characterises the kind of distribution from which the distractor stimuli are sampled and presented to the *receiver/listener*. The basic *referential game* has distractors *uniformly-distributed*. Lazaridou et al. [47] investigated the effect of sampling distractors from a "target-specific context distribution" that emphasizes how common objects actually co-occur in the real world, thus making "the target *goat* more likely being mixed with *sheep* and *cow* as distractors rather than *bike* or *eggplant*". They reported that, in this non-uniform case, the game is made significantly more difficult, reporting lower data-efficiency, as similar stimuli co-occur more often. More importantly, this feature influenced the "organisation"/structure of the emerging languages. Lazaridou et al. [47] opens an avenue to explore the naturalness of emerging languages, as it is a growing concern in the community [45, 15, 16, 3], in view of better human-machine interface.

2.6.2 Descriptive-only

The basic (discriminative) *referential game* allows the *receiver/listener* to perform pragmatic reasoning on its stimuli, since it experiences at least two stimuli, the target stimulus and at least one distractor stimulus. In this variant, the *receiver/listener* only experiences one stimulus that *may or may not be* the target stimulus, and it is tasked to output whether the stimulus it has experienced is the same as the *sender/speaker* has. Only the descriptive-part of the *referential game* is emphasised here. It is argued that the work of Choi et al. [19] deals with such a variant, described as the *Two-Person Image Description Game*. The effect of this setting has not yet been compared to other *referential game* variants.

2.6.3 Stimulus vs. Object Focused

The basic (discriminative) *referential game* is stimulus-focused, which assumes that both agents would be somehow embodied in the same body, and they are tasked to discriminate between given stimuli. On the other hand, the object-focused variant incorporates the issues that stem from the difference of embodiment. The agents are tasked with discriminating between objects (or scenes) independently of the viewpoint from which they may experience it. In this variant, the game is more about bridging the gap between each other's cognition rather than (just) finding a common language. It is solely featured in the work of Choi et al. [19], in its descriptive-only form. Needless to say that the object-focused variant adds a considerable degree of difficulty to the task. It has been highlighted that embodiment may hold some key to the systematic generalisation abilities of the learning agent [31], and therefore it is highlighted as a very important research direction to pursue.

Choi et al. [19] shows that the obverter technique enforces great concept alignment between the two agents (to which extent do the (visual) sensory features align from one agent to another agent, when projected into a similar (linguistic) space?), here aligning the languages spoken by both agents

and also aligning the way modalities are internally represented. Interesting research directions to highlight dwell in (i) the evaluation of the efficiency of the other agents architecture in this object-focused setting, and (ii) the agent-to-agent concept/feature/modality representation alignment.

In an even more abstract approach, the object-focused setting could be acknowledged as an emphasis on the concept or semantic meaning behind the observed stimulus, and the *receiver/listener* would thus be tasked with learning the semantic, while being prompted with different instances of it. As detailed further in Section 3, this viewpoint opens up numerous applications of referential games for classification tasks.

2.7 Limitations

Finally, with regards to some other important examples, it can be argued that the works of Das et al. [24], Kottur et al. [45], Cogswell et al. [21] also feature *generative multi-modal / partially-observable / $2/N$-players / $1/L$-signal / R-round* variants, referred to as the *Task & Talk Game* or as a *Goal-Driven Neural Dialog*. Indeed, the modality of the *receiver/listener* (*questioner*) is different from that of the *sender/speaker* (*answerer)*, since the latter experiences a one-hot-encoded vector describing the task to solve, i.e. to find the answer to an attribute-focused question about the stimulus experienced by the *sender/speaker*. For example, the question can concern the value instantiated in the *shape*, *color*, or *style* attribute of said stimulus, that can represent for instance a *dashed green circle*. It is important to note that although these games are not explicitly implemented as referential games, they could be instantiated as a generative referential game variant.

Furthermore, although an explicit difference is made between the generative and discriminative forms, it can be noticed that the generative form is implicitly instantiating a discriminative form, where the *receiver/listener* modality consists of the space of all possible stimuli. For instance, in the case of the *Task & Talk Game*, this space consists of all the possible values permitted on each attribute axis. It ensues that the latter agent would be tasked to choose the value of the queried attribute that is instantiated in the stimulus observed by the *sender/speaker*, among the whole collection of values that can be instantiated for each queryable attribute of the stimulus. In the nomenclature presented in this paper, the decision was made to make an explicit difference between the two forms. Each form entails a different practical framing of the task at the level of the agent, which is known to have an impact on the agent's performance at any given task. This has given rise to sub-fields in deep learning literature, e.g. the discriminative form is close to adversarial deep learning, whilst the generative form is close to auto-regressive/generative deep learning.

3 ReferentialGym Architecture

Based on the previously-detailed nomenclature in Section 2, ReferentialGym was developed to provide a common framework for improving comparisons between architectures when investigating the impact of features on emerging languages and their properties. ReferentialGym aims at providing a coherent framework where such comparisons can be undertaken quickly, easily and fairly, as much of the heavy-lifting has already been done, so users can focus on prototyping and experimenting with different architectures. In the remainder of this section, a walkthrough is presented of the architecture and design principles at the heart of the ReferentialGym framework.

The features identified by the nomenclature are associated to some hyperparameter entrypoints that the user is asked to set when defining the kind of referential game to instantiate. It takes the form of an instance of the class `ReferentialGame` that handles the training and testing phases, in multiple epochs (with a different time granularity within each epoch), on a data set of the user's choosing. The only limitation is that this data set ought to be provided as an instance of the PyTorch `torch.utils.data.dataset` class. It is then internally wrapped multiple times to accommodate sampling the stimuli of the target and distractors, which depends on the form of the game the user has defined, among the many possibilities highlighted in the nomenclature defined in Section 2, when setting the hyperparameters of the `ReferentialGame` instance.

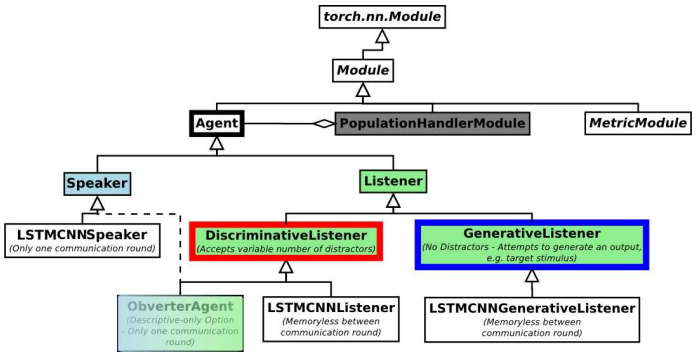

Figure 2: Hierarchy of referential game agents implemented in ReferentialGym and their relationship to the more general `Module` class.

It is important to note that the different internal wrappers are handling all the data management strategies, in order to allow any image classification data set to be used for experiments in most of the referential game variants. Notably, different distractor sampling strategies are made available with respect to the class of each stimulus in the classification data set. Object-focus can also be enforced in the form of a focus on the semantic or the class of each image, as images of the same class can be seen as a different viewpoint or instance of the same semantic scene/object (as we previously highlighted in Section 2.6.3).

3.1 Data Streaming-Oriented Design, Logging & Metrics

The framework is built with modularity and re-usability in mind, therefore it relies on modular building blocks interconnected in acyclic graphs, where each building blocks inherits from a `Module` class (itself inherited from PyTorch's `torch.nn.Module` class) that requires the definition of input data streams on which to operate. The output data streams produced by each `Module` instance are then handled by an instance of the `StreamHandler` class that is in charge of serving each module with their desired input data streams.

To the user, all of this is hidden and happens seemingly in the background, similarly to some statistics logging and agent management processes (e.g. saving, loading, etc). The whole framework implements logging mechanisms relying on TensorboardX [32], which are then visualizable in Tensorboard [52].

In addition to statistics logging, the framework provides implementations of major metrics in the field of language emergence, accounting for both *positive signalling* and *positive listening* metrics, following the work of Lowe et al. [50]. These include: *topographic similarity* [10], in the form of the `TopographicSimilarityMetricModule` class; *ambiguity* in the emerging language, in the form of a hook function attached to each `Speaker` module (see Section 3.2 for more details about hooking mechanisms); and *instantaneous coordination* [35], in the form of the `InstantaneousCoordinationMetric`
`Module` class. Moreover, as the framework lies at the interface between language emergence and language grounding in the visual modality, it aims to provide the means to evaluate the learned representations at different levels, not only the level of language, but also the level of the visual modality. In this initial release, this is achieved by providing implementations of *disentanglement metrics*, in the form of the `FactorVAEDisentanglementMetric`
`Module` that follows the metric proposed by Kim and Mnih [38].

Finally, each acyclic graph is defined as a pipeline, or ordered list of modules that the `StreamHandler` instance serves at each timestep. Pipelines are provided by the user as ordered list of module identifications.

3.2 Agent Hierarchy

Everything in the framework inherits from the `Module` class. This is evidently the case of referential game agents whose inheritance hierarchy is illustrated in Figure 2. Following the detailed nomenclature in Section 2, while the *sender/speaker* is invariant to the form that the game takes (and thus is always instantiated in the `Speaker` class), the *receiver/listener* computation obviously varies. Playing in the discriminative form requires the instantiation of the *receiver/listener* in the `DiscriminativeListener` class hierarchy, whereas playing in the generative form requires its instantiation to be in the `GenerativeListener` class hierarchy.

In more detail, classes that inherit from the `Agent` class come with a hooking mechanism that enables the user to augment the functionality of each pre-defined agent, by defining and registering hook functions that would operate on the output data stream of the agents, before the `StreamHandler` sees them. This hooking mechanism is primarily used in the background to define different loss functions that accommodate the different referential game variants, and different logging mechanisms that may, for instance, be specific to the agent type (e.g. ambiguity metric specific to `Speaker` agents, as mentioned earlier).

The framework also implements a very powerful abstraction, in which modules can also be considered as possible input data streams, thereby allowing some modules to operate on some other modules. The main purpose of this abstraction is to enable multi-player variants of the game, in addition to different management approaches to the culture of agents if the user chooses so, via an instance of the `PopulationHandlerModule` class, whose output streams are placeholder agents that are subsequently playing a round of the referential game.

4 Related & Future Works

Language Grounding - The proposed framework is inspired by the possibilities offered by the PyTorchPipe (PTP) framework [44] and aims to provide a similar tool towards investigating artificial language emergence and, in latter releases, investigating translation between natural and artificial languages. ReferentialGym and PTP both operate in the language grounding subfield, among others, and share some design principles, as they are both focusing on the modularity, and therefore re-usability of each component, which can be arranged by the user.

Language Emergence - In the current release, ReferentialGym is grounded in a broader range of features found in the relevant literature than the EGG framework [37], albeit substantially less mature. ReferentialGym is mainly focused with visual stimuli, while the EGG framework remains rather general on that side. Going forward, ReferentialGym will focus on dynamic stimuli, in the form of videos, and plans to accommodate video data sets in order to investigate language emergence and grounding over transformations or time-sensitive data/stimuli.

Disentanglement & Compositionality - In the current release, ReferentialGym provides modular implementations of disentanglement and compositionality metrics, in addition to state-of-the-art autoregressive inference modules, such as: β-VAE and variants [30, 12], FactorVAE [38], and MONet [13]. Integration of relevant datasets to study the relationship between disentanglement in the learned representations and compositionality in the emerging languages are well underway, starting with the dSprites [30, 53] data set.

5 Conclusion

This paper provides two main contributions to the research community. Firstly, a nomenclature is proposed for the different language games that have spawned under the umbrella of referential games with the goal of studying language emergence and grounding. The many features and varying constraints that the literature exhibits have been discussed, and their relevance towards the development of language emergence and grounding abilities, as prerequisites for general AI, have been highlighted. Secondly, this paper introduces ReferentialGym, a deep learning framework that implements the main features of the proposed nomenclature. ReferentialGym is based on PyTorch[56]

and dedicated to further the exploration of language emergence and grounding in the visual modality, by providing baseline implementations of major algorithms and metrics of the surveyed literature. It is hoped that this work will ease the entry barrier to the field and enable the community to perform more thorough and fair comparisons, thanks to the collection of common implementations.

Broader Impact

This work consists solely of simulations, thus evacuating some of the ethical concerns, as well as the concerns with regards to the consequences of failure of the system presented. With regards to the ethical aspects related to its inclusion in the field of Artificial Intelligence, we argue that our work aims to have positive outcomes on the development of human-machine interfaces, albeit being not yet mature enough to aim for this goal. The current state of our work does not allow us to extrapolate towards negative outcomes.

This work should benefit the research community of language emergence and grounding, in its current state.

Acknowledgments and Disclosure of Funding

This work was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/L015846/1].

We would like to thank the anonymous reviewers for their constructive feedback on the draft of this paper.

We gratefully acknowledge the use of Python[65], IPython[58], SciPy[66], Scikit-learn[57], Scikit-image[64], NumPy[28], Pandas[67, 55], OpenCV[8], PyTorch[56], TensorboardX[32], and Tensorboard from the Tensorflow ecosystem[1], without which this work would not be possible.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] M. Baroni. Linguistic generalization and compositionality in modern artificial neural networks. mar 2019. URL `http://arxiv.org/abs/1904.00157`.

[3] M. Baroni. Rat big, cat eaten! Ideas for a useful deep-agent protolanguage. mar 2020. URL `http://arxiv.org/abs/2003.11922`.

[4] M. Baroni, A. Joulin, A. Jabri, G. Kruszewski, A. Lazaridou, K. Simonic, and T. Mikolov. CommAI: Evaluating the first steps towards a useful general AI. jan 2017. URL `http://arxiv.org/abs/1701.08954`.

[5] A. Blume, D. V. DeJong, Y.-G. Kim, and G. B. Sprinkle. Experimental evidence on the evolution of meaning of messages in sender-receiver games. *The American Economic Review*, 88(5): 1323–1340, 1998.

[6] B. Bogin, M. Geva, and J. Berant. Emergence of Communication in an Interactive World with Consistent Speakers. sep 2018. URL `http://arxiv.org/abs/1809.00549`.

[7] D. Bouchacourt and M. Baroni. How agents see things: On visual representations in an emergent language game. aug 2018. URL `http://arxiv.org/abs/1808.10696`.

[8] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[9] H. Brighton and S. Kirby. The survival of the smallest: Stability conditions for the cultural evolution of compositional language. In *European Conference on Artificial Life*, pages 592–601. Springer, 2001.

[10] H. Brighton and S. Kirby. Understanding Linguistic Evolution by Visualizing the Emergence of Topographic Mappings. *Artificial Life*, 12(2):229–242, jan 2006. ISSN 1064-5462. doi: 10.1162/artl.2006.12.2.229. URL http://www.mitpressjournals.org/doi/10.1162/artl.2006.12.2.229.

[11] T. Briscoe. *Linguistic evolution through language acquisition*. Cambridge University Press, 2002.

[12] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-VAE. apr 2018. URL http://arxiv.org/abs/1804.03599.

[13] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

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

[15] R. Chaabouni, E. Kharitonov, E. Dupoux, and M. Baroni. Anti-efficient encoding in emergent communication. *NeurIPS*, may 2019. URL http://arxiv.org/abs/1905.12561.

[16] R. Chaabouni, E. Kharitonov, A. Lazaric, E. Dupoux, and M. Baroni. Word-order biases in deep-agent emergent communication. may 2019. URL http://arxiv.org/abs/1905.12330.

[17] R. Chaabouni, E. Kharitonov, D. Bouchacourt, E. Dupoux, and M. Baroni. Compositionality and Generalization in Emergent Languages. apr 2020. URL http://arxiv.org/abs/2004.09124.

[18] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. oct 2018. URL http://arxiv.org/abs/1810.08272.

[19] E. Choi, A. Lazaridou, and N. de Freitas. Compositional Obverter Communication Learning From Raw Visual Input. apr 2018. URL http://arxiv.org/abs/1804.02341.

[20] N. A. Chomsky. Reflections on language. 1976.

[21] M. Cogswell, J. Lu, S. Lee, D. Parikh, and D. Batra. Emergence of Compositional Language with Deep Generational Transmission. apr 2019. URL http://arxiv.org/abs/1904.09067.

[22] V. Crawford. A survey of experiments on communication via cheap talk. *Journal of Economic theory*, 78(2):286–298, 1998.

[23] V. P. Crawford and J. Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451, 1982.

[24] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. mar 2017. URL http://arxiv.org/abs/1703.06585.

[25] K. Evtimova, A. Drozdov, D. Kiela, and K. Cho. Emergent Communication in a Multi-Modal, Multi-Step Referential Game. may 2017. URL http://arxiv.org/abs/1705.10369.

[26] J. Farrell and M. Rabin. Cheap talk. *Journal of Economic perspectives*, 10(3):103–118, 1996.

[27] S. Guo, Y. Ren, S. Havrylov, S. Frank, I. Titov, and K. Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

[28] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[29] S. Havrylov and I. Titov. Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. may 2017. URL http://arxiv.org/abs/1705.11192.

[30] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2 (5):6, 2017.

[31] F. Hill, A. Lampinen, R. Schneider, S. Clark, M. Botvinick, J. L. McClelland, and A. Santoro. Emergent Systematic Generalization in a Situated Agent. oct 2019. URL http://arxiv.org/abs/1910.00571.

[32] T.-W. Huang. Tensorboardx, 2018. URL https://github.com/lanpa/tensorboardX.

[33] J. R. Hurford. Language and number: The emergence of a cognitive system. 1987.

[34] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[35] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *arXiv preprint arXiv:1810.08647*, 2018.

[36] A. Kendall, Y. Gal, and R. Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. may 2017. URL http://arxiv.org/abs/1705.07115.

[37] E. Kharitonov, R. Chaabouni, D. Bouchacourt, and M. Baroni. Egg: a toolkit for research on emergence of language in games. *arXiv preprint arXiv:1907.00852*, 2019.

[38] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[39] S. Kirby. Learning, bottlenecks and the evolution of recursive syntax. 1998. URL https://pdfs.semanticscholar.org/0def/76e16d1becfbb60e9dad80105926298e9686.pdf.

[40] S. Kirby and J. R. Hurford. The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model. In *Simulating the Evolution of Language*, pages 121–147. Springer London, 2002. doi: 10.1007/978-1-4471-0663-0_6.

[41] S. Kirby, T. Griffiths, and K. Smith. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014. URL https://www.sciencedirect.com/science/article/pii/S0959438814001421.

[42] S. Kirby, M. Tamariz, H. Cornish, and K. Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, aug 2015. ISSN 0010-0277. doi: 10.1016/J.COGNITION.2015.03.016. URL https://www.sciencedirect.com/science/article/pii/S0010027715000815.

[43] T. Korbak, J. Zubek, Ł. Kuciński, P. Miłoś, and J. Rączaszek-Leonardi. Developmentally motivated emergence of compositional communication via template transfer. oct 2019. URL http://arxiv.org/abs/1910.06079.

[44] T. Kornuta. Pytorchpipe: a framework for rapid prototyping of pipelines combining language and vision. *arXiv preprint arXiv:1910.08654*, 2019.

[45] S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. jun 2017. URL http://arxiv.org/abs/1706.08502.

[46] A. Lazaridou, A. Peysakhovich, and M. Baroni. Multi-Agent Cooperation and the Emergence of (Natural) Language. dec 2016. URL http://arxiv.org/abs/1612.07182.

[47] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. apr 2018. URL http://arxiv.org/abs/1804.03984.

[48] D. Lewis. Convention: A philosophical study. 1969.

[49] F. Li and M. Bowling. Ease-of-Teaching and Language Structure from Emergent Communication. jun 2019. URL http://arxiv.org/abs/1906.02403.

[50] R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, and Y. Dauphin. On the Pitfalls of Measuring Emergent Communication. mar 2019. URL http://arxiv.org/abs/1903.05168.

[51] C. J. Maddison, A. Mnih, and Y. W. Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*, 2017.

[52] D. Mané et al. Tensorboard: Tensorflow's visualization toolkit, 2015.

[53] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[54] T. Mikolov, A. Joulin, and M. Baroni. A Roadmap towards Machine Intelligence. nov 2015. URL http://arxiv.org/abs/1511.08130.

[55] T. pandas development team. pandas-dev/pandas: Pandas, Feb. 2020. URL https://doi.org/10.5281/zenodo.3509134.

[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

[58] F. Perez and B. E. Granger. Ipython: A system for interactive scientific computing. *Computing in Science Engineering*, 9(3):21–29, 2007. doi: 10.1109/MCSE.2007.53.

[59] Y. Ren, S. Guo, M. Labeau, S. B. Cohen, and S. Kirby. Compositional Languages Emerge in a Neural Iterated Learning Model. feb 2020. URL http://arxiv.org/abs/2002.01365.

[60] C. Resnick, A. Gupta, J. Foerster, A. M. Dai, and K. Cho. Capacity, Bandwidth, and Compositionality in Emergent Language Learning. oct 2019. URL http://arxiv.org/abs/1910.11424.

[61] K. Smith, S. Kirby, H. B. A. Life, and U. 2003. Iterated learning: A framework for the emergence of language. *Artificial Life*, 9(4):371–389, 2003. URL https://www.mitpressjournals.org/doi/abs/10.1162/106454603322694825.

[62] M. Spike, K. Stadler, S. Kirby, and K. Smith. Minimal requirements for the emergence of learned signaling. *Cognitive science*, 41(3):623–658, 2017.

[63] L. Steels and M. Loetzsch. The grounded naming game. *Experiments in cultural language evolution*, 3:41–59, 2012.

[64] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[65] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

[66] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[67] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

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

[69] T. Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, jan 1972. ISSN 0010-0285. doi: 10.1016/0010-0285(72)90002-3. URL `https://www.sciencedirect.com/science/article/pii/0010028572900023?via{%}3Dihub`.

[70] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.