Translating Neuralese

Abstract

Several approaches have recently been proposed for learning decentralized deep multiagent policies that coordinate via a differentiable communication channel. While these policies are effective for many tasks, interpretation of their induced communication strategies has remained a challenge. Here we propose to interpret agents' messages by translating them. Unlike in typical machine translation problems, we have no parallel data to learn from. Instead we develop a translation model based on the insight that agent messages and natural language strings mean the same thing if they induce the same belief about the world in a listener. We present theoretical guarantees and empirical evidence that our approach preserves both the semantics and pragmatics of messages by ensuring that players communicating through a translation layer do not suffer a substantial loss in reward relative to players with a common language.

Introduction

Several recent papers have described approaches for learning deep communicating policies (DCPs): decentralized representations of behavior that enable multiple agents to communicate via a differentiable channel that can be formulated as a recurrent neural network. DCPs have been shown to solve a variety of coordination problems, including reference games BIBREF0 , logic puzzles BIBREF1 , and simple control BIBREF2 . Appealingly, the agents' communication protocol can be learned via direct backpropagation through the communication channel, avoiding many of the challenging inference problems associated with learning in classical decentralized decision processes BIBREF3 .

But analysis of the strategies induced by DCPs has remained a challenge. As an example, fig:teaser

depicts a driving game in which two cars, which are unable to see each other, must both cross an intersection without colliding. In order to ensure success, it is clear that the cars must communicate with each other. But a number of successful communication strategies are possible—for example, they might report their exact (x, y) coordinates at every timestep, or they might simply announce whenever they are entering and leaving the intersection. If these messages were communicated in natural language, it would be straightforward to determine which strategy was being employed. However, DCP agents instead communicate with an automatically induced protocol of unstructured, real-valued recurrent state vectors—an artificial language we might call “neuralese,” which superficially bears little resemblance to natural language, and thus frustrates attempts at direct interpretation.

We propose to understand neuralese messages by translating them. In this work, we present a simple technique for inducing a dictionary that maps between neuralese message vectors and short natural language strings, given only examples of DCP agents interacting with other agents, and humans interacting with other humans. Natural language already provides a rich set of tools for describing beliefs, observations, and plans—our thesis is that these tools provide a useful complement to the visualization and ablation techniques used in previous work on understanding complex models BIBREF4 , BIBREF5 .

While structurally quite similar to the task of machine translation between pairs of human languages, interpretation of neuralese poses a number of novel challenges. First, there is no natural source of parallel data: there are no bilingual “speakers” of both neuralese and natural language. Second, there may not be a direct correspondence between the strategy employed by humans and DCP agents: even if it were constrained to communicate using natural language, an automated agent might choose to produce a different message from humans in a given state. We tackle both of these challenges by appealing to the grounding of messages in gameplay. Our approach is based on one of the core insights in natural language semantics: messages (whether in neuralese or natural language) have similar meanings when they induce similar beliefs about the state of the world.

Based on this intuition, we introduce a translation criterion that matches neuralese messages with natural language strings by minimizing statistical distance in a common representation space of distributions over speaker states. We explore several related questions:

Our translation model and analysis are general, and in fact apply equally to human–computer and human–human translation problems grounded in gameplay. In this paper, we focus our experiments specifically on the problem of interpreting communication in deep policies, and apply our approach to the driving game in fig:teaser and two reference games of the kind shown in fig:bird-examples. We find that this approach outperforms a more conventional machine translation criterion both when attempting to interoperate with neuralese speakers and when predicting their state.

Related work

A variety of approaches for learning deep policies with communication were proposed essentially simultaneously in the past year. We have broadly labeled these as “deep communicating policies”; concrete examples include Lazaridou16Communication, Foerster16Communication, and Sukhbaatar16CommNet. The policy representation we employ in this paper is similar to the latter two of these, although the general framework is agnostic to low-level modeling details and could be straightforwardly applied to other architectures. Analysis of communication strategies in all these papers has been largely ad-hoc, obtained by clustering states from which similar messages are emitted and attempting to manually assign semantics to these clusters. The present work aims at developing tools for performing this analysis automatically.

Most closely related to our approach is that of Lazaridou16LanguageGame, who also develop a model for assigning natural language interpretations to learned messages; however, this approach relies on supervised cluster labels and is targeted specifically towards referring expression games. Here we

attempt to develop an approach that can handle general multiagent interactions without assuming a prior discrete structure in space of observations.

The literature on learning decentralized multi-agent policies in general is considerably larger BIBREF6 , BIBREF7 . This includes work focused on communication in multiagent settings BIBREF3 and even communication using natural language messages BIBREF8 . All of these approaches employ structured communication schemes with manually engineered messaging protocols; these are, in some sense, automatically interpretable, but at the cost of introducing considerable complexity into both training and inference.

Our evaluation in this paper investigates communication strategies that arise in a number of different games, including reference games and an extended-horizon driving game. Communication strategies for reference games were previously explored by Vogel13Grice, Andreas16Pragmatics and Kazemzadeh14ReferIt, and reference games specifically featuring end-to-end communication protocols by Yu16Reinforcer. On the control side, a long line of work considers nonverbal communication strategies in multiagent policies BIBREF9 .

Another group of related approaches focuses on the development of more general machinery for interpreting deep models in which messages have no explicit semantics. This includes both visualization techniques BIBREF10 , BIBREF4 , and approaches focused on generating explanations in the form of natural language BIBREF11 , BIBREF12 .

What's in a translation?

What does it mean for a message z_h to be a “translation” of a message z_r ? In standard machine translation problems, the answer is that z_h is likely to co-occur in parallel data with z_r ; that is,

$p(z_h |

z_r)$ is large. Here we have no parallel data: even if we could observe natural language and neuralese messages produced by agents in the same state, we would have no guarantee that these messages actually served the same function. Our answer must instead appeal to the fact that both natural language and neuralese messages are grounded in a common environment. For a given neuralese message z_r , we will first compute a grounded representation of that message's meaning; to translate, we find a natural-language message whose meaning is most similar. The key question is then what form this grounded meaning representation should take. The existing literature suggests two broad approaches:

Translation models

In this section, we build on the intuition that messages should be translated via their semantics to define a concrete translation model—a procedure for constructing a natural language $\leftrightarrow $ neuralese dictionary given agent and human interactions.

We understand the meaning of a message z_a to be represented by the distribution $p(x_a|z_a, x_b)$ it induces over speaker states given listener context. We can formalize this by defining the belief distribution $\beta $ for a message z and context x_b as:

Here we have modeled the listener as performing a single step of Bayesian inference, using the listener state and the message generation model (by assumption shared between players) to compute the posterior over speaker states. While in general neither humans nor DCP agents compute explicit representations of this posterior, past work has found that both humans and suitably-trained neural networks can be modeled as Bayesian reasoners BIBREF15 , BIBREF16 .

This provides a context-specific representation of belief, but for messages z and $z^{\prime }$ to have

the same semantics, they must induce the same belief over all contexts in which they occur. In our probabilistic formulation, this introduces an outer expectation over contexts, providing a final measure q of the quality of a translation from z to $z^{\prime }$:

$$&q(z, z^{\prime }) = \mathbb {E}\big [\mathcal {D}_{\textrm {KL}}(\beta (z, X_b)\ ||\ \beta (z^{\prime }, X_b))\ |\ z, z^{\prime }\big] \nonumber \\
&= \sum _{x_a, x_b} p(x_a, x_b | z, z^{\prime }) \nonumber \mathcal {D}_{\textrm {KL}}(\beta (z, x_b)\ ||\ \beta (z^{\prime }, x_b)) \nonumber \\
&\propto \sum _{x_a, x_b} p(x_a, x_b) \cdot p(z| x_a) \cdot p(z^{\prime } | x_a) \nonumber \\[-.9em]
&\qquad \qquad \ \cdot \mathcal {D}_{\textrm {KL}}(\beta (z, x_b)\ ||\ \beta (z^{\prime }, x_b));$$ (Eq. 15)

recalling that in this setting

$$&\hspace{-8.99994pt}\mathcal {D}_{\textrm {KL}}(\beta \ ||\ \beta ^{\prime }) = \sum _{x_a} p(x_a | z, x_b) \log \frac{p(x_a
| z, x_b)}{p(x_a | z^{\prime }, x_b)}
\nonumber \\
&\hspace{-8.99994pt}\propto \sum _{x_a} p(x_a, x_b) p(z| x_a) \log \frac{p(z|
x_a)}{p(z^{\prime } | x_a)} \frac{p(z^{\prime })}{p(z)}$$ (Eq. 16)

which is zero when the messages z and $z^{\prime }$ give rise to identical belief distributions and increases as they grow more dissimilar. To translate, we would like to compute $\textit {tr}(z_r) = \operatornamewithlimits{arg\,min}_{z_h} q(z_r, z_h)$ and $\textit {tr}(z_h) = \operatornamewithlimits{arg\,min}_{z_r} q(z_h, z_r)$. Intuitively, eq:q says that we will measure the quality of a proposed translation $z\mapsto z^{\prime }$ by asking the following question: in contexts where z is likely to be used, how frequently does $z^{\prime }$ induce the same belief about speaker

states as z ?

While this translation criterion directly encodes the semantic notion of meaning described in sec:philosophy, it is doubly intractable: the KL divergence and outer expectation involve a sum over all observations x_a and x_b respectively; these sums are not in general possible to compute efficiently. To avoid this, we approximate eq:q by sampling. We draw a collection of samples (x_a, x_b) from the prior over world states, and then generate for each sample a sequence of distractors $(x_a^{\prime }, x_b)$ from $p(x_a^{\prime } | x_b)$ (we assume access to both of these distributions from the problem representation). The KL term in eq:q is computed over each true sample and its distractors, which are then normalized and averaged to compute the final score.

[t] given: a phrase inventory L translate z $\operatornamewithlimits{arg\,min}_{z^{\prime } \in L} \hat{q}(z, z^{\prime })$

 $\hat{q}$ $z, z^{\prime }$ // sample contexts and distractors $x_{ai}, x_{bi} \sim p(X_a, X_b) \textrm { for $ i=1..n $}$ $x_{ai}^{\prime } \sim p(X_a | x_{bi})$ // compute context weights $\tilde{w}_i \leftarrow p(z | x_{ai}) \cdot p(z^{\prime } | x_{ai})$ $w_i \leftarrow \tilde{w}_i / \sum _j \tilde{w}_j$ // compute divergences $ k_i \leftarrow \sum _{x \in \lbrace x_a, x_a^{\prime }\rbrace } p(z|x) \log \frac{p(z|x)}{p(z^{\prime }|x)}\frac{p(z^{\prime })}{p(z)}$ $\sum _i w_i k_i$

 Translating messages

Sampling accounts for the outer $p(x_a, x_b)$ in eq:q and the inner $p(x_a|x_b)$ in eq:kl. The only quantities remaining are of the form $p(z|x_a)$ and $p(z)$. In the case of neuralese, these are determined by the agent policy $\pi _r$. For natural language, we use transcripts of human interactions to fit a model that maps from world states to a distribution over frequent utterances as discussed in

sec:formulation. Details of these model implementations are provided in sec:impl, and the full translation procedure is given in alg:translation.

Belief and behavior

The translation criterion in the previous section makes no reference to listener actions at all. The shapes example in sec:philosophy shows that some model performance might be lost under translation. It is thus reasonable to ask whether this translation model of sec:models can make any guarantees about the effect of translation on behavior. In this section we explore the relationship between belief-preserving translations and the behaviors they produce, by examining the effect of belief accuracy and strategy mismatch on the reward obtained by cooperating agents.

To facilitate this analysis, we consider a simplified family of communication games with the structure depicted in fig:simplegame. These games can be viewed as a subset of the family depicted in fig:model; and consist of two steps: a listener makes an observation x_a and sends a single message z to a speaker, which makes its own observation x_b , takes a single action u , and receives a reward. We emphasize that the results in this section concern the theoretical properties of idealized games, and are presented to provide intuition about high-level properties of our approach. sec:results investigates empirical behavior of this approach on real-world tasks where these ideal conditions do not hold.

Our first result is that translations that minimize semantic dissimilarity q cause the listener to take near-optimal actions:

Proposition 1

Semantic translations reward rational listeners.Define a rational listener as one that chooses the best

action in expectation over the speaker's state: $ U(z, x_b) = \operatornamewithlimits{arg\,max}_u \sum _{x_a} p(x_a | x_b, z) r(x_a, x_b, u) $

for a reward function $r \in [0, 1]$ that depends only on the two observations and the action. Now let a be a speaker of a language r , b be a listener of the same language r , and $b^{\prime }$ be a listener of a different language h . Suppose that we wish for a and $b^{\prime }$ to interact via the translator $\textit {tr}:
z_r \mapsto z_h$ (so that a0 produces a message a1 , and a2 takes an action a3). If a4 respects the semantics of a5 , then the bilingual pair a6 and a7 achieves only boundedly worse reward than the monolingual pair a8 and a9 . Specifically, if r0 , then

$$&\mathbb {E}r(X_a, X_b, U(\textit {tr}(Z)) \nonumber \\
&\qquad \ge \mathbb {E}r(X_a, X_b, U(Z)) - \sqrt{2D}$$ (Eq. 21)

So as discussed in sec:philosophy, even by committing to a semantic approach to meaning representation, we have still succeeded in (approximately) capturing the nice properties of the pragmatic approach.

sec:philosophy examined the consequences of a mismatch between the set of primitives available in two languages. In general we would like some measure of our approach's robustness to the lack of an exact correspondence between two languages. In the case of humans in particular we expect that a variety of different strategies will be employed, many of which will not correspond to the behavior of the learned agent. It is natural to want some assurance that we can identify the DCP's strategy as long as some human strategy mirrors it. Our second observation is that it is possible to exactly recover a translation of a DCP strategy from a mixture of humans playing different strategies:

Proposition 2

encoding=*-30Semantic translations find hidden correspondences. encoding=*0Consider a fixed robot policy $\pi _r$ and a set of human policies $\lbrace \pi _{h1}, \pi _{h2}, \dots \rbrace $ (recalling from sec:formulation that each $\pi $ is defined by distributions $p(z|x_a)$ and $p(u|z,x_b)$). Suppose further that the messages employed by these human strategies are disjoint; that is, if $p_{hi}(z|x_a) > 0$, then $p_{hj}(z|x_a) = 0$ for all $j \ne i$. Now suppose that all $q(z_r, z_h) = 0$ for all messages in the support of some $p_{hi}(z|x_a)$ and $\lbrace \pi _{h1}, \pi _{h2}, \dots \rbrace $0 for all $\lbrace \pi _{h1}, \pi _{h2}, \dots \rbrace $1 . Then every message $\lbrace \pi _{h1}, \pi _{h2}, \dots \rbrace $2 is translated into a message produced by $\lbrace \pi _{h1}, \pi _{h2}, \dots \rbrace $3 , and messages from other strategies are ignored.

This observation follows immediately from the definition of $q(z_r, z_h)$, but demonstrates one of the key distinctions between our approach and a conventional machine translation criterion. Maximizing $p(z_h | z_r)$ will produce the natural language message most often produced in contexts where z_r is observed, regardless of whether that message is useful or informative. By contrast, minimizing $q(z_h, z_r)$ will find the z_h that corresponds most closely to z_r even when z_h is rarely used.

The disjointness condition, while seemingly quite strong, in fact arises naturally in many circumstances—for example, players in the driving game reporting their spatial locations in absolute vs. relative coordinates, or speakers in a color reference game (fig:tasks) discriminating based on lightness vs. hue. It is also possible to relax the above condition to require that strategies be only locally disjoint (i.e. with the disjointness condition holding for each fixed x_a), in which case overlapping human strategies are allowed, and the recovered robot strategy is a context-weighted mixture of these.

Tasks

In the remainder of the paper, we evaluate the empirical behavior of our approach to translation. Our evaluation considers two kinds of tasks: reference games and navigation games. In a reference game (e.g. fig:tasksa), both players observe a pair of candidate referents. A speaker is assigned a target referent; it must communicate this target to a listener, who then performs a choice action corresponding to its belief about the true target. In this paper we consider two variants on the reference game: a simple color-naming task, and a more complex task involving natural images of birds. For examples of human communication strategies for these tasks, we obtain the XKCD color dataset BIBREF17 , BIBREF18 and the Caltech–UCSD Birds dataset BIBREF19 with accompanying natural language descriptions BIBREF20 . We use standard train / validation / test splits for both of these datasets.

The final task we consider is the driving task (fig:tasksc) first discussed in the introduction. In this task, two cars, invisible to each other, must each navigate between randomly assigned start and goal positions without colliding. This task takes a number of steps to complete, and potentially involves a much broader range of communication strategies. To obtain human annotations for this task, we recorded both actions and messages generated by pairs of human Amazon Mechanical Turk workers playing the driving game with each other. We collected close to 400 games, with a total of more than 2000 messages exchanged, from which we held out 100 game traces as a test set.

We use the version of the XKCD dataset prepared by McMahan15Colors. Here the input feature vector is simply the LAB representation of each color, and the message inventory taken to be all unigrams that appear at least five times.

We use the dataset of Welinder10Birds with natural language annotations from Reed16Birds. The model's input feature representations are a final 256-dimensional hidden feature vector from a compact bilinear pooling model BIBREF24 pre-trained for classification. The message inventory consists of the 50 most frequent bigrams to appear in natural language descriptions; example human traces are generated

by for every frequent (bigram, image) pair in the dataset.

Driving data is collected from pairs of human workers on Mechanical Turk. Workers received the following description of the task:

Your goal is to drive the red car onto the red square. Be careful! You're driving in a thick fog, and there is another car on the road that you cannot see. However, you can talk to the other driver to make sure you both reach your destinations safely.

Players were restricted to messages of 1–3 words, and required to send at least one message per game. Each player was paid $0.25 per game. 382 games were collected with 5 different road layouts, each represented as an 8x8 grid presented to players as in fig:drive-examples. The action space is discrete: players can move forward, back, turn left, turn right, or wait. These were divided into a 282-game training set and 100-game test set. The message inventory consists of all messages sent more than 3 times. Input features consists of indicators on the agent's current position and orientation, goal position, and map identity. Data is available for download at http://github.com/jacobandreas/neuralese.

Metrics

A mechanism for understanding the behavior of a learned model should allow a human user both to correctly infer its beliefs and to successfully interoperate with it; we accordingly report results of both “belief” and “behavior” evaluations.

To support easy reproduction and comparison (and in keeping with standard practice in machine translation), we focus on developing automatic measures of system performance. We use the available training data to develop simulated models of human decisions; by first showing that these models track

well with human judgments, we can be confident that their use in evaluations will correlate with human understanding. We employ the following two metrics:

This evaluation focuses on the denotational perspective in semantics that motivated the initial development of our model. We have successfully understood the semantics of a message z_r if, after translating $z_r \mapsto z_h$, a human listener can form a correct belief about the state in which z_r was produced. We construct a simple state-guessing game where the listener is presented with a translated message and two state observations, and must guess which state the speaker was in when the message was emitted.

When translating from natural language to neuralese, we use the learned agent model to directly guess the hidden state. For neuralese to natural language we must first construct a “model human listener” to map from strings back to state representations; we do this by using the training data to fit a simple regression model that scores (state, sentence) pairs using a bag-of-words sentence representation. We find that our “model human” matches the judgments of real humans 83% of the time on the colors task, 77% of the time on the birds task, and 77% of the time on the driving task. This gives us confidence that the model human gives a reasonably accurate proxy for human interpretation.

This evaluation focuses on the cooperative aspects of interpretability: we measure the extent to which learned models are able to interoperate with each other by way of a translation layer. In the case of reference games, the goal of this semantic evaluation is identical to the goal of the game itself (to identify the hidden state of the speaker), so we perform this additional pragmatic evaluation only for the driving game. We found that the most reliable way to make use of human game traces was to construct a speaker-only model human. The evaluation selects a full game trace from a human player, and replays both the human's actions and messages exactly (disregarding any incoming messages); the evaluation measures the quality of the natural-language-to-neuralese translator, and the extent to which the learned

agent model can accommodate a (real) human given translations of the human's messages.

We compare our approach to two baselines: a random baseline that chooses a translation of each input uniformly from messages observed during training, and a direct baseline that directly maximizes $p(z^{\prime } | z)$ (by analogy to a conventional machine translation system). This is accomplished by sampling from a DCP speaker in training states labeled with natural language strings.

Results

In all below, “R” indicates a DCP agent, “H” indicates a real human, and “H*” indicates a model human player.

Conclusion

We have investigated the problem of interpreting message vectors from deep networks by translating them. After introducing a translation criterion based on matching listener beliefs about speaker states, we presented both theoretical and empirical evidence that this criterion outperforms a conventional machine translation approach at recovering the content of message vectors and facilitating collaboration between humans and learned agents.

While our evaluation has focused on understanding the behavior of deep communicating policies, the framework proposed in this paper could be much more generally applied. Any encoder–decoder model BIBREF21 can be thought of as a kind of communication game played between the encoder and the decoder, so we can analogously imagine computing and translating “beliefs” induced by the encoding to explain what features of the input are being transmitted. The current work has focused on learning a purely categorical model of the translation process, supported by an unstructured inventory of translation

candidates, and future work could explore the compositional structure of messages, and attempt to synthesize novel natural language or neuralese messages from scratch. More broadly, the work here shows that the denotational perspective from formal semantics provides a framework for precisely framing the demands of interpretable machine learning BIBREF22 , and particularly for ensuring that human users without prior exposure to a learned model are able to interoperate with it, predict its behavior, and diagnose its errors.

Acknowledgments

JA is supported by a Facebook Graduate Fellowship and a Berkeley AI / Huawei Fellowship. We are grateful to Lisa Anne Hendricks for assistance with the Caltech–UCSD Birds dataset, and to Liang Huang and Sebastian Schuster for useful feedback.

Agents

Learned agents have the following form:

where h is a hidden state, z is a message from the other agent, u is a distribution over actions, and x is an observation of the world. A single hidden layer with 256 units and a $\tanh $ nonlinearity is used for the MLP. The GRU hidden state is also of size 256, and the message vector is of size 64.

Agents are trained via interaction with the world as in Hausknecht15DRQN using the adam optimizer BIBREF28 and a discount factor of 0.9. The step size was chosen as 0.003 for reference games and 0.0003 for the driving game. An $\epsilon $ -greedy exploration strategy is employed, with the exploration parameter for timestep t given by:

$

\epsilon = \max {\left\lbrace \begin{array}{ll}

(1000 - t) / 1000 \\

(5000 - t) / 50000 \\

0

\end{array}\right.}

$

As in Foerster16Communication, we found it useful to add noise to the communication channel: in this case, isotropic Gaussian noise with mean 0 and standard deviation 0.3. This also helps smooth $p(z|x_a)$ when computing the translation criterion.

Representational models

As discussed in sec:models, the translation criterion is computed based on the quantity $p(z|x)$. The policy representation above actually defines a distribution $p(z|x, h)$, additionally involving the agent's hidden state h from a previous timestep. While in principle it is possible to eliminate the dependence on h by introducing an additional sampling step into alg:translation, we found that it simplified inference to simply learn an additional model of $p(z|x)$ directly. For simplicity, we treat the term $\log (p(z^{\prime }) / p(z))$ as constant, those these could be more accurately approximated with a learned density estimator.

This model is trained alongside the learned agent to imitate its decisions, but does not get to observe the recurrent state, like so:

Here the multilayer perceptron has a single hidden layer with $\tanh $ nonlinearities and size 128. It is also trained with adam and a step size of 0.0003.

We use exactly the same model and parameters to implement representations of $p(z|x)$ for human speakers, but in this case the vector z is taken to be a distribution over messages in the natural language inventory, and the model is trained to maximize the likelihood of labeled human traces.