Robust Semantic Planning for Robots with Learned Causal Commonsense Knowledge

***Abstract*—**

***Index Terms*—NLP, Robotics**

I. Problem Formulation

The problem is **learning common sense knowledge pertaining to under-specified tools for plan synthesis**. Candidate examples:

- Carry the fruits to the kitchen [tray usage is implicit]
- Hang the painting on the wall [needs nail/hammer]
- Pack away the books [using a box to pack is implicit]
- Illuminate the room [turning the switch on or opening the window]
- Pour the liquid in the barrel [funnel use]
- Clean the spilled water the floor [using a mop]
- Turn on the appliance [checking the on-switch, if the wire is connected]
- Bring my water bottle without spilling [placing a lid or a cap]
- Exit the room [look for the door]
- Stick the painting on wall [needs glue]
- Join the torn sheet of paper [needs tape]
- Collect all the objects on the floor and throw in the bin [using a trash bag is useful]

We will approach more abstract knowledge (e.g. feed me, illuminate the room) subsequently.

II. Related Work

[Commonsense Reasoning System] Chen et al. [1] develop a Language-Model based Commonsense Reasoning (LCMR) method which enables robots to listen to incompletely specified instructions and uses environmental context and commonsense approach to fill in missing information. They first convert the raw instruction to 'verb frames'. A verb frame is set containing predicate (verb) and semantic roles. If the human says "pour me some water". The verb frame (pour, Theme: water, Destination: ?) is generated using Semantic Role Labeling (SRL) by He et al. [2]. The original instruction does not specify where to pour into, LCMR infers out destination on its own, i.e. a cup through environment observations. They use YouCook2 [3] and Now You're Cooking datasets as training corpora. They used motion planning toolkit Moveit! [4] to compute a motion plan for the robot to accomplish each task (based on ROS). System Model: (1) Speech Recognition using Google Cloud API, (2) Detect objects in environment using Mask R-CNN, (3) Predicate Argument Parsing using RNN to (a) conference resolution (b) SRL (c) Rule based mapping system using PropBank to convert to verb frame and do pruning. (3) Complete verb frame by plugging in ? in verb frames by objects in vicinity based on probability ranking (trained using croudsourcing Amazon Turks and used RNN model with GRUs). (4) Motion planning based on complete verb frames. Strength: they compared with other works like Co-occur, Word2Vec, ConceptNet and their results are much better. Weakness: Only restricted to knowledge from environment, can extend this to web. Dataset used is a recipe dataset and looks like testing is also done mostly on related tasks. Dataset is noisy. Can try finetuning BERT on a different dataset. Assume only one of the two targets for the verb frame will be missing.

[Web Data sources] Tenorth et al. [5] describe and discuss use of World Wide Web based information for autonomous service robots. They show web sources like: (1) *ehow.com* and *wikihow.com* which give step wise instructions for millions of everyday activities like making pancakes. (2) Lexical databases like *wordnet.princeton.edu* group verbs, adverbs and nouns semantically into sets of synonyms (synsets), which are linked to concepts in encyclopedic knowledge bases like *opencyc.org* (ontological relationships). Knowledge is represented as first order logic to parse say instuctions on how to make pancakes into usable forms. (3) Common sense knowledge bases like *openmind.hri-us.com* to tell like what are pancakes, tools, kitchen, stove, etc. (4) Object appearance like *germandeli.com*, *images.google.com* to show how stove looks like, or other things like pancake or pancake mix look like. (5) Object shape in 3D CAD models to identify objects in vicinity *sketchup.google.com/3dwarehouse/* (6) Object properties like those extracted from shopping websites *germandeli.com*. Use wikihow to form stream of instructions however implicit instructions difficult to infer.

[Knowledge representation] Saxena et al. [6] describe a knowledge engine (called RoboBrain) which learns and shares knowledge representations for robots to carry out variety of tasks. There are multiple sources of *knowledge* including physical interaction with environment, the Internet and several learned representations from various robotic research groups. They demonstrate its use in three important research areas: grounding natural language, perception, and planning. RoboBrain enables sharing from multiple sources by representing the knowledge in a graph structure. Traversals on the RoboBrain graph allow robots to gather the specific information

Identify applicable funding agency here. If none, delete this.

they need for a task. This includes the semantic information, such as different grasps of the same object, as well as the functional knowledge, such as spatial constraints (e.g., a bottle is kept on the table and not the other way around). RoboBrain accepts new information in the form of set of edges, which they call a feed. A feed can either be from an automated algorithm crawling the Internet sources or from one of RoboBrains partner projects. After adding the feed to the graph they perform inference to update the graph based on this new knowledge. The inference outputs a sequence of graph operations which are then performed on the graph. These graph operations modify the graph by adding new nodes or edges to the graph, deleting nodes or edges from the graph, merging or splitting nodes, etc. Strength: A well structured way of consolidating all sources of knowledge into a single model for ease of maintenance and query.

[Partially known workspaces/tasks] Nyga et al. [7] show how to interpret and execute high-level tasks conveyed using natural language in partially-observed environment or missing background knowledge. The ability to infer missing plan constituents enables information-seeking actions such as visual exploration or dialogue with the human to acquire new knowledge to fill incomplete plans. They present a probabilistic model that enables a robot to infer symbolic plans from natural language commands in scenarios where the workspace is partially observed or the robots background knowledge is insufficient. Robots knowledge is encoded in terms of instantiated predicates On(block, table), Left(box, robot) etc. Robot can affect the environment by executing actions such as moving an object from a source location to a destination Move(cup, table, tray). Based on sparse evidence and partially observable environment, the robot decides its actions. It also gains missing background knowledge from online interaction.

[Common sense (size of objects)] Forbes et al. [8] try to infer relative physical knowledge of actions and objects along five dimensions (size, weight, speed etc.). They acquire this knowledge from unstructured natural language and try to overcome reporting bias. Relations are inferred based on the first order relations implied by physical verbs through a probabilistic model. They have created a VerbPhysics dataset having 3500 object pairs along with the five knowledge dimensions. Relations are started from a seed knowledge set, and grown using the similarity of objects occuring on the same side of verbs and relations between verbs. Knowledge also captures metaphorical meanings though intended very only real objects through action verbs. The model also infers nonsensical relations like “for” is used as a verb which may be followed by a duration or purpose instead of an object. There may be some errors due to polysemous use of verbs.

[Grounding natural language to plans/objects] Tellex et al. [9] give a model to perform navigation in semi structured environments, specifically a forklift in the warehouse. The model instantiates a probabilistic graphical model for a natural language command through the commands structure. Evaluation is done on a robot simulators with turkers. The structure of the language is exploited using SDCs (Spatial Description Clauses). The system infers a grounding on the basis of the natural language command given to it and then executes it as a plan. The model is basically a CRF where the search space has been reduced by manually specyfing groups of SDCs into EVENT,PATH,PLACE,OBJECT. Beam search has been used to make the search tractable. Corpus used for training has been generated by showing AMTs videos of the forklift working and asking them to give the command they would give to an expert to execute that. SDCs were then manually annotated. They found that commands executed with a high confidence were almost always correct so the system knew when it should ask for confirmation. The system has difficulty recognizing commands with a frame of reference. Dataset is small so some words(prepositions) occur rarely. As dataset is small unsupervised learning should be used. Complex linguistic devices (negation, abstract objects etc) are not supported.

III. System Architecture

IV. Data Sources

1) Instructions dataset (WikiHow): https://github.com/mahnazkoupaee/WikiHow-Dataset
2) Tools use for task completion [10]: Link: https://bair.berkeley.edu/blog/2019/04/11/tools/, Video: https://sites.google.com/view/anonymous-dgvf/home, They used unsupervised learning. In particular, the robot autonomously collects data in two different ways: by taking random sequences of actions and by sampling from the action proposal model introduced in the previous section. The latter allows the robot to grasp at tools and move them randomly. This experience is crucial to learn about multi-object interactions.
3) RoboTurk [11]: Robotic Manipulation Dataset through Human Reasoning and Dexterity. It is a dataset containing 2114 total demonstrations of different tasks. We can pick those requiring tools. More info at http://roboturk.stanford.edu/realrobotdataset
4) HowToKB dataset: HowToKB is a large-scale knowledge base which represents how-to (task) knowledge. Each task is represented by a frame with attributes for parent task, preceding sub-task, following sub-task, required tools or other items, and linkage to visual illustrations. Their methodology first applies Open-IE techniques to WikiHow articles, in order to extract - noisy and ambiguous - candidates for task and sub-tasks. Subsequently, they use judiciously devised clustering techniques to clean and organize these candidates, and to infer attribute values. To canonicalize tasks and sub-tasks, they leverage word embeddings to distinguish different meanings of the same phrase (e.g., ”use keyboard”). Dataset: https://github.molgen.mpg.de/cxchu/HowToKB,
   More info: https://www.mpi-inf.mpg.de/departments/

databases-and-information-systems/research/yago-naga/commonsense/howtokb/

5) Stanford OpenIE: To extract information like ontologies, affordances, casue-effects. Link: https://github.com/philipperemy/Stanford-OpenIE-Python
6) Household tasks dataset [12]: STAIR Actions is a video dataset consisting of 100 everyday human action categories. Each category contains around 900 to 1800 trimmed video clips. Each clip lasts 5 to 6 seconds. Clips are taken from YouTube video or made by crowd-source workers. Link: https://actions.stair.center

V. Simulation Environment

The simulation environments we can use:

1) Virtual Home [13]: VirtualHome is a platform to simulate complex household activities via programs. Key aspect of VirtualHome is that it allows complex interactions with the environment, such as picking up objects, switching on/off appliances, opening appliances, etc. Our simulator can easily be called with a Python API: write the activity as a simple sequence of instructions which then get rendered in VirtualHome. You can choose between different agents and environments, as well as modify environments on the fly. You can also stream different ground-truth such as time-stamped actions, instance/semantic segmentation, and optical flow and depth. Check out more details of the environmnent and platform in www.virtual-home.org.
2) Husky UR5 robot with ROS+Gazebo+RViz or PyBullet: See more details at https://www.clearpathrobotics.com/assets/guides/husky/HuskyManip.html and http://wiki.ros.org/husky_ur5_moveit_config/Tutorials/Husky%20UR5%20Mobile%20Manipulation%20Demo and https://pybullet.org/wordpress/
3) PyRobot is a framework and ecosystem that enables AI researchers and students to get up and running with a robot in just a few hours, without specialized knowledge of the hardware or of details such as device drivers, control, and planning https://ai.facebook.com/blog/open-sourcing-pyrobot-to-accelerate-ai-robotics-research/. Environment and simulation platform can be based on HabitatAI https://aihabitat.org/

VI. Evaluation Metrics

Most works like [10] only see task completion (binary) for a large number of tasks. Other metrics like Information Effort (IE) and Neglect Tolerance (NT) are defined in [14].

Metrics used in [15]

- Exact Match (EM). As in [16], EM is 1 if a predicted plan matches exactly the ground truth; otherwise it is 0.
- F1 score (F1). The harmonic average of the precision and recall over all the test set [17].
- Edit Distance (ED). The minimum number of insertions, deletions or swap operations required to transform a predicted sequence of behaviors into the ground truth sequence [18]. Instruction Edit Distance (IED) and Execution Edit Distance (EED) specified in [19]
- Goal Match (GM). GM is 1 if a predicted plan reaches the ground truth destination (even if the full sequence of behaviors does not match exactly the ground truth). Otherwise, GM is 0.

References

[1] H. Chen, H. Tan, A. Kuntz, M. Bansal, and R. Alterovitz, “Enabling robots to understand incomplete natural language instructions using commonsense reasoning,” *arXiv preprint arXiv:1904.12907*, 2019.

[2] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, “Deep semantic role labeling: What works and whats next,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 473–483.

[3] L. Zhou, C. Xu, and J. J. Corso, “Towards automatic learning of procedures from web instructional videos,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] S. Chitta, I. Sucan, and S. Cousins, “Moveit![ros topics],” *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[5] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz, “Web-enabled robots-robots that use the web as an information resource,” *IEEE Robotics and Automation Magazine*, vol. 18, no. 2, p. 58, 2011.

[6] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula, “Robobrain: Large-scale knowledge engine for robots,” *arXiv preprint arXiv:1412.0691*, 2014.

[7] D. Nyga, S. Roy, R. Paul, D. Park, M. Pomarlan, M. Beetz, and N. Roy, “Grounding robot plans from natural language instructions with incomplete world knowledge,” in *Conference on Robot Learning*, 2018, pp. 714–723.

[8] M. Forbes and Y. Choi, “Verb physics: Relative physical knowledge of actions and objects,” *CoRR*, vol. abs/1706.03799, 2017. [Online]. Available: http://arxiv.org/abs/1706.03799

[9] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, “Understanding natural language commands for robotic navigation and mobile manipulation,” in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[10] A. Xie, F. Ebert, S. Levine, and C. Finn, “Improvisation through physical understanding: Using novel objects as tools with visual foresight,” *arXiv preprint arXiv:1904.05538*, 2019.

[11] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, “Roboturk: A crowdsourcing platform for robotic skill learning through imitation,” *arXiv preprint arXiv:1811.02790*, 2018.

[12] Y. Yoshikawa, J. Lin, and A. Takeuchi, “Stair actions: A video dataset of everyday home actions,” *arXiv preprint arXiv:1804.04326*, 2018.

[13] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, “Virtualhome: Simulating household activities via programs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8494–8502.

[14] D. R. Olsen and M. A. Goodrich, “Metrics for evaluating human-robot interactions,” in *Proceedings of PERMIS*, vol. 2003, 2003, p. 4.

[15] X. Zang, A. Pokle, M. Vázquez, K. Chen, J. C. Niebles, A. Soto, and S. Savarese, “Translating navigation instructions in natural language to a high-level plan for behavioral robot navigation,” *arXiv preprint arXiv:1810.00663*, 2018.

[16] N. Shimizu and A. Haas, “Learning to follow navigational route instructions,” in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[17] B. M. Sundheim, “Tipster/muc-5: information extraction system evaluation,” in *Proceedings of the 5th conference on Message understanding*. Association for Computational Linguistics, 1993, pp. 27–44.

[18] G. Navarro, “A guided tour to approximate string matching,” *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[19] D. K. Misra, J. Sung, K. Lee, and A. Saxena, “Tell me dave: Context-sensitive grounding of natural language to manipulation instructions,” *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 281–300, 2016.