Deep compositional models for robotic planning and language

Yen-Ling Kuo, Andrei Barbu, and Boris Katz

***Abstract*— We demonstrate how compositionality, the key attribute of natural language, can be encoded into a series of recurrent neural networks which can guide an agent to carry out never-before-seen plans. Using little training data, our model simultaneously learns to embed features from the environment into a space useful for planning and automatically discovers the concepts underlying different plans. A plan, a sentence fragment describing a command that a robot should carry out, is parsed into RNNs, one for each constraint or concept implied by the command, which are used to construct a novel compositional sampling-based robotic planner. We train these RNNs jointly, each training example providing a path and a list of which models should constrain that path, without manually annotating what features each model should refer to in either the path or the surrounding environment. At test time, we take as input a natural-language sentence, decompose it into a series of RNNs which are structured according to the parse of the sentence, and demonstrate how these enable the robot to carry out novel linguistic commands.**

In order to survive, humans and animals must be able to carry out new plans in new environments. To do so, they bring to bear their knowledge of previous plans and environments: how those plans were broken down into components and what they expects to observe around them when carrying out a plan. This ability is fundamental to data-efficient learning, knowing most of the components of a plan allows you to more easily learn the rest. It is also critical to grounding language in robotic planning for two reasons; first, language is inherently compositional which must be reflected in the compositions of the models which are used, and second, one must disentangle which parts of a plan different words and phrases refer to. We demonstrate how to give robots this ability to carry out new plans specified by sentences by breaking plans down into compositions of previously-learned primitives trained in a weakly-supervised setting.

The planner presented here combines a sampling-based planner, RRT [1], with a collection of RNNs, recurrent neural networks [2, 3, 4]. It builds on our two earlier contributions where we develop a non-compositional variant of RRT combined with a single RNN [5] and where we show a sentence can be used to create a series of temporal models which encode its meaning [6]. Given a command to a robot, that command is broken down into constraints or individual instructions, each is looked up in a lexicon mapping instructions to RNNs, the selected RNNs are then collected and combined with the sampling-based planner. At each time step, conditioned on local observations around the robot, the preferred direction according to the original sampling-based planner, and the states of all of the RNNs, a new direction is sampled. Each RNN predicts the parameters of a multivariate normal distribution that determine the new location to move to in the configuration space. Less certain or temporarily irrelevant RNNs, for example an RNN that attempts to avoid a wall while far away from any walls, will produce a large covariance matrix rendering their preferred directions irrelevant. A new location is sampled from the joint distribution of all RNNs, the search tree is extended to that location, each of the RNNs adds a new state for that location connected to the state of the previous location, and the process repeats. The result is a search tree in which we find a high-likelihood path that ends in node which is likely to be the end of a plan.

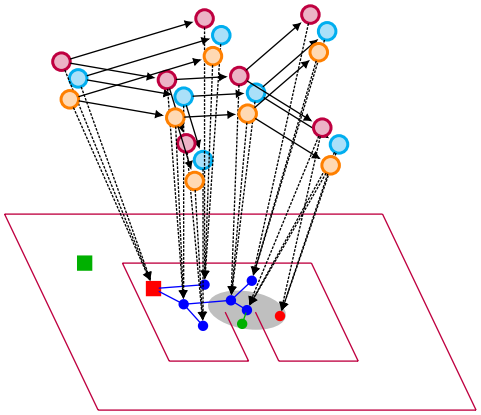

Fig. 1. An example of the compositional planner in action. The robot starting at the red square is instructed to find the green square by going through the narrow channel and avoiding any walls. The human instructions not only guide the robot but provide crucial information that helps it avoid getting stuck in the bug trap. Each constraint in the human instruction corresponds to a different RNN that is added to the planner. New nodes in the search tree, are sampled jointly from all RNNs while each RNN maintains its own state and observes the local environment. All the RNNs work together to guide the robot to execute a new plan it has never encountered before.

While the configuration space, action space, features, reasoning, and internal state of the planner are all continuous, we build symbolic notions into the structure of the planner itself while also having a probabilistic interpretation for its behavior. The particular RNNs used to constrain a plan derive from the command given; few RNNs are used for any one plan. The RNNs diverge in their states when the search tree diverges; different hypotheses about spatial paths lead to different hypotheses about the internal states of the RNNs. This structures reasoning in a very intuitive and easy to visualize way. Moreover, at each time step, each RNN predicts

This work was supported by the Center for Brains, Minds and Machines, CBMM, NSF STC award 1231216, as well as The Toyota Research Institute and the MIT-IBM Brain-Inspired Multimedia Comprehension project.

Computer Science and AI Laboratory, MIT
{ylkuo,abarbu,boris}@mit.edu

its likelihood of accepting the current state as the end of a path; this is used to find terminal nodes in the search tree but also lends interpretability to why the robot might decide to stop at a certain location. Since each RNN also produces a distribution over the next move, each has some level of interpretability on its own outside of the model as a whole. This means its impact on a particular path can be measured — one can in principle determine which RNNs led to a particular decision. Together these features not only speed up inference but also provides significantly more interpretability compared to black-box RL or even large POMDP approaches.

At training time, the planner is weakly-supervised, much like an RL agent that is given a single reward after a long sequence of actions. The planner here has both a harder and an easier task. It must not only learn the meanings of different plan components — their temporal evolution and expected observations — but also learn to disentangle what each RNN, i.e., word, means. When doing so, being able to take advantage of the context of a plan and the already-learned plan components is critical. Intuitively, if you already know what a plan means, acquiring the rest of it should be much easier. Because of this, we train plans jointly. At each training step, we provide an entire path from a source to a goal along with the models which contribute to that path without indicating when each model should hold or what its meaning should be. We then jointly update the parameters of all models.

In fig. 1, we show an example of planning with multiple sequence models; an extension of our earlier work [5]. The robot begins at the red square and is told to find a green square while going through the narrow channel and staying away from any walls. The search tree, shown as blue circles, is mirrored by a collection of sequence models, RNNs. Each sequence model corresponds to one portion of the command given to the robot, for example, the node in purple might bias toward the channel, the node in orange might detect toward the green square, and the node in might bias against being near walls. There is nothing in the structure of the planner to distinguish different models for different concepts, each guides the planner equally according to its parameters and each is independent of the others. When expanding, a free-space sample is drawn, steered toward, and the resulting node, shown as a red circle, is used to find the closest node in the tree; as in RRT. The models, with states corresponding to that closest node, observe this free-space sample, the path leading to this node (technically including all free-space samples originally produced when creating that path; omitted for clarity), each of their prior states along this path, local visual or map features, shown in gray, and predict a modified direction, shown in green, which is then connected to the search tree. A new hybrid continuous and symbolic state for each sequence model is also predicted and connected. The state contains a continuous vector, allowing the RNNs to store knowledge going forward, and a symbolic portion, a boolean representing the relevance of the RNN. This process co-evolves a planner with multiple sequence models that encode a novel command that might never have been seen before and together execute that command.

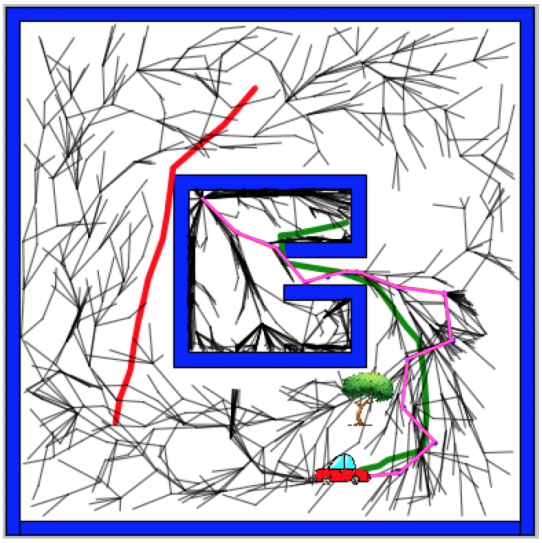

Fig. 2. Given the sentence "Weave through green path to reach the car" we generate the above path and search tree. The agent needs to escape the bug trap while satisfying the constraints specified by the sentence. The magenta line is the found solution. The black lines are the search tree. Note that the RNNs guide the planner to sample densely around the green path, do indeed weave through it, and choose a solution that ends near the car.

Given a sentence which describes a plan, we parse that sentence using an off-the-shelf syntactic parser. Similarly to our earlier approach to recognizing sentences that describe videos, given a lexicon of trained RNNs we look up each word and select the appropriate RNN for it and given the parse we determine a linking function which connects RNNs to the track and to each other [6]. This collection of RNNs constrain the track the robot must follow and the sampling process combined with RRT as descried above is used to produce allowable paths. In this manner, given a sentence, we can produce a robotic plan described by it. Figure 2 is an example solution produced by our planner for the sentence "Weave through green path to reach the car". We are at present looking at how such models can be made more interpretable and how to express complex relationships between words such as modification by having networks which affect each other's parameters or internal states.

References

[1] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[2] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[5] Y.-L. Kuo, A. Barbu, and B. Katz, "Deep sequential models for sampling-based planning," in *IROS*, 2018.

[6] H. Yu, N. Siddharth, A. Barbu, and J. M. Siskind, "A compositional framework for grounding language inference, generation, and acquisition in video." *JAIR*, 2015.