Analogs of Linguistic Structure in Deep Representations

Jacob Andreas and **Dan Klein**
Computer Science Division
University of California, Berkeley
jda,klein@cs.berkeley.edu

Abstract

We investigate the compositional structure of message vectors computed by a deep network trained on a communication game. By comparing truth-conditional representations of encoder-produced message vectors to human-produced referring expressions, we are able to identify aligned (vector, utterance) pairs with the same meaning. We then search for structured relationships among these aligned pairs to discover simple vector space transformations corresponding to negation, conjunction, and disjunction. Our results suggest that neural representations are capable of spontaneously developing a "syntax" with functional analogues to qualitative properties of natural language.[1]

1 Introduction

The past year has seen a renewal of interest in end-to-end learning of communication strategies between pairs of agents represented with deep networks (Wagner et al., 2003). Approaches of this kind make it possible to learn decentralized policies from scratch (Foerster et al., 2016; Sukhbaatar et al., 2016), with multiple agents coordinating via learned communication protocol. More generally, any encoder–decoder model (Sutskever et al., 2014) can be viewed as implementing an analogous communication protocol, with the input encoding playing the role of a message in an artificial "language" shared by the encoder and decoder (Yu et al., 2016). Earlier work has found that under suitable conditions, these protocols acquire simple interpretable lexical (Dircks and Stoness, 1999; Lazaridou et al., 2016) and sequential structure (Mordatch and Abbeel, 2017), even without natural language training data.

[1] Code and data are available at http://github.com/jacobandreas/rnn-syn.

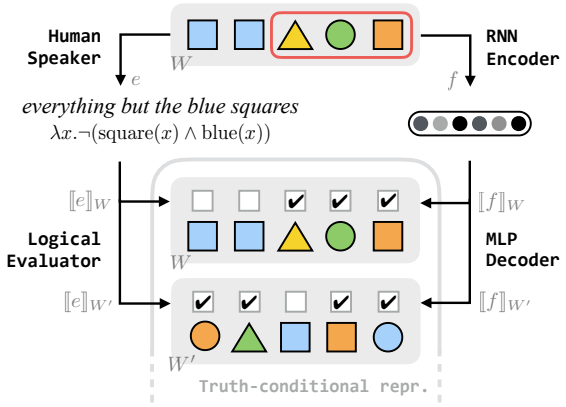

Figure 1: Overview of our task. Given a dataset of referring expression games, example human expressions, and their associated logical forms, we compute explicit denotations both for the original task and in other possible tasks—giving rise to a truth-conditional representation of the natural language. We train a recurrent encoder–decoder model to solve the same tasks directly, and use the decoder to generate comparable truth-conditional representations of neural encodings.

One of the distinguishing features of natural language is compositionality: the existence of operations like negation and coordination that can be applied to utterances with predictable effects on meaning. RNN models trained for natural language processing tasks have been found to learn representations that encode some of this compositional structure—for example, sentence representations for machine translation encode explicit features for certain syntactic phenomena (Shi et al., 2016) and represent some semantic relationships translationally (Levy et al., 2014). It is thus natural to ask whether these "language-like" structures also arise spontaneously in models trained directly from an environment signal. Rather than using language as a form of supervision, we propose to use it as a *probe*—exploiting post-hoc statistical correspondences between natural language descriptions and neural encodings to discover regular structure in representation space.

To do this, we need to find (vector, string) pairs with matching semantics, which requires first aligning unpaired examples of human–human

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2893–2897
Copenhagen, Denmark, September 7–11, 2017.

communication with network hidden states. This is similar to the problem of "translating" RNN representations recently investigated in Andreas et al. (2017). Here we build on that approach in order to perform a detailed analysis of *compositional* structure in learned "languages". We investigate a communication game previously studied by FitzGerald et al. (2013), and make two discoveries: in a model trained without any access to language data,

1. The strategies employed by human speakers in a given communicative context are surprisingly good predictors of RNN behavior in the same context: humans and RNNs send messages whose interpretations agree on nearly 90% of object-level decisions, even outside the contexts in which they were produced.

2. Interpretable language-like structure naturally arises in the space of representations. We identify geometric regularities corresponding to negation, conjunction, and disjunction, and show that it is possible to linearly transform representations in ways that approximately correspond to these logical operations.

2 Task

We focus our evaluation on a communication game due to FitzGerald et al. (2013) (Figure 1, top). In this game, the *speaker* observes (1) a *world* W of 1–20 objects labeled with with attributes and (2) a designated *target* subset X of objects in the world. The *listener* observes only W, and the speaker's goal is to communicate a representation of X that enables the listener to accurately reconstruct it. The GENX dataset collected for this purpose contains 4170 human-generated natural-language referring expressions and corresponding logical forms for 273 instances of this game. Because these human-generated expressions have all been pre-annotated, we treat language and logic interchangeably and refer to both with the symbol e. We write $e(W)$ for the expression generated by a human for a particular world W, and $[\![e]\!]_W$ for the result of evaluating the logical form e against W.

We are interested in using language data of this kind to analyze the behavior of a deep model trained to play the same game. We focus our analysis on a standard RNN encoder–decoder, with the encoder playing the role of the speaker and the decoder playing the role of the listener. The encoder is a single-layer RNN with GRU cells (Cho et al., 2014) that consumes both the input world and target labeling and outputs a 64-dimensional hidden representation. We write $f(W)$ for the output of this encoder model on a world W. To make predictions, this representation is passed to a decoder implemented as a multilayer perceptron. The decoder makes an independent labeling decision about every object in W (taking as input both f and a feature representation of a particular object W_i). We write $[\![f]\!]_W$ for the full vector of decoder outputs on W. We train the model maximize classification accuracy on randomly-generated scenes and target sets of the same form as in the GENX dataset.

3 Approach

We are not concerned with the RNN model's raw performance on this task (it achieves nearly perfect accuracy). Instead, our goal is to explore what kinds of messages the model computes in order to achieve this accuracy—and specifically whether these messages contain high-level semantics and low-level structure similar to the referring expressions produced by humans. But how do we judge semantic equivalence between natural language and vector representations? Here, as in Andreas et al. (2017), we adopt an approach inspired by formal semantics, and represent the meaning of messages via their *truth conditions* (Figure 1).

For every problem instance W in the dataset, we have access to one or more human messages $e(W)$ as well as the RNN encoding $f(W)$. The truth-conditional account of meaning suggests that we should judge e and f to be equivalent if they designate the same set of of objects in the world (Davidson, 1967). But it is not enough to compare their predictions solely in the context where they were generated—testing if $[\![e]\!]_W = [\![f]\!]_W$—because any pair of models that achieve perfect accuracy on the referring expression task will make the same predictions in this initial context, regardless of the meaning conveyed.

Instead, we sample a collection of alternative worlds $\{W_i\}$ observed elsewhere in the dataset, and compute a tabular meaning representation $\textit{rep}(e) = \{[\![e]\!]_{W_i}\}$ by evaluating e in each world W_i. We similarly compute $\textit{rep}(f) = \{[\![f]\!]_{W_i}\}$, allowing the learned decoder model to play the role of logical evaluation for message vectors. For

	Theory	Objects	Worlds	Tables
All	Random	0.50	0.00	0.00
	Literal	0.74	0.27	0.05
	Human	0.92	0.63	0.35

Table 1: Agreement with predicted model behavior for the high-level semantic correspondence task, computed for objects (single entries in tabular representation), worlds (rows), and full tables. Referring expressions e generated by humans in a single communicative context are highly predictive of how learned representations f will be interpreted by the decoder across multiple contexts.

logically equivalent messages, these tabular representations are guaranteed to be identical, so the sampling procedure can be viewed as an approximate test of equivalence. It additionally allows us to compute softer notions of equivalence by measuring agreement on individual worlds or objects.

4 Interpreting the meaning of messages

We begin with the simplest question we can answer with this tool: how often do the messages generated by the encoder model have the same meaning as messages generated by humans for the same context? Again, our goal is not to evaluate the performance of the RNN model, but instead our ability to understand its behavior. Does it send messages with human-like semantics? Is it more explicit? Or does it behave in a way indistinguishable from a random classifier?

For each scene in the GENX test set, we compute the model-generated message f and its tabular representation $rep(f)$, and measure the extent to which this agrees with representations produced by three "theories" of model behavior (Figure 2): (1) a **random** theory that accepts or rejects objects with uniform probability, (2) a **literal** theory that predicts membership only for objects that exactly match some object in the original target set, and (3) a **human** theory that predicts according to the most frequent logical form associated with natural language descriptions of the target set (as described in the preceding section). We evaluate agreement at the level of individual objects, worlds, and full tabular meaning representations.

Results are shown in Table 1. Model behavior is well explained by human decisions in the same context: object-level decisions can be predicted with close to 90% accuracy based on human judgments alone, and a third of message pairs agree exactly in every sampled scene, providing strong evidence that they carry the same semantics.

These results suggest that the model has learned a communication strategy that is at least superficially language-like: it admits representations of the same kinds of communicative abstractions that humans use, and makes use of these abstractions with some frequency. But this is purely a statement about the high-level behavior of the model, and not about the structure of the space of representations. Our primary goal is to determine whether this behavior is achieved using low-level structural regularities in vector space that can themselves be associated with aspects of natural language communication.

5 Interpreting the structure of messages

For this we turn to a focused investigation of three specific logical constructions used in natural language: a unary operation (negation) and two binary operations (conjunction and disjunction). All are used in the training data, with a variety of scopes (e.g. *all green objects that are not a triangle*, *all the pieces that are not tan arches*).

Because humans often find it useful to specify the target set by exclusion rather than inclusion, we first hypothesize that the RNN language might find it useful to incorporate some mechanism cor-

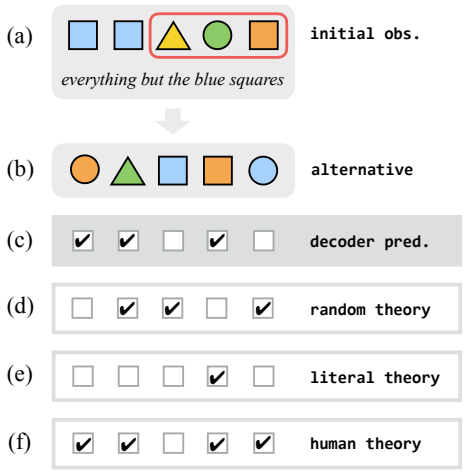

Figure 2: Evaluating theories of model behavior. First, the encoder is run on an initial world (a), producing a representation whose meaning we would like to understand (see Figure 1). We then observe the behavior of the decoder holding this representation fixed but replacing the underlying world representation with alternatives like (b). We compare the true decoder output to a number of *theories* of its behavior. The random theory (d) outputs a random decision for every object. The literal theory (e) predicts that the decoder will output a positive label only on those objects that exactly match some object in the initial observation. The human theory (f) assigns labels according to the logical semantics of the utterance produced by a human presented with the initial observation.

	Theory	Objects	Worlds	Tables
Neg.	Random	0.50	0.00	0.00
	Literal	0.50	0.12	0.03
	Negation	0.97	0.81	0.45
Disj.	Random	0.50	0.00	0.00
	Literal	0.58	0.09	0.01
	Disjunction	0.92	0.54	0.19
Conj.	Random	0.50	0.00	0.00
	Literal	0.81	0.19	0.01
	Conjunction	0.90	0.56	0.37

Table 2: Agreement with predicted model behavior for negation, conjunction, and disjunction tasks (top to bottom). Evaluation is performed on transformed message vectors as described in Section 5. We discover a robust linear transformation of message vectors corresponding to negation, as well as evidence of structured representations of binary operations.

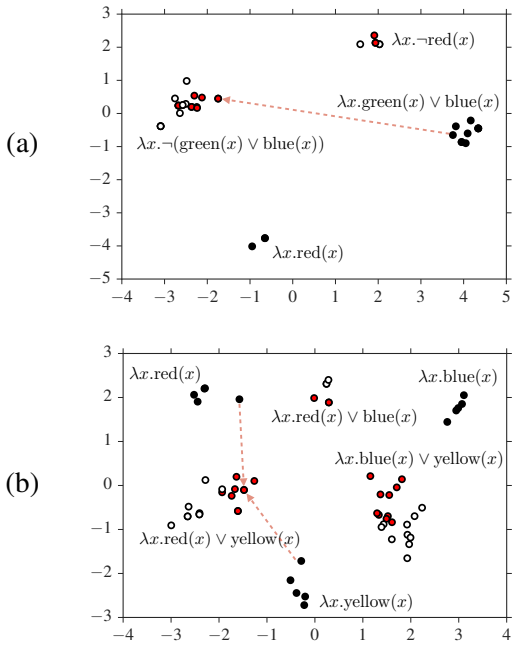

Figure 3: Principal components of structured message transformations discovered by our experiments. (a) Negation: black and white dots show raw message vectors denotationally equivalent to the provided logical cluster label (Section 3). Red dots show the result of transforming black dots with the estimated negation operation N. (b) The corresponding experiment for disjunction using the transformation M.

responding to negation, and that messages can be predictably "negated" in vector space. To test this hypothesis, we first collect examples of the form (e, f, e', f'), where $e' = \neg e$, $\textit{rep}(e) = \textit{rep}(f)$, and $\textit{rep}(e') = \textit{rep}(f')$. In other words, we find pairs of pairs of RNN representations f and f' for which the natural language messages (e, e') serve as a denotational *certificate* that f' behaves as a negation of f. If the learned model does not have any kind of primitive notion of negation, we expect that it will not be possible to find any kind of predictable relationship between pairs (f, f'). (As an extreme example, we could imagine every possible prediction rule being associated with a different point in the representation space, with the correspondence between position and behavior essentially random.) Conversely, if there is a first-class notion of negation, we should be able to select an arbitrary representation vector f with an associated referring expression e, apply some transformation N to f, and be able to predict *a priori* how the decoder model will interpret the representation Nf—i.e. in correspondence with $\neg e$.

Here we make the strong assumption that the negation operation is not only predictable but *linear*. Previous work has found that linear operators are powerful enough to capture many hierarchical and relational structures (Paccanaro and Hinton, 2002; Bordes et al., 2014). Using examples (f, f') collected from the training set as described above, we compute the least-squares estimate $\hat{N} = \arg\min_N \sum ||Nf - f'||_2^2$. To evaluate, we collect example representations from the test set that are equivalent to known logical forms, and measure how frequently model behaviors $\textit{rep}(Nf)$ agree with the logical predictions $\textit{rep}(\neg e)$—in other words, how often the linear operator N actually corresponds to logical negation. Results are shown in the top portion of Table 2. Correspondence with the logical form is quite high, resulting in 97% agreement at the level of individual objects and 45% agreement on full representations. We conclude that the estimated linear operator $\hat{N}$ is analogous to negation in natural language. Indeed, the behavior of this operator is readily visible in Figure 3: predicted negated forms (in red) lie close in vector space to their true values, and negation corresponds roughly to mirroring across a central point.

In our final experiment, we explore whether the same kinds of linear maps can be learned for the binary operations of conjunction and disjunction. As in the previous section, we collect examples from the training data of representations whose denotations are known to correspond to groups of logical forms in the desired relationship—in this case tuples (e, f, e', f', e'', f''), where $\textit{rep}(e) = \textit{rep}(f)$, $\textit{rep}(e') = \textit{rep}(f')$, $\textit{rep}(e'') = \textit{rep}(f'')$ and either $e'' = e \wedge e'$ (conjunction) or $e'' = e \vee e'$ (disjunction). Since we expect that our operator will be symmetric in its arguments, we solve for $\hat{M} = \arg\min_M \sum ||Mf + Mf' - f''||_2^2$.

Results are shown in the bottom portions of Table 2. Correspondence between the behavior predicted by the contextual logical form and the model's actual behavior is less tight than for negation. At the same time, the estimated operators are clearly capturing some structure: in the case of disjunction, for example, model interpretations are correctly modeled by the logical form 92% of the time at the object level and 19% of the time at the denotation level. This suggests that the operations of conjunction and disjunction do have some functional counterparts in the RNN language, but that these functions are not everywhere well approximated as linear.

6 Conclusions

Building on earlier tools for identifying neural codes with natural language strings, we have presented a technique for exploring compositional structure in a space of vector-valued representations. Our analysis of an encoder–decoder model trained on a reference game identified a number of language-like properties in the model's representation space, including transformations corresponding to negation, disjunction, and conjunction. One major question left open by this analysis is what happens when multiple transformations are applied hierarchically, and future work might focus on extending the techniques in this paper to explore recursive structure. We believe our experiments so far highlight the usefulness of a denotational perspective from formal semantics when interpreting the behavior of deep models.

References

Jacob Andreas, Anca Dragan, and Dan Klein. 2017. Translating neuralese. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676* .

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Donald Davidson. 1967. Truth and meaning. *Synthese* 17(1):304–323.

Christopher Dircks and Scott Stoness. 1999. Effective lexicon change in the absence of population flux. *Advances in Artificial Life* pages 720–724.

Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. 2013. Learning distributions over logical forms for referring expression generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jakob Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*. pages 2137–2145.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2016. Towards multi-agent communication-based language learning. *arXiv preprint arXiv:1605.07133* .

Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic regularities in sparse and explicit word representations. pages 171–180.

Igor Mordatch and Pieter Abbeel. 2017. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908* .

Alberto Paccanaro and Jefferey Hinton. 2002. Learning hierarchical structures with linear relational embedding. In *Advances in Neural Information Processing Systems*. Vancouver, BC, Canada, volume 14, page 857.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*. pages 2244–2252.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. pages 3104–3112.

Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. 2003. Progress in the simulation of emergent communication and language. *Adaptive Behavior* 11(1):37–69.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2016. A joint speaker-listener-reinforcer model for referring expressions. *arXiv preprint arXiv:1612.09542* .