Grounded Language Learning in a Simulated 3D World

Karl Moritz Hermann[*†], **Felix Hill**[*], **Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, Marcus Wainwright, Chris Apps, Demis Hassabis and Phil Blunsom**[†]

Deepmind
London, UK

Abstract

We are increasingly surrounded by artificially intelligent technology that takes decisions and executes actions on our behalf. This creates a pressing need for general means to communicate with, instruct and guide artificial agents, with human language the most compelling means for such communication. To achieve this in a scalable fashion, agents must be able to relate language to the world and to actions; that is, their understanding of language must be grounded and embodied. However, learning grounded language is a notoriously challenging problem in artificial intelligence research. Here we present an agent that learns to interpret language in a simulated 3D environment where it is rewarded for the successful execution of written instructions. Trained via a combination of reinforcement and unsupervised learning, and beginning with minimal prior knowledge, the agent learns to relate linguistic symbols to emergent perceptual representations of its physical surroundings and to pertinent sequences of actions. The agent's comprehension of language extends beyond its prior experience, enabling it to apply familiar language to unfamiliar situations and to interpret entirely novel instructions. Moreover, the speed with which this agent learns new words increases as its semantic knowledge grows. This facility for generalising and bootstrapping semantic knowledge indicates the potential of the present approach for reconciling ambiguous natural language with the complexity of the physical world.

1. Introduction

Endowing machines with the ability to relate language to the physical world is a long-standing challenge for the development of Artificial Intelligence. As situated intelligent technology becomes ubiquitous, the development of computational approaches to understanding grounded language has become critical to human-AI interaction. Beginning with Winograd (1972), early attempts to ground language understanding in a physical world were constrained by their reliance on the laborious hard-coding of linguistic and physical rules. Modern devices with voice control may appear more competent but suffer from the same limitation in that their language understanding components are mostly rule-based and do not generalise or scale beyond their programmed domains.

∗. These authors contributed equally to this work.
†. Corresponding authors: kmh@google.com and pblunsom@google.com.

This work presents a novel paradigm for simulating language learning and understanding. The approach differs from conventional computational language learning in that the learning and understanding take place with respect to a continuous, situated environment. Simultaneously, we go beyond rule-based approaches to situated language understanding as our paradigm requires agents to learn end-to-end the grounding for linguistic expressions in the context of using language to complete tasks given only pixel-level visual input.

The initial experiments presented in this paper take place in an extended version of the DeepMind Lab (Beattie et al., 2016) environment, where agents are tasked with finding and picking up objects based on a textual description of each task. While the paradigm outlined gives rise to a large number of possible learning tasks, even the simple setup of object retrieval presents challenges for conventional machine learning approaches. Critically, we find that language learning is contingent on a combination of reinforcement (reward-based) and unsupervised learning. By combining these techniques, our agents learn to connect words and phrases with emergent representations of the visible surroundings and embodied experience. We show that the semantic knowledge acquired during this process generalises both with respect to new situations and new language. Our agents exhibit zero-shot comprehension of novel instructions, and the speed at which they acquire new words accelerates as their semantic knowledge grows. Further, by employing a curriculum training regime, we train a single agent to execute phrasal instructions pertaining to multiple tasks requiring distinct action policies as well as lexical semantic and object knowledge.[1]

2. Related work

Learning semantic grounding without prior knowledge is notoriously difficult, given the limitless possible referents for each linguistic expression (Quine, 1960). A learner must discover correlations in a stream of low level inputs, relate these correlations to both its own actions and to linguistic expressions and retain these relationships in memory. Perhaps unsurprisingly, the few systems that attempt to learn language grounding in artificial agents do so with respect to environments that are far simpler than the continuous, noisy sensory experience encountered by humans (Steels, 2008; Roy and Pentland, 2002; Krening et al., 2016; Yu et al., 2017).

The idea of programming computers to understand how to relate language to a simulated physical environment was pioneered in the seminal work of Winograd (1972). His SHRDLU system was programmed to understand user generated language containing a small number of words and predicates, to execute corresponding actions or to ask questions requesting more information. While initially impressive, this system required that all of the syntax and semantics (in terms of the physical world) of each word be hard coded a priori, and thus it was unable to learn new concepts or actions. Such rule-based approaches to language understanding have come to be considered too brittle to scale to the full complexities of natural language. Since this early work, research on language grounding has taken place across a number of disciplines, primarily in robotics, computer vision and computational linguistics. Research in both natural language processing and computer vision has pointed to the importance of cross modal approaches to grounded concept learning. For instance, it was shown that learnt concept representation spaces more faithfully reflect human semantic

1. See `https://youtu.be/wJjdu1bPJ04` for a video of the trained agents.

intuitions if induced from information about the perceptible properties of objects as well as from raw text (Silberer and Lapata, 2012).

Semantic parsing, as pursued the field of natural language processing, has predominantly focussed on building a compositional mapping from natural language to formal semantic representations that are then grounded in a database or knowledge graph (Zettlemoyer and Collins, 2005; Berant et al., 2013). The focus of this direction of work is on the compositional mapping between the two abstract modalities, natural language and logical form, where the grounding is usually discrete and high level. This is in contrast to the work presented in this paper where we focus on learning to ground language in low level perception and actions.

Siskind (1995) represents an early attempt to ground language in perception by seeking to link objects and events in stick-figure animations to language. Broadly this can be seen as a precursor to more recent work on mapping language to actions in video and similar modalities (Siskind, 2001; Chen and Mooney, 2008; Yu and Siskind, 2013). In a similar vein, the work of Roy and Pentland (2002) applies machine learning to aspects of grounded language learning, connecting speech or text input with images, videos or even robotic controllers. These systems consisted of modular pipelines in which machine learning was used to optimise individual components while complementing hard-coded representations of the input data. Within robotics, there has been interest in using language to facilitate human-robot communication, as part of which it is necessary to devise mechanisms for grounding a perceptible environment with language (Hemachandra et al., 2014; Walter et al., 2014). In general, the amount of actual learning in these prior works is heavily constrained, either through the extensive use of hand-written grammars and mechanisms to support the grounding, or through simplification in terms of the setup and environment.

Other related work focuses on language grounding from the perspective of human-machine communication (Thomason et al., 2015; Wang et al., 2016; Arumugam et al., 2017). The key difference between these approaches and our work is that here again language is grounded to highly structured environments, as opposed to the continuous perceptible input our learning environment provides.

In the field of computer vision, image classification (Krizhevsky et al., 2012) can be interpreted as aligning visual data and semantic or lexical concepts. Moreover, neural networks can effectively map image or video representations from these classification networks to human-written image captions. These mappings can also yield plausible descriptions of visual scenes that were not observed during training (Xu et al., 2015; Vendrov et al., 2015). However, unlike our approach, these captioning models typically learn visual and linguistic processing and representation from fixed datasets as part of two separate, independent optimisations. Moreover, they do not model the grounding of linguistic symbols in actions or a visual stimuli that constantly change based on the exploration policy of the agent.

The idea that reinforcement-style learning could play a role in language learning has been considered for decades (Chomsky, 1959). Recently, however, RL agents controlled by deep neural nets have been trained to solve tasks in both 2D (Mnih et al., 2015) and 3D (Mnih et al., 2016) environments. Our language learning agents build on these approaches and algorithms, but with an agent architecture and auxiliary unsupervised objectives that are specific to our multi-modal learning task. Other recently-proposed frameworks for interactive language learning involve unimodal (text-only) settings (Narasimhan et al., 2015; Mikolov et al., 2015).

3. The 3D language learning environment

To conduct our language learning experiments we integrated a language channel into a 3D simulated world (DeepMind Lab, Beattie et al. (2016)). In this environment, an agent perceives its surroundings via a constant stream of continuous visual input and a textual instruction. It perceives the world actively, controlling what it sees via movement of its visual field and exploration of its surroundings. One can specify the general configuration of layouts and possible objects in this environment together with the form of language instructions that describe how the agent can obtain rewards. While the high-level configuration of these simulations is customisable, the precise world experienced by the agent is chosen at random from billions of possibilities, corresponding to different instantiations of objects, their colours, surface patterns, relative positions and the overall layout of the 3D world.

To illustrate this setup, consider a very simple environment comprising two connected rooms, each containing two objects. To train the agent to understand simple referring expressions, the environment could be configured to issue an instruction of the form pick the *X* in each episode. During training, the agent experiences multiple episodes with the shape, colour and pattern of the objects themselves differing in accordance with the instruction. Thus, when the instruction is pick the pink striped ladder, the environment might contain, in random positions, a pink striped ladder (with positive reward), an entirely pink ladder, a pink striped chair and a blue striped hairbrush (all with negative reward).

It is important to emphasise the complexity of the learning challenge faced by the agent, even for a simple reference task such as this. To obtain positive rewards across multiple training episodes, the agent must learn to efficiently explore the environment and inspect candidate objects (requiring the execution of hundreds of inter-dependent actions) while simultaneously learning the (compositional) meanings of multi-word expressions and how they pertain to visual features of different objects (Figure 1)

We also construct more complex tasks pertaining to other characteristics of human language understanding, such as the generalisation of linguistic predicates to novel objects, the productive composition of words and short phrases to interpret unfamiliar instructions and the grounding of language in relations and actions as well as concrete objects.

4. Agent design

Our agent consists of four inter-connected modules optimised as a single neural network. At each time step t, the visual input v_t is encoded by the convolutional *vision module* $\mathbf{V}$ and a recurrent (LSTM, Hochreiter and Schmidhuber (1997)) *language module* $\mathbf{L}$ encodes the instruction string l_t. A *mixing module* $\mathbf{M}$ determines how these signals are combined before they are passed to a two-layer LSTM *action module* $\mathbf{A}$. The hidden state s_t of the upper LSTM in $\mathbf{A}$ is fed to a policy function, which computes a probability distribution over possible motor actions $\pi(a_t|s_t)$, and a state-value function approximator $Val(s_t)$, which computes a scalar estimate of the agent value function for optimisation. To learn from the scalar rewards that can be issued by the environment, the agent employs an actor-critic algorithm (Mnih et al., 2016).

The policy π is a distribution over a discrete set of actions. The baseline function Val estimates the expected discounted future return following the state the agent is currently in. In

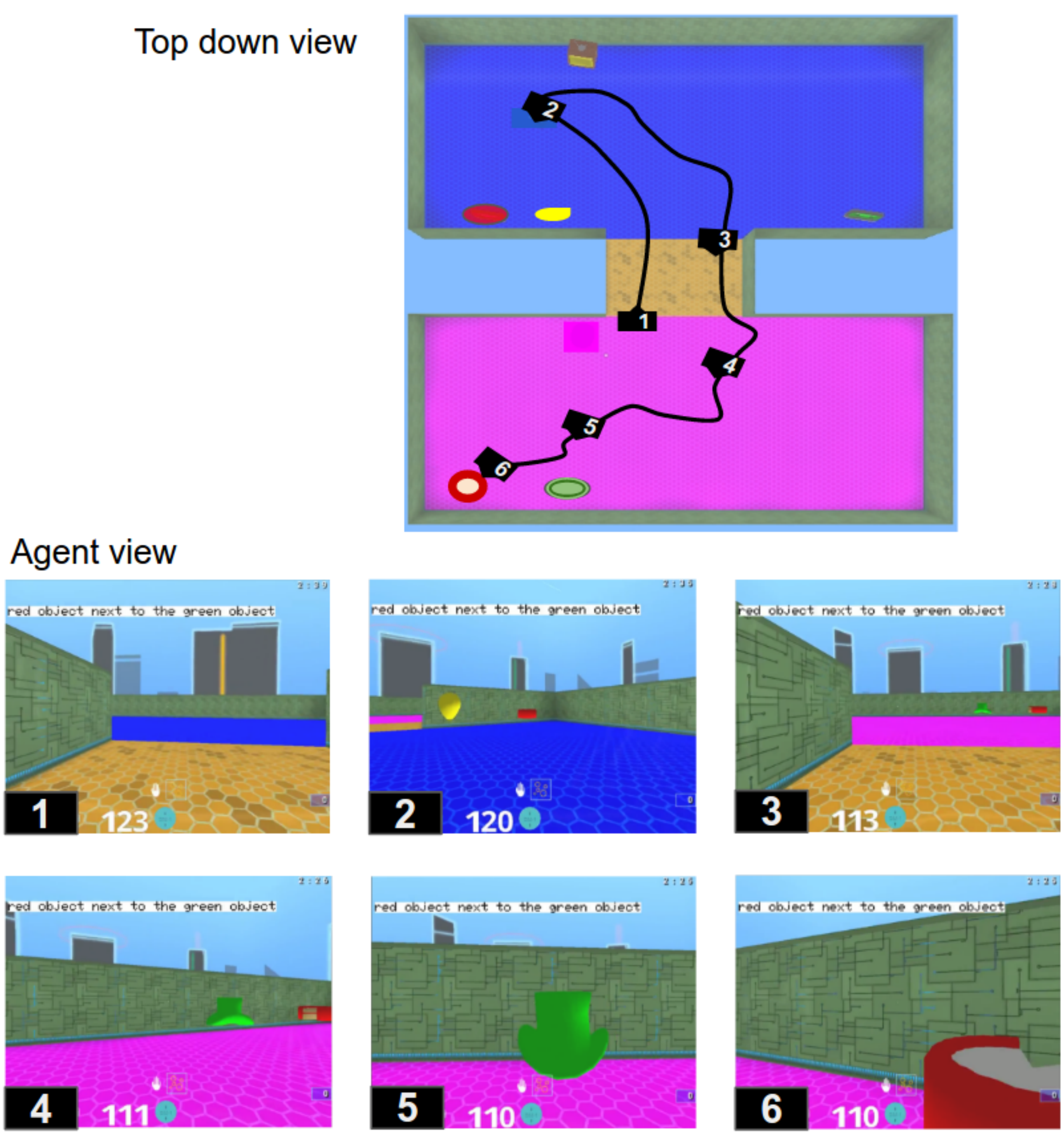

Figure 1: In this example, the agent begins in position **1** and immediately receives the instruction pick the red object next to the green object. It explores the two-room layout, viewing objects and their relative positions before retrieving the object that best conforms to the instruction. This exploration and selection behaviour emerges entirely from the reward-driven learning and is not preprogrammed. When training on a task such as this, there are billions of possible episodes that the agent can experience, containing different objects in different positions across different room layouts.

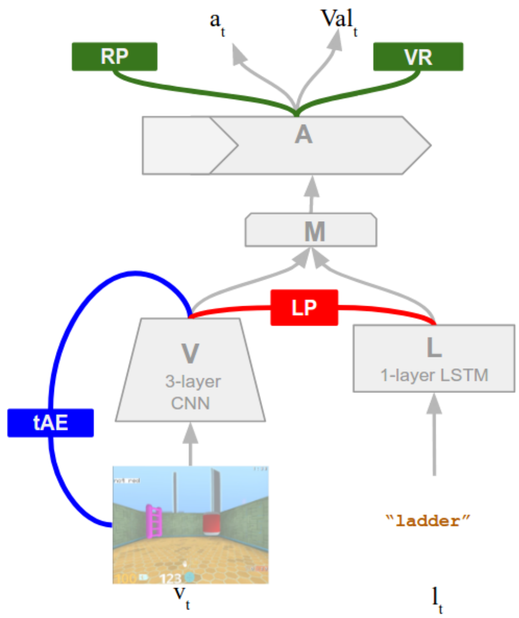

Figure 2: Schematic organisation of the network modules (grey) supplemented with auxiliary learning objectives (coloured components)

other words, it approximates the state-value function $Val_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \lambda^t r_{t+k+1} \mid S_t = s]$ where S_t is the state of the environment at time t when following policy π and r_t is the reward received following the action performed at time t. $\lambda \in [0, 1]$ is a discount parameter.

The agent's primary objective is is to find a policy which maximizes the expected discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \lambda^t r_t]$. We apply the Advantage Actor Critic algorithm (Mnih et al., 2016) to optimize the policy π—a Softmax multinomial distribution parametrized by the agent's network—towards higher discounted returns.

Parameters are updated according to the RMSProp update rule (Tieleman and Hinton, 2012). We share a single parameter vector across 32 asynchronous threads. This configuration offers a suitable trade-off between increased speed and loss of accuracy due to the asynchronous updates (Mnih et al., 2016).

Importantly, early simulation results revealed that this initial design does not learn to solve even comparably simple tasks in our setup. As described thus far, the agent can learn only from comparatively infrequent object selection rewards, without exploiting the stream of potentially useful perceptual feedback available at each time step when exploring the environment. We address this by endowing the agent with ways to learn in an unsupervised manner from its immediate surroundings, by means of auto-regressive objectives that are applied concurrently with the reward-based learning and involve predicting or modelling aspects of the agent's surroundings (Jaderberg et al., 2016).

Temporal autoencoding The temporal autoencoder auxiliary task **tAE** is designed to illicit intuitions in our agent about how the perceptible world will change as a consequence of its actions. The objective is to predict the visual environment v_{t+1} conditioned on the prior visual input v_t and the action a_t (Oh et al., 2015). Our implementation reuses the standard visual module **V** and combines the representation of v_t with an embedded representation of a_t. The combined representation is passed to a deconvolutional network to predict v_{t+1}. As well as providing a means to fine-tune the visual system **V**, the **tAE** auxiliary task results in additional training of the action-policy network, since the action representations can be shared between **tAE** and the policy network π.

Language prediction To strengthen the ability of the agent to reconcile visual and linguistic modalities we design a word prediction objective **LP** that estimates instruction words l_t given the visual observation v_t, using model parameters shared with both **V** and **L**. The **LP** network can also serve to make the behaviour of trained agents more interpretable, as the agent emits words that it considers to best describe what it is currently observing.

The **tAE** and **LP** auxiliary networks were optimised with mini-batch gradient descent based on the mean-squared error and negative-log-likelihood respectively. We also experimented with reward prediction (**RP**) and value replay (**VR**) as additional auxiliary tasks to stabilise reinforcement based training (Jaderberg et al., 2016).

Figure 2 gives a schematic organisation of the agent with all the above auxiliary learning objectives. Precise implementation details of the agent are given in Appendix A.

5. Experiments

In evaluating the agent, we constructed tasks designed to test its capacity to cope with various challenges inherent in language learning and understanding. We first test its ability to efficiently acquire a varied vocabulary of words pertaining to physically observable aspects of the environment. We then examine whether the agent can combine this lexical knowledge to interpret both familiar and unfamiliar word combinations (phrases). This analysis includes phrases whose meaning is dependent of word order, and cases in which the agent must induce and re-use lexical knowledge directly from (potentially ambiguous) phrases. Finally, we test the agent's ability to learn less concrete aspects of language, including instructions referring to relational concepts (Doumas et al., 2008) and phrases referring to actions and behaviours.

5.1 Role of unsupervised learning

Our first experiment explored the effect of the auxiliary objectives on the ability of the agent to acquire a vocabulary of different concrete words (and associated lexical concepts). Training consisted of multiple episodes in a single room containing two objects. For each episode, at time $t = 0$, the agent was spawned in a position equidistant from the two objects, and received a single-word instruction that unambiguously referred to one of the two objects. It received a reward of 1 if it walked over to and selected the correct referent object and -1 if it picked the incorrect object. A new episode began immediately after an object was selected, or if the agent had not selected either object after 300 steps. Objects and instructions were sampled at random from the full set of factors available in the simulation environment.[2] We trained 16 replicas for each agent configuration (Figure 3) with fixed hyperparameters from

2. See Appendix B for a complete list.

the standard settings and random hyperparameters sampled uniformly from the standard ranges.[3]

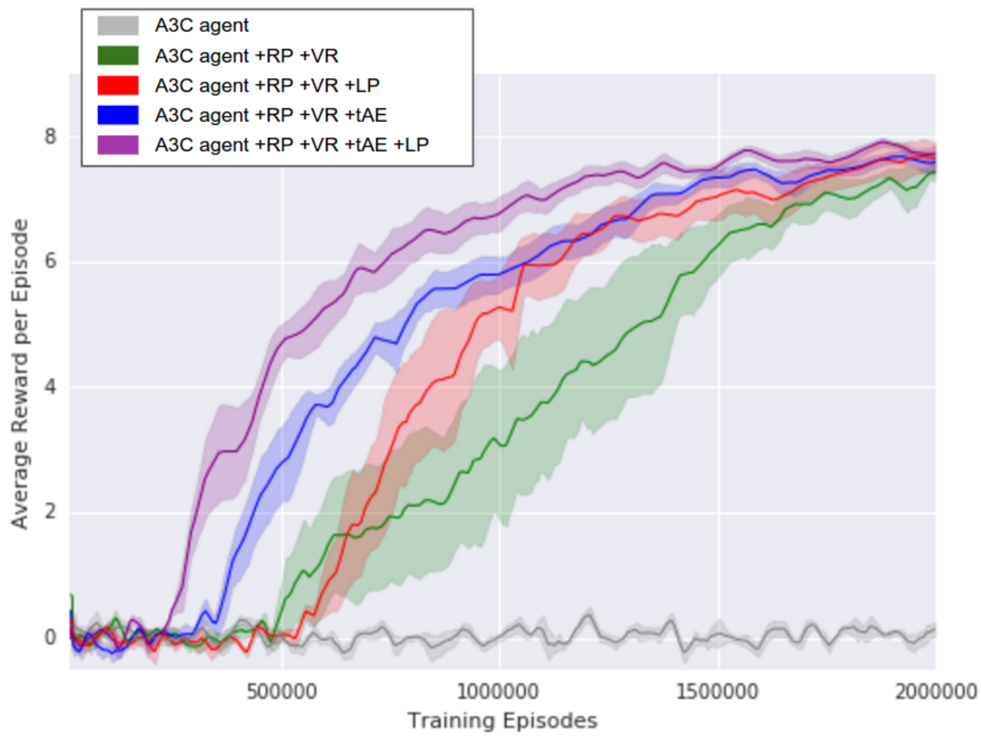

Figure 3: **Unsupervised learning via auxiliary prediction objectives facilitates word learning**. Learning curves for a vocabulary acquisition task. The agent is situated in a single room faced with two objects and must select the object that correctly matches the textual instruction. A total of 59 different words were used as instructions during training, referring to either the shape, colours, relative size (larger, smaller), relative shade (lighter, darker) or surface pattern (striped, spotted, etc.) of the target object. **RP**: reward prediction, **VR**: value replay, **LP**: language prediction, **tAE**: temporal autoencoder. Data show mean and confidence bands (CB) across best five of 16 hyperparameter settings sampled at random from ranges specified in the appendix. Training episodes counts individual levels seen during training.

As shown in Figure 3, when relying on reinforcement learning alone, the agent exhibited no learning even after millions of training episodes. The fastest learning was exhibited by an agent applying both temporal auto-encoding and language prediction in conjunction with value replay and reward prediction. These results demonstrate that auto-regressive objectives can extract information that is critical for language learning from the perceptible environment, even when explicit reinforcement is not available.

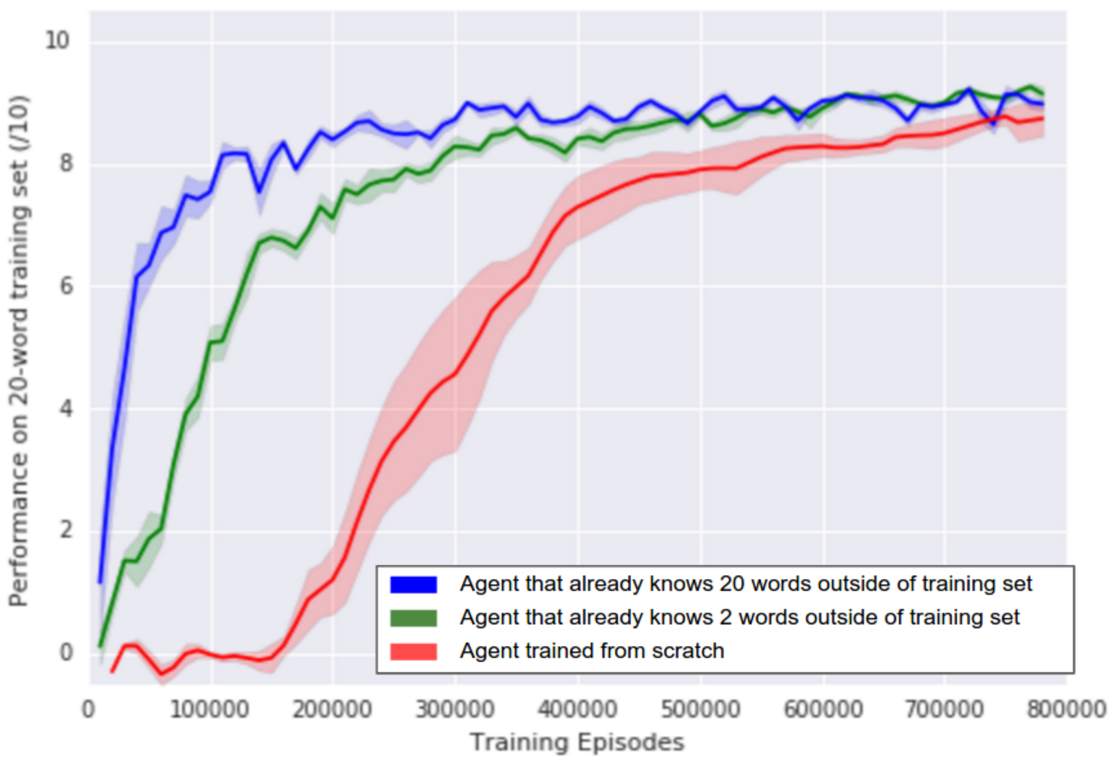

Figure 4: **Word learning is much faster once some words are already known** The rate at which agents learned a vocabulary of 20 shape words was measured in agents in three conditions. In one condition, the agent had prior knowledge of 20 shapes and their names outside of the training data used here. In the second condition, the agent had prior knowledge of two shape words outside of the target vocabulary (same number of pre-training steps). In the third condition, the agent was trained from scratch. All agents used **RP**, **VR**, **LP** and **tAE** auxiliary objectives. Data show mean and confidence bands across best five of 16 hyperparameter settings in each condition, sampled at random from ranges specified in Appendix C.

5.2 Word learning speed experiment

Before it can exhibit any lexical knowledge, the agent must learn various skills and capacities that are independent of the specifics of any particular language instruction. These include an awareness of objects as distinct from floors or walls; some capacity to sense ways in which those objects differ; and the ability to both look and move in the same direction. In addition, the agent must infer that solving the solution to tasks is always contingent on both visual and linguistic input, without any prior programming or explicit teaching of the importance of inter-modal interaction. Given the complexity of this learning challenge, it is perhaps unsurprising that the agent requires thousands of training episodes before evidence of word learning emerges.

To establish the importance of this 'pre-linguistic' learning, we compared the speed of vocabulary acquisition in agents with different degrees of prior knowledge. The training set consisted of instructions (and corresponding environments) from the twenty shape terms *banana, cherries, cow, flower, fork, fridge, hammer, jug, knife, pig, pincer, plant, saxophone, shoe, spoon, tennis-racket, tomato, tree, wine-glass* and *zebra.* The agent with most prior knowledge was trained in advance (in a single room setting with two objects) on the remaining twenty shapes from the full environment. The agent with minimal prior knowledge was trained only on the two terms *ball* and *tv.* Both regimes of advanced training were stopped once the agent reached an average reward of 9.5/10 across 1,000 episodes. The agent with no prior knowledge began learning directly on the training set.

The comparison presented in Figure 4 demonstrates that much of the initial learning in an agent trained from scratch involves acquiring visual and motor, rather than expressly linguistic, capabilities. An agent already knowing two words (and therefore exhibiting rudimentary motor and visual skills) learned new words at a notably faster rate than an agent trained from scratch. Moreover, the speed of word learning appeared to accelerate as more words were learned. This shows that the acquisition of new words is supported not only by general-purpose motor-skills and perception, but also existing lexical or semantic knowledge. In other words, the agent is able to bootstrap its existing semantic knowledge to enable the acquisition of new semantic knowledge.

5.3 One-shot learning experiments

Two important facets of natural language understanding are the ability to compose the meanings of known words to interpret otherwise unfamiliar phrases, and the ability to generalise linguistic knowledge learned in one setting to make sense of new situations. To examine these capacities in our agent, we trained it in settings where its (linguistic or visual) experience was constrained to a training set, and simultaneously as it learned from the training set, tested the performance of the agent on situations outside of this set (Figure 5).

In the **colour-shape composition** experiment, the training instructions were either unigrams or bigrams. Possible unigrams were the 40 shape and the 13 colour terms listed in Appendix B. The possible bigrams were any colour-shape combination except those containing the shapes *ice_lolly, ladder, mug, pencil, suitcase* or the colours *red, magenta, grey, purple* (subsets selected randomly). The test instructions consisted of all possible bigrams excluded from the training set. In each training episode, the target object was

3. See Appendix C for details.

rendered to match the instruction (in colour, shape or both) and the confounding object did not correspond to any of the bigrams in the test set. Similarly, in each test episode, both the target object and the confounding object corresponded to bigrams in the test instructions. These constraints ensured that the agent could not interpret test instructions by excluding other objects or terms that it had seen in the training set.

The **colour-shape decomposition / composition** experiment is similar in design to the colour-shape composition experiment. The test tasks were identical, but the possible training instructions consisted only of the bigram instructions from the colour-shape composition training set. To achieve above chance performance on the test set, the agent must therefore isolate aspects of the world that correspond to each of the constituent words in the bigram instructions (decomposition), and then build an interpretation of novel bigrams using these constituent concepts.

The **relative size** and **relative shade** experiments were designed to test the generality of agents' representation of relational concepts (in this case *larger*, *smaller*, *larger* and *darker*. Training and testing episodes again took place in a single room with two objects. The relative size experiment involved the 16 shapes in our environment whose size could be varied while preserving their shape. The possible instructions in both training and test episodes were simply the unigrams *larger* and *smaller*. The agent was required to choose between two objects of the same shape but different size (and possibly different colour) according to the instruction. All training episodes involved target and confounding objects whose shape was either a *tv, ball, balloon, cake, can, cassette, chair, guitar, hairbrush* or *hat*. All test episodes involved objects whose shape was either an *ice_lolly, ladder, mug, pencil* or *toothbrush*.

The relative shade experiment followed the same design, but the agent was presented with two objects of possibly differing shape that differed only in the shade of their colouring (e.g. one light blue and one dark blue). The training colours were *green, blue, cyan, yellow, pink, brown* and *orange*. The test colours were *red, magenta, grey* and *purple*.

When trained on colour and shape unigrams together with a limited number of colour-shape bigrams, the agent naturally understood additional colour-shape bigrams if it is familiar with both constituent words. Moreover, this ability to productively compose known words to interpret novel phrases was not contingent on explicit training of those words in isolation. When exposed only to bigram phrases during training, the agent inferred the constituent lexical concepts and reapplied these concepts to novel combinations at test time. Indeed, in this condition (the decomposition/composition case), the agent learned to generalise after fewer training instances than in the apparently simpler composition case. This can be explained by by the fact that episodes involving bigram instructions convey greater information content, such that the latter condition avails the agent of more information per training episode. Critically, the agent's ability to decompose phrases into constituent (emergent) lexical concepts reflects an ability that may be essential for human-like language learning in naturalistic environments, since linguistic stimuli rarely contain words in isolation.

Another key requirement for linguistic generalisation is the ability to extend category terms beyond the specific exemplars from which those concepts were learned (Quinn et al., 1993; Rogers and McClelland, 2004). This capacity was also observed in our agent; when trained on the relational concepts *larger* and *smaller* in the context of particular shapes

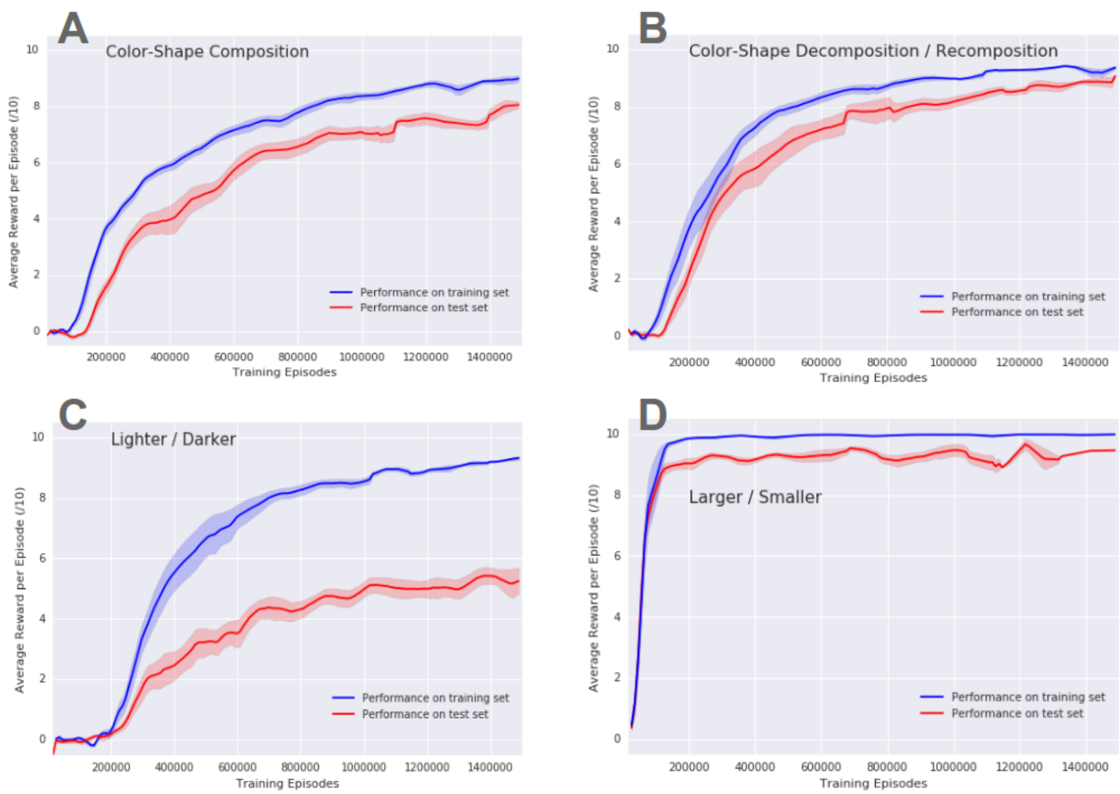

Figure 5: **Semantic knowledge generalises to unfamiliar language and objects. Composition (A)**: training covered all shape and colour unigrams and $\sim 90\%$ of possible colour-shape bigrams, such as blue ladder. Agents were periodically tested on the remaining 10% of bigrams without updating parameters. **Decomposition-composition (B)**: the same regime as in A, but without any training on unigram descriptors. **Lighter / darker (C)**: agents were trained to interpret the terms lighter and darker applied to a set of colours, and tested on the terms in the context of a set of different colours. **Relative size (D)**: agents were trained to interpret the terms larger and smaller applied to a set of shapes, and tested on the terms in the context of a set of different shapes. Data show mean and CB across best five of 16 randomly sampled hyperparameter settings in each condition. See Appendix B for hyperparameter ranges and exact train/test stimuli.

it naturally applied them to novel shapes with almost perfect accuracy. In contrast, the ability to generalise *lighter* and *darker* to unfamiliar colours was significantly above chance but less than perfect. This may be because it is particularly difficult to infer the mapping corresponding to lighter and darker (as understood by humans) in an RGB colour space from the small number of examples observed during training.

Taken together, these instances of generalisation demonstrate that our agent does not simply ground language in hard coded features of the environment such as pixel activations or specific action sequences, but rather learns to ground meaning in more abstract semantic representations. More practically, these results also suggest how artificial agents that are necessarily exposed to finite training regimes may ultimately come to exhibit the productivity characteristic of human language understanding.

5.4 Extending learning via a curriculum

A consequence of the agent's facility for re-using its acquired knowledge for further learning is the potential to train the agent on more complex language and tasks via exposure to a curriculum of levels. Figure 6 shows an example for the successful application of such a curriculum, here applied to the task of selecting an object based on the floor colour of the room it is located in.

We also applied a curriculum to train an agent on a range of multi-word referring instructions of the form pick the X, where X represents a string consisting of either a single noun (shape term, such as *chair*) an adjective and a noun (a colour term, pattern term or shade term, followed by a shape term, such as *striped ladder*) or two adjectives and a noun (a shade term or a pattern term, followed by a colour term, followed by a shape term, such as *dark purple toothbrush*). The latter two cases were also possible with the generic term 'object' in place of a shape term. In each case, the training episode involved one object that coincided with the instruction and some number of distractors that did not. Learning curves for this 'referring expression agent' are illustrated in Figure 7.

5.5 Multi-task learning

Language is typically used to refer to actions and behaviours as much as to objects and entities. To test the ability of our agents to ground such words in corresponding procedures, we trained a single agent to follow instructions pertaining to three dissociable tasks. We constructed these tasks using a two-room world with both floor colourings and object properties sampled at random.

In this environment, the **Selection** task involved instructions of the form pick the X object or pick all X, where X denotes a colour term. The **Next to** task involved instructions of the form pick the X object next to the Y object, where X and Y refer to objects. Finally, the **In room** task involved instructions of the form pick the X in the Y room, where Y referred to the colour of the floor in the target room. Both the **Next to** and the **In room** task employed large degrees of ambiguity, i.e. a given **Next to** level may contain several objects X and Y, but in a constellation that only one X would be located next to a Y.

The agent was exposed to instances of each task with equal probability during training. The possible values for variables X and Y in these instructions were *red, blue, green, yellow,*

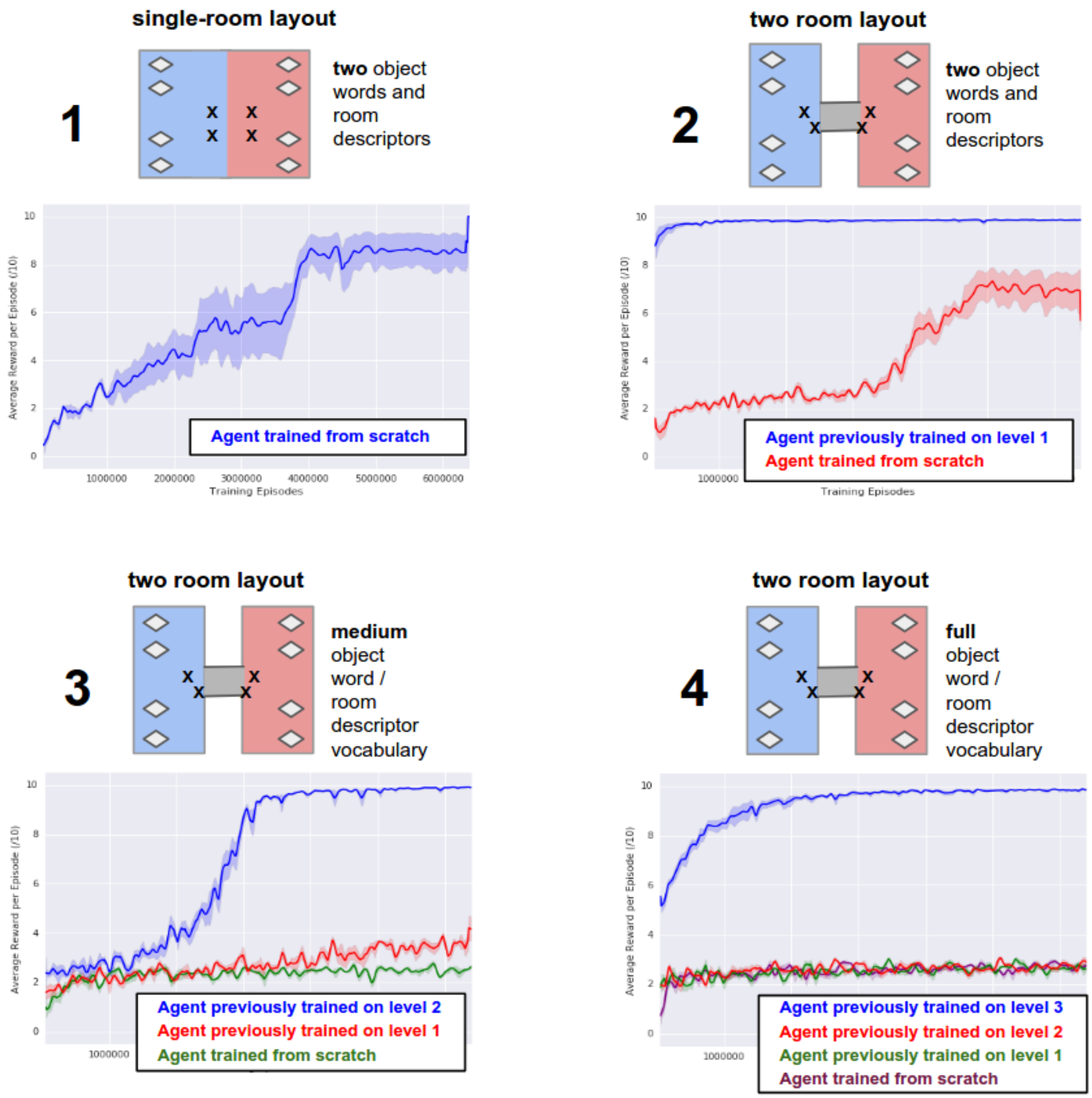

Figure 6: **Curriculum learning is necessary for solving more complex tasks.** For the agent to learn to retrieve an object in a particular room as instructed, a four-lesson training curriculum was required. Each lesson involved a more complex layout or a wider selection of objects and words, and was only solved by an agent that had successfully solved the previous lesson. The schematic layout and vocabulary scope for each lesson is shown above the training curves for that lesson. The initial (spawn) position of this agent varies randomly during training among the locations marked **x**, as do the position of the four possible objects among the positions marked with a white diamond. Data show mean and CB across best five of 16 randomly sampled hyperparameter settings in each condition.

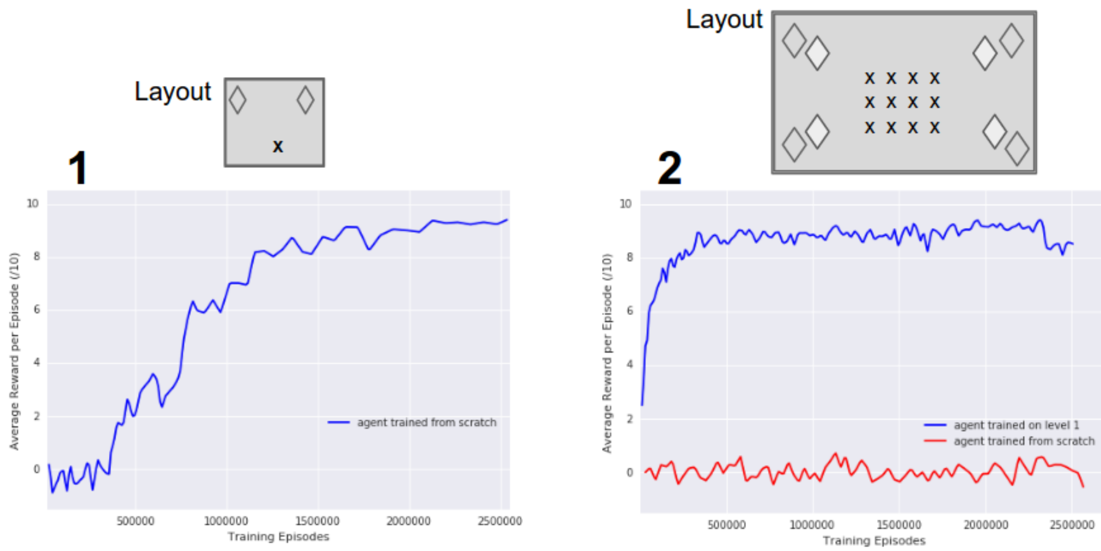

Figure 7: **Learning curve for the referring expression agent.** The trained agent is able to select the correct object in a two-object setup when described using a compositional expression. This ability transfers to more complex environments with a larger number of confounding objects.

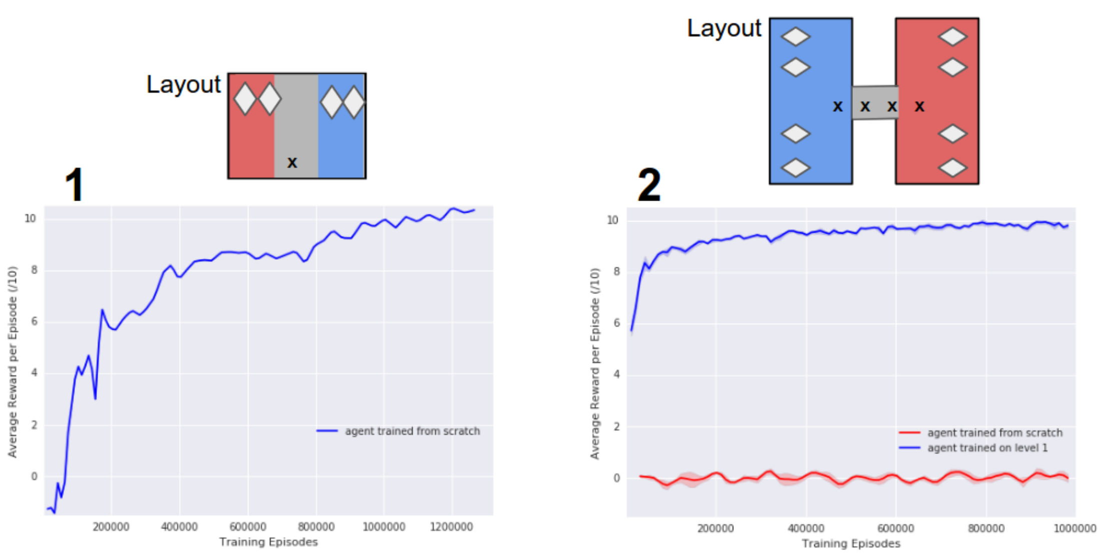

Figure 8: **Multi-task learning via an efficient curriculum of two steps.** A single agent can learn to solve a number of different tasks following a two-lesson training curriculum. The different tasks cannot be distinguished based on visual information alone, but require the agent to use the language input to identify the task in question.

cyan and *magenta*. The shape of all objects in the environment was selected randomly from 40 possibilities.

As previously, a curriculum was required to achieve the best possible agent performance on these tasks (see Figure 8). When trained from scratch, the agent learned to solve all three types of task in a single room where the colour of the floor was used as a proxy for a different room. However, it was unable to achieve the same learning in a larger layout with two distinct rooms separated by a corridor. When the agent trained in a single room was transferred to the larger environment, it continued learning and eventually was able to solve the more difficult task.[4]

By learning these tasks, this agent demonstrates an ability to ground language referring not only to single (concrete) objects, but also to (more abstract) sequences of actions, plans and inter-entity relationships. Moreover, in mastering the **Next to** and **In room** tasks, the agent exhibits sensitivity to a critical facet of many natural languages, namely the dependence of utterance meaning on word order. The ability to solve more complex tasks by curriculum training emphasises the generality of the emergent semantic representations acquired by the agent, allowing it to transfer learning from one scenario to a related but more complex environment.

6. Conclusion

An artificial agent capable of relating natural languages to the physical world would transform everyday interactions between humans and technology. We have taken an important step towards this goal by describing an agent that learns to execute a large number of multi-word instructions in a simulated three-dimensional world, with no pre-programming or hard-coded knowledge. The agent learns simple language by making predictions about the world in which that language occurs, and by discovering which combinations of words, perceptual cues and action decisions result in positive outcomes. Its knowledge is distributed across language, vision and policy networks, and pertains to modifiers, relational concepts and actions, as well as concrete objects. Its semantic representations enable the agent to productively interpret novel word combinations, to apply known relations and modifiers to unfamiliar objects and to re-use knowledge pertinent to the concepts it already has in the process of acquiring new concepts.

While our simulations focus on language, the outcomes are relevant to machine learning in a more general sense. In particular, the agent exhibits active, multi-modal concept induction, the ability to transfer its learning and apply its knowledge representations in unfamiliar settings, a facility for learning multiple, distinct tasks, and the effective synthesis of unsupervised and reinforcement learning. At the same time, learning in the agent reflects various effects that are characteristic of human development, such as rapidly accelerating rates of vocabulary growth, the ability to learn from both rewarded interactions and predictions about the world, a natural tendency to generalise and re-use semantic knowledge, and improved outcomes when learning is moderated by curricula (Vosniadou and Brewer, 1992; Smith et al., 1996; Pinker, 1987, 2009). Taken together, these contributions open many avenues for future investigations of language learning, and learning more generally, in both humans and artificial agents.

4. See `https://youtu.be/wJjdu1bPJ04` for a video of the final trained agent.

References

Dilip Arumugam, Siddharth Karamcheti, Nakul Gopalan, Lawson L. S. Wong, and Stefanie Tellex. Accurately and efficiently interpreting human-robot instructions of varying granularities. CoRR, abs/1704.06616, 2017. URL http://arxiv.org/abs/1704.06616.

Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. CoRR, abs/1612.03801, 2016. URL http://arxiv.org/abs/1612.03801.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In EMNLP, pages 1533–1544. ACL, 2013. ISBN 978-1-937284-97-8. URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#BerantCFL13.

David L. Chen and Raymond J. Mooney. Learning to sportscast: A test of grounded language acquisition. In Proceedings of the 25th International Conference on Machine Learning (ICML), Helsinki, Finland, July 2008. URL http://www.cs.utexas.edu/users/ai-lab/?chen:icml08.

Noam Chomsky. A review of BF Skinner's Verbal Behavior. Language, 35(1):26–58, 1959.

Leonidas AA Doumas, John E Hummel, and Catherine M Sandhofer. A theory of the discovery and predication of relational concepts. Psychological review, 115(1):1, 2008.

Sachithra Hemachandra, Matthew R. Walter, Stefanie Tellex, and Seth Teller. Learning spatially-semantic representations from natural language descriptions and scene classifications. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, May 2014.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In International Conference on Learning Representations, 2016.

Samantha Krening, Brent Harrison, Karen M Feigh, Charles Isbell, Mark Riedl, and Andrea Thomaz. Learning from explanations using sentiment and advice in RL. In 2016 IEEE Transactions on Cognitive and Developmental Systems. IEEE, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

Tomas Mikolov, Armand Joulin, and Marco Baroni. A roadmap towards machine intelligence. arXiv preprint arXiv:1511.08130, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning, 2016.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. In Advances in Neural Information Processing Systems 28, 2015.

Steven Pinker. The bootstrapping problem in language acquisition. Mechanisms of language acquisition, pages 399–441, 1987.

Steven Pinker. Language learnability and language development, volume 7. Harvard University Press, 2009.

W. V. O. Quine. Word & Object. MIT Press, 1960.

Paul C Quinn, Peter D Eimas, and Stacey L Rosenkrantz. Evidence for representations of perceptually similar natural categories by 3-month-old and 4-month-old infants. Perception, 22(4):463–475, 1993.

Timothy T Rogers and James L McClelland. Semantic cognition: A parallel distributed processing approach. MIT press, 2004.

Deb K Roy and Alex P Pentland. Learning words from sights and sounds: A computational model. Cognitive science, 26(1):113–146, 2002.

Carina Silberer and Mirella Lapata. Grounded models of semantic representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2012.

Jeffrey Mark Siskind. Grounding Language in Perception, pages 207–227. Springer Netherlands, Dordrecht, 1995. ISBN 978-94-011-0273-5. doi: 10.1007/978-94-011-0273-5_12. URL `http://dx.doi.org/10.1007/978-94-011-0273-5_12`.

Jeffrey Mark Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. J. Artif. Intell. Res. (JAIR), 15:31–90, 2001. doi: 10.1613/jair.790. URL `https://doi.org/10.1613/jair.790`.

Linda B Smith, Susan S Jones, and Barbara Landau. Naming in young children: A dumb attentional mechanism? Cognition, 60(2):143–171, 1996.

Luc Steels. The symbol grounding problem has been solved. so what's next. Symbols and embodiment: Debates on meaning and cognition, pages 223–244, 2008.

Jesse Thomason, Shiqi Zhang, Raymond Mooney, and Peter Stone. Learning to interpret natural language commands through human-robot dialog. In Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pages 1923–1929. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832415.2832516.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2), 2012.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. CoRR, abs/1511.06361, 2015. URL http://arxiv.org/abs/1511.06361.

Stella Vosniadou and William F Brewer. Mental models of the earth: A study of conceptual change in childhood. Cognitive psychology, 24(4):535–585, 1992.

Matthew R. Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth Teller. A framework for learning semantic maps from grounded natural language descriptions. The International Journal of Robotics Research, 33(9):1167–1190, 2014. doi: 10.1177/0278364914537359. URL http://dx.doi.org/10.1177/0278364914537359.

S. I. Wang, P. Liang, and C. Manning. Learning language games through interaction. In Association for Computational Linguistics (ACL), 2016.

Terry Winograd. Understanding natural language. Cognitive psychology, 3(1):1–191, 1972.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. International Conference of Machine Learning, 2(3):5, 2015.

Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In ACL, pages 53–63. The Association for Computer Linguistics, 2013. ISBN 978-1-937284-50-3.

Haonan Yu, Haichao Zhang, and Wei Xu. A deep compositional framework for human-like language acquisition in virtual environment. CoRR, abs/1703.09831, 2017. URL https://arxiv.org/abs/1703.09831.

Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI'05, pages 658–666, Arlington, Virginia, United States, 2005. AUAI Press. ISBN 0-9749039-1-4. URL http://dl.acm.org/citation.cfm?id=3020336.3020416.

Appendix A. Agent details

A.1 Agent core

At every time-step t the vision module $\mathbf{V}$ receives an 84×84 pixel RGB representation of the agent's (first person) view of the environment ($x_t^v \in \mathbb{R}^{3\times 84\times 84}$), which is then processed with a three-layer convolutional neural network (LeCun et al., 1989) to emit an output representation $v_t \in \mathbb{R}^{64\times 7\times 7}$. The first layer of the convolutional network contains 8 kernels applied at stride width 4, resulting in 32 (20×20) output channels. The second layer applies 4 kernels at stride with 2 yielding 64 (9×9) output channels. The third layer applies 3 kernels at stride width 1 resulting again in 64 (7×7) output channels.

The language module receives an input $x_t^l \in \mathbb{N}^s$, where s is the maximum instruction length with words represented as indices in a dictionary. For tasks that require sensitivity to the order of words in the language instruction, the language module $\mathbf{L}$ encodes x_t^l with a recurrent (LSTM) architecture (Hochreiter and Schmidhuber, 1997). For other tasks, we applied a simpler bag-of-words (BOW) encoder, in which an instruction is represented as the sum of the embeddings of its constituent words, as this resulted in faster training. Both the LSTM and BOW encoders use word embeddings of dimension 128, and the hidden layer of the LSTM is also of dimension 128, resulting in both cases in an output representation $l_t \in \mathbb{R}^{128}$.

In the mixing module $\mathbf{M}$, outputs v_t and l_t are combined by flattening v_t into a single vector and concatenating the two resultant vectors into a shared representation m_t. The output from $\mathbf{M}$ at each time-step is fed to the action module $\mathbf{A}$ which maintains the agent state $h_t \in \mathbb{R}^d$. h_t is updated using an LSTM network combining output m_t from $\mathbf{M}$ and h_{t-1} from the previous time-step. By default we set $d = 256$ in all our experiments.

A.2 Auxiliary networks

Temporal Autoencoder The temporal autoencoder auxiliary network **tAE** samples sequences containing two data points x_i, x_{i+1} as well as one-shot action representation $a_i \in \mathbb{N}^a$. It encodes x_i^v using the convolutional network defined by $\mathbf{V}$ into $y \in \mathbb{R}^{64\times 7\times 7}$. The feature representation is then transformed using the action a_i,

$$\hat{y} = W_{\hat{y}} \left(W_b a_i \odot W_v y \right),$$

with $\hat{y} \in \mathbb{R}^{64\times 7\times 7}$. The weight matrix W_b shares its weights with the final layer of the perceptron computing π in the core policy head. The transformed visual encoding $\hat{y}$ is passed into a deconvolutional network (mirroring the configuration of the convolutional encoder) to emit a predicted input $w \in \mathbb{R}^{3\times 84\times 84}$. The **tAE** module is optimised on the mean-squared loss between w and x_{i+1}^v.

Language Prediction At each time-step t, the language prediction auxiliary network **LP** applies a replica of $\mathbf{V}$ (with shared weights) to encode v_t. A linear layer followed by a rectified linear activation function is applied to transform this representation from size $64 \times 7 \times 7$ to a flat vector of dimension 128 (the same size as the word embedding dimension in $\mathbf{L}$). This representation is then transformed to an output layer with the same number of units as the agent's vocabulary. The weights in this final layer are shared with the initial layer (word embedding) weights from $\mathbf{L}$. The output activations are fed through a Softmax activation function to yield a probability distribution over words in the vocabulary, and the negative log likelihood of the instruction word l_t is computed as the loss. Note that this objective requires a single meaningful word to be extracted from the instruction as the target.

Appendix B. Environment details

The environment can contain any number of rooms connected through corridors. A level in the simulated 3D world is described by a map (a combination of rooms), object specifiers, language and a reward function. Objects in the world are drawn from a fixed inventory and can be described using a combination of five factors.

Shapes (40) *tv, ball, balloon, cake, can, cassette, chair, guitar, hairbrush, hat, ice_lolly, ladder, mug, pencil, suitcase, toothbrush, key, bottle, car, cherries, fork, fridge, hammer, knife, spoon, apple, banana, cow, flower, jug, pig, pincer, plant, saxophone, shoe, tennis racket, tomato, tree, wine glass, zebra.*

Colours (13) *red , blue , white , grey , cyan , pink , orange , black , green , magenta , brown , purple , yellow.*

Patterns (9) *plain, chequered, crosses, stripes, discs, hex, pinstripe, spots, swirls.*

Shades (3) *light, dark, neutral.*

Sizes (3) *small, large, medium.*

Within an environment, agent spawn points and object locations can be specified or randomly sampled. The environment itself is subdivided into multiple rooms which can be distinguished through randomly sampled (unique) floor colours. We use up to seven factors to describe a particular object: the five object-internal factors, the room it is placed in and its proximity to another object, which can itself be described by its five internal factors.

In all simulations presented here, reward is attached to picking up a particular object. Reward is scaled to be in $[-10; 10]$ and, where possible, balanced so that a random agent would have an expected reward of 0. This prevents agents from learning degenerate strategies that could otherwise allow them to perform well in a given task without needing to learn to ground the textual instructions.

Appendix C. Hyperparameters

Tables 1 and 2 show parameter setting used throughout the experiments presented in this paper. We report results with confidence bands (CB) equivalent to $\pm$ one standard deviation on the mean, assuming normal distribution.

Hyperparameter	Value	Description
train_steps	640m	Theoretical maximum number of time steps (across all episodes) for which the agent will be trained.
env_steps_per_core_step	4	Number of time steps between each action decision (action smoothing)
num_workers	32	Number of independent workers running replicas of the environment with asynchronous updating.
unroll_length	50	Number of time steps through which error is backpropagated in the core LSTM action module
auxiliary networks		
vr_batch_size	1	Aggregated time steps processed by value replay auxiliary for each weight update.
rp_batch_size	10	Aggregated time steps processed by reward prediction auxiliary for each weight update.
lp_batch_size	10	Aggregated time steps processed by language prediction auxiliary for each weight update.
tae_batch_size	10	Aggregated time steps processed by temporal AE auxiliary for each weight update.
language encoder		
encoder_type	BOW	Whether the language encoder uses an additive bag-of-words (BOW) or an LSTM architecture.
cost calculation		
additional_discounting	0.99	Discount used to compute the long-term return R_t in the A3C objective
cost_base	0.5	Multiplicative scaling of all computed gradients on the backward pass in the network
optimisation		
clip_grad_norm	100	Limit on the norm of the gradient across all agent network parameters (if above, scale down)
decay	0.99	Decay term in RMSprop gradient averaging function
epsilon	0.1	Epsilon term in RMSprop gradient averaging function
learning_rate_finish	0	Learning rate at the end of training, based on which linear annealing of is applied.
momentum	0	Momentum parameter in RMSprop gradient averaging function

Table 1: Agent hyperparameters that are fixed throughout our experimentation but otherwise not specified in the text.

Hyperparameter	Value	Description
auxiliary networks		
vr_weight	*uniform(0.1, 1)*	Scalar weighting of value replay auxiliary loss relative to the core (A3C) objective.
rp_weight	*uniform(0.1, 1)*	Scalar weighting of reward prediction auxiliary loss.
lp_weight	*uniform(0.1, 1)*	Scalar weighting of language prediction auxiliary loss.
tae_weight	*uniform(0.1, 1)*	Scalar weighting of temporal autoencoder prediction auxiliary.
language encoder		
embed_init	*uniform(0.5, 1)*	Standard deviation of normal distribution (mean = 0) for sampling initial values of word-embedding weights in **L**.
optimisation		
entropy_cost	*uniform(0.0005, 0.005)*	Strength of the (additive) entropy regularisation term in the A3C cost function.
learning_rate_start	*loguniform(0.0001, 0.002)*	Learning rate at the beginning of training annealed linearly to reach learning_rate_finish at the end of train_steps.

Table 2: Agent hyperparameters that randomly sampled in order to yield different replicas of our agents for training. *uniform*(x, y) indicates that values are sampled uniformly from the range $[x, y]$. *loguniform*(x, y) indicates that values are sampled from a uniform distribution in log-space (favouring lower values) on the range $[x, y]$.