Merging New Knowledge Into FOON network and Searching for Cooking Instructions

Nicholas Eales

***Abstract*—The purpose of this project is to gain an understanding of how to represent knowledge for robots. To do this we used ”Foon” or Functional Object-Oriented Network. We annotated videos as foon graphs using the predetermined format, Derived a merging algorithm for two separate foon files and, Derived a search algorithm for retrieving a task tree from a universal foon.**

I. Introduction

FOON the Functional Object-Oriented Network, was the primary focus of this project. This network contains functional units which consist of input nodes, a motion, and output nodes. The foon we did our work on contained information about cooking recipes, with the universal foon containing several recipes in one. To begin the project we created our own foon file for single recipes using a given template and a video. After this we developed a merging algorithm for multiple foons with the purpose of merging our newly created foon to the universal foon. Lastly we developed a search algorithm for the purpose of discovering what meals could be made from a given kitchen. This work was done as a continuation of the previous work done about observing manipulations seen in daily household tasks and routines (See:[1], [2]) and work done analyzing motions and trajectories. (See:[3], [4], [5], [6])

II. Video Annotation

For the video annotation we were provided two partially annotated files and two videos. To finish the annotation we were required to match the format of currently existing foon files. the format of a functional unit in foon consists of object, state and motion lines. and object line has an id, the object name, and a 1 or 0 describing if the object is moving or not. An object line is always accompanied by a state node which contains an id, the objects state, and a list of ingredients if necessary. finally their are motion lines which contain an id, the motion name, and a start and end time. Foon formatting requires this information to be separated by tabs and for each functional unit to be separated by a ”//” string. This is a visualization of the two annotated graphs. After The proper edits were made to the files using the videos and other provided information a merging algorithm was required.

Course Name: Intro to Robotics. The work is supervised by Professor Yu Sun and TA David Paulius.

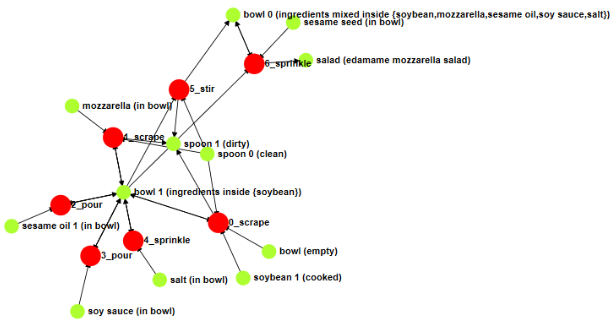

Fig. 1. Graph visualization of the annotated file for the edamame mozzarella salad foon file.

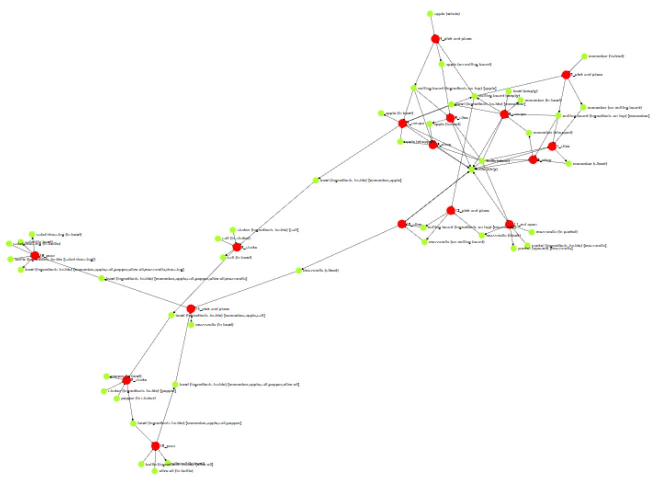

Fig. 2. Graph visualization of the annotated file for the cucumber apple salad foon file.

III. Merging Algorithm

The purpose of the merging algorithm is to integrate our new foon files into the universal foon. The merging process is relatively simple, just take the two files and insert the functional units from one to the other skipping any units that are already in the resulting file. I choose to implement this algorithm in python. To implement this algorithm I read both of the files, then divided the two files into functional units by splitting along the ”//” string. I then checked all functional units in one file against the other and added it to the file if it was not already present. To test this algorithm I used the two files I annotated for part one and final testing was done with the universal foon. This is a visualization of the graphs in figure 1 and 2 merged.

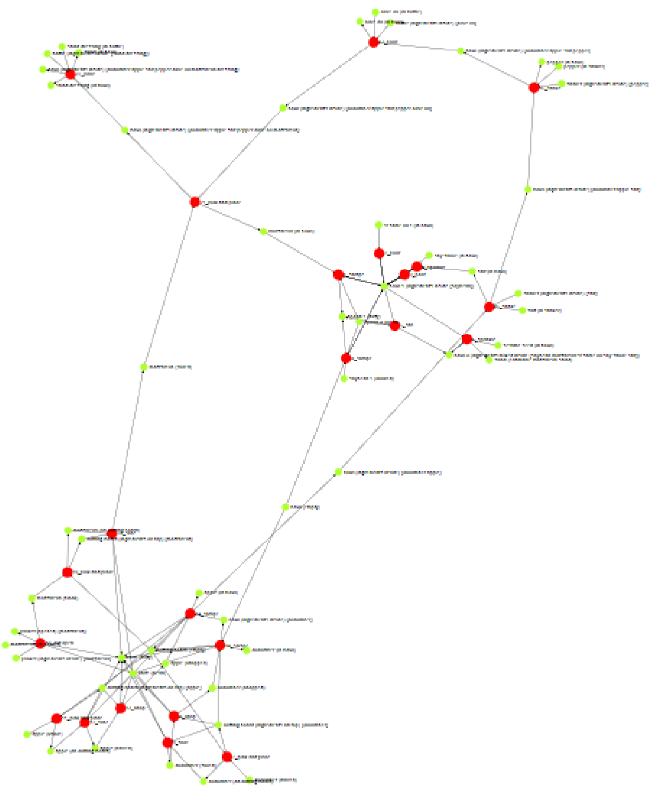

Fig. 3. Graph visualization of the merged foon files from part 1.

IV. Searching Algorithm

The purpose of the searching algorithm is to navigate the universal foon and find the required ingredients to make a given meal. To do this we were given a kitchen file and a meals file. The kitchen file contains ingredients, some of which can be several steps into a single recipe. The meal file contains the a goal to reach in foon, some of the goals are the ends of recipes and some are earlier in a given recipe. To develop this algorithm we were given psudeocode for the search. I started my algorithm by breaking all the functional unit into input nodes, motions, and output nodes. Each of the nodes was simply a string of lines like those from the annotated video file. The object nodes contain the object line as a string concatenated to its state line, while the motion line is just a single line. All comparisons done for the search were done by comparing these strings together. After the algorithm was competed I tested it on the meals and kitchen given. A graph of results is given below as well as the resulting tree from one meal.

Goal #	Goal Name	Tree Found?	FU in tree
Goal 1:	O13 pan S251 pigs in a blanket{dough, cheese, sausage}	Yes	13
Goal 2:	O16 shrimp S62 cooked	Yes	7
Goal 3:	O12 bread S116 garlic bread{butter, parsley, garlic, salt, pepper}	Yes	18
Goal 4:	O39 potato S138 hash browns{onion, parsley, flour, corn starch, pepper, salt, egg}	N/A	N/A
Goal 5:	O15 plate S234 pancake {flour, sugar, baking, powder, salt, milk, egg, vegetable oil}	No	N/A
Goal 6:	O103 bean S66 cooked black beans {olive oil, jalapeno, onion, garlic, salt, pepper, water, bay leaf}	Yes	41
Goal 7:	O59 sandwich S246 peanut butter and jelly {peanut butter, jelly}	Yes	3
Goal 8:	O59 sandwich S56 club sandwich {mayonnaise, ham, turkey, cheese, bacon, lettuce, tomato, toothpick}	No	N/A
Goal 9:	O2 bowl S269 salad {mayonnaise, sour cream, sugar, salt, carrot, lemon juice}	Yes	50
Goal 10:	O101 smoothie S321 watermelon-passion smoothie {watermelon, tea, strawberry, syrup}	No	N/A

Fig. 4. Table of values collected from the search algorithm.

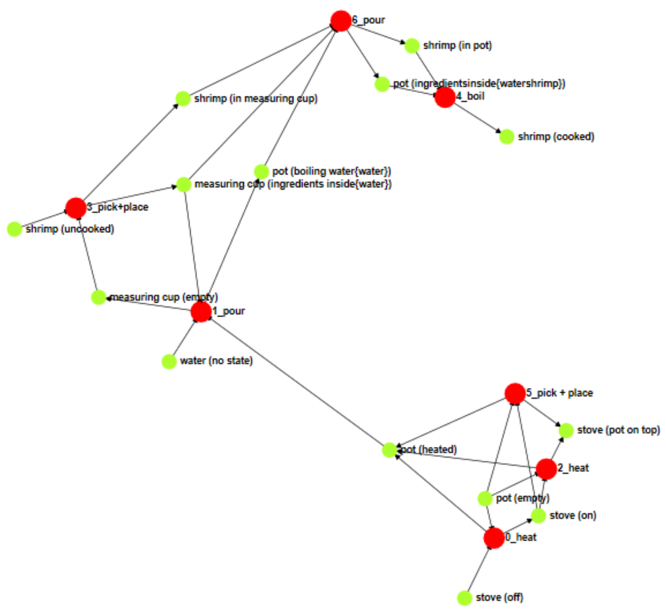

Fig. 5. Graph visualization of a retrieval tree from the Cooked Shrimp meal.

V. Conclusion

The purpose of this project was to gain an understanding of how to represent knowledge for robots. To do this we used "Foon" or Functional Object-Oriented Network. In step one We annotated videos and visualized them as foon graphs using the predetermined format. In the second step we derived a merging algorithm to bring together two separate foon files and to integrate a foon file into the universal foon. In the third and final step we derived a search algorithm for retrieving a task tree from the universal foon. This project provided an interesting look at how knowledge is represented in a robotic system.

REFERENCES

[1] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2655–2662.

[2] Y. Huang and Y. Sun, "A dataset of daily interactive manipulation," *arXiv preprint arXiv:1807.00858*, 2018.

[3] ——, "Learning to pour," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 7005–7010.

[4] ——, "Datasets on object manipulation and interaction: a survey," *arXiv preprint arXiv:1607.00442*, 2016.

[5] Y. Huang, M. Bianchi, M. Liarokapis, and Y. Sun, "Recent data sets on object manipulation: A survey," *Big data*, vol. 4, no. 4, pp. 197–216, 2016.

[6] Y. Huang and Y. Sun, "Generating manipulation trajectory using motion harmonics," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4949–4954.