Functional Object-Oriented Network: Construction & Expansion

David Paulius, Ahmad Babaeian Jelodar, and Yu Sun
Department of Computer Science & Engineering
University of South Florida
Tampa, FL, 33620.
Contact Email: davidpaulius@mail.usf.edu, yusun@cse.usf.edu

I. Abstract

Our work presents the *functional object-oriented network* (FOON), a structured knowledge representation which can be used for representing object-motion affordances as observed in a variety of activities. Ideally, a FOON can be learned from observations of human activities either from instructional videos or from demonstration. From these sources, we can learn about the objects and manipulative motions needed to produce a certain effect observed as state changes; an objects state before and after a motion action is executed is captured in an atomic unit which we refer to as functional units (shown in Figure 1). A FOON will generally be comprised of many of such structures.

A. Overview of FOON

A FOON is a *bipartite network* [10] containing two types of nodes: *object* nodes and *motion* nodes. Object nodes are identified by their object type, their observable state and, if it is a container, its ingredient content. Motions are identified by a type. This graphical structure is similar to Petri Nets (PNs) [13] [9], where object nodes are parallel to place nodes and motion nodes to transition nodes. It is also a directed, acyclic graph, meaning that there exists edges within the graph that indicate the flow or sequence in an object's change of state.

A *functional unit* is the basic learning unit of a FOON, and this unit reflects a single action in a manipulation task. A collection of such units that reflect an entire activity is referred to as a *subgraph*. A functional unit contains three parts: input and output object nodes (much like input and output places in PNs) and a motion node. The motion node describes the action that causes an object's change in state from a single manipulation action. Functional units make our FOON behave like a transition system like PNs, as they can be traced along to show the series of steps and objects required to make an object of a particular state. With a collection of subgraphs, we can use a merging procedure to combine the knowledge from all sources to create a *universal FOON*. For a more detailed explanation of the basics of a FOON and how we represent a FOON, we refer readers to [12].

B. Learning a Generalized FOON

In our previous work [12], we demonstrated how we can use knowledge from several sources to produce novel and flexible procedures for solving problems using knowledge retrieval taking ideas from graph searching principles. We can build a FOON comprising of knowledge from multiple video sources (where each video is annotated by hand, producing a subgraph a set of functional units which describe the procedure needed to create a single meal) and merge their contents into one single universal FOON. With a universal FOON, we can solve a manipulation problem through a knowledge retrieval process given a target node and a set of objects which are available to the robotic system. The outcome of this graph search is a *task tree*, a series of functional units (or simply steps) which can be executed in sequence to create the target object node. The knowledge contained in task trees can span several video sources, hence the novelty of the manipulations.

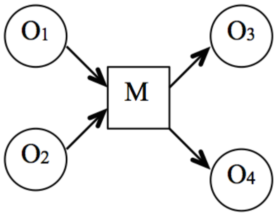

Fig. 1. The basic functional unit with two input objects, an interactive motion node, and two output objects.

Despite the ability to produce novel task trees, our FOON is not designed to handle unknown objects or unfamiliar states for known objects due to a lack of information from source videos. We would solely be limited to the object states we have observed in videos. Our searching algorithm will only work when we have the exact items needed to create a specific goal object node. In our present work, we are currently investigating a means of generalizing knowledge so we can apply it to those unknown objects which are similar to those which are represented in FOON without the need for annotating additional sources of knowledge. We can either expand our network using object similarity to create new functional units from those we already have or we can abstract the knowledge even further using object categories. With our first method, we expand our network by adding new functional units based on those we have seen already but extending them to other objects which are similar to them.

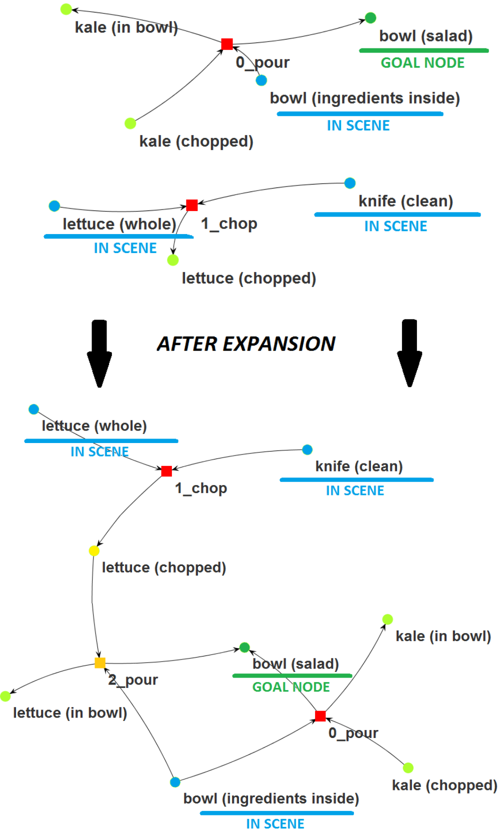

Fig. 2. An example of how expansion helps us to add knowledge which can be useful for solving a situational problem. In this example, we wish to make a salad (goal node denoted in dark green) using lettuce and other items in the environment (denoted in blue); however, we initially only have knowledge on making salads with kale. Using similarity, we can connect the knowledge of chopping lettuce and adding it to a bowl with other ingredients to make a salad.

An example of this is shown in Figure 2, where we can apply the knowledge we have of cutting lettuce to a similar object kale. The second method uses a similar principle, but instead of expanding the network, we condense the network to a much more generalized state by substituting specific objects with object categories. For example, objects like "tomato" and "orange" can be generalized to a category "fruits". The issue with object similarity is determining a method for measuring how similar two objects are; we solve this by using WordNet [4] and semantic similarity metrics (specifically Wu-Palmers [18] [3] metric).

C. Evaluation of Proposed Methods

We compare the efficacy of expansion and abstraction through experiments. We simulate random kitchen environments with which we try to find task trees for 50 random objects over 10 trials using each network type: 1) a regular FOON with knowledge from just 65 videos, FOON-REG, 2) an expanded version of our regular FOON (by adding new object nodes with similarity), FOON-EXP, and 3) a generalized, compacted and abstracted version of our regular FOON (using object types rather than objects), FOON-GEN.

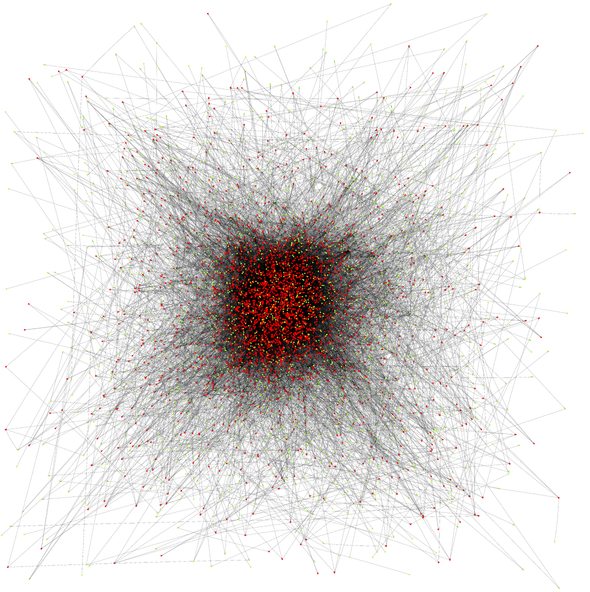

Fig. 3. Our current universal FOON that is constructed from 65 instructional videos.

The FOON that performs best will be indicated by the number of successful task trees found (i.e. the number of objects for which we found a tree out of the 50 goal nodes). We have shown that theoretically, we should be able to use a similar subset of objects in the same way, i.e. they should have similar affordances attributed to them. By using object categories as in FOON-GEN, we can find more task trees since the knowledge is generalized.

D. Future Work

Our present (and future) goals are to implement the use of FOON in problem solving with real robots and to investigate the implications of using expansion or abstraction of a FOON in real-world scenarios. A drawback to using such methods is the drastic increase in size of functional units because each object-state combination must be expressed individually if we use an expanded version of FOON. We can compact the functional units based on object types; however, we are not able to represent this knowledge as formal expressions. The main question we wish to answer is as follows: how do we extend manipulation knowledge (grasp types, motion trajectories, etc.) of objects we have in FOON to those we do not know in the real world? We are also exploring event recognition for annotating new videos and sources of information using probabilities based on knowledge contained in FOON or other datasets and apply them to a system which can be used for identifying objects in a scene and/or the action taking place. A deep learning object and affordance recognition approach would be taken to solve this problem of recognizing activities for semi-automatic construction.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1421418.

References

[1] E. Aksoy, A. Abramov, F. Worgotter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In *IEEE Intl. Conference on Robotics and Automation*, pages 398–405, 2010.

[2] B. D. Argall, S. Chernova, and et al. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[4] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[5] J.J. Gibson. The theory of affordances. In R. Shaw and J. Bransford, editors, *Perceiving, Acting and Knowing*. Hillsdale, NJ: Erlbaum, 1977.

[6] Y. Huang and Y. Sun. Generating manipulation trajectories using motion harmonics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4949–4954, 2015.

[7] Dominik Jain, Lorenz Mosenlechner, and Michael Beetz. Equipping robot control programs with first-order probabilistic reasoning capabilities. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3626–3631. IEEE, 2009.

[8] G. D. Konidaris, S.R. Kuindersma, R.A. Grupen, and A.G Barto. Robot learning from demonstration by constructing skill trees. *Intl J Robotics Research*, 31(3): 360–375, 2012.

[9] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[10] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, USA, 2010. ISBN 0199206651.

[11] E. Oztop, M. Kawato, and M. Arbib. Mirror neurons and imitation: a computationally guided review. *Epub Neural Networks*, 19:254–271, 2006.

[12] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun. Functional object-oriented network for manipulation learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[13] C. Adam Petri and W. Reisig. Petri net. *Scholarpedia*, 3(4):6477, 2008. revision 91646.

[14] Shaogang Ren and Yu Sun. Human-object-object-interaction affordance. In *Workshop on Robot Vision*, 2013.

[15] G. Rizzolatti and L. Craighero. The mirror neuron system. *Ann. Rev. Neurosci.*, 27:169–192, 2004.

[16] G. Rizzolatti and Craighero L. Mirror neuron: A neurological approach to empathy. In Jean-Pierre Changeux, Antonio R. Damasio, Wolf Singer, and Yves Christen, editors, *Neurobiology of Human Values*. Springer, Berlin and Heidelberg, 2005.

[17] Y. Sun, S. Ren, and Y. Lin. Object-object interaction affordance learning. *Robotics and Autonomous Systems*, 2013.

[18] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.