Natural Language to Long-Range Robot Navigation in Outdoor Environments

Matthew Berg

October 2020

1 Introduction

Tremendous progress has been made in the field of natural language processing for robots. Advanced language models have been integrated with multiple types of robots, paving a path towards seamless human-robot interaction. Still, there remains significant work to accomplish. Consider a pilot instructing an autonomous drone to “travel from Boston to Providence and go through towns with charging stops.” To fulfill the pilot’s request, the robot must resolve “Boston,” “Providence,” and “towns with charging stops” to real-world coordinates and identify a sequence of locations that satisfies the pilot’s non-sequential instruction. These tasks are computationally expensive in small domains, let alone a miles-wide geographic region. Further, the robot should not assume advance training on the Boston and Providence regions. Such training is expensive and difficult to scale. There is need for a computationally modest, domain-generalized system that enables the robot to fulfill natural language instructions in large outdoor environments.

Fulfilling users’ instructions is a language grounding and path planning problem. On the language side, the robot must handle unconstrained diction and non-sequential structure. There are numerous ways to describe the same navigational goal(s), which presents a challenge for the robot: distill a user’s intent from highly variable input. On the planning side, the robot needs to efficiently search a map that contains multiple levels of abstraction (eg., states, counties, and towns). The hierarchical map structure presents the robot with an additional challenge of fluidly planning over multiple levels of abstraction to fulfill the user’s instruction in a computationally efficient way.

This proposal outlines a natural language instruction-to-path plan system for the robot. It draws on previous joint work [1, 2], previous work from the Humans to Robots Laboratory [3], and introduces new language processing and map construction components. I anticipate the system offering a concrete step towards more efficient and seamless interaction with robots operating in large outdoor environments.

2 Approach

The proposed system should enable the robot to follow language instructions for long-range outdoor tasks. The input is a language instruction and the output is a path plan. By leveraging a loosely expected language structure
and the hierarchical map, the robot can reduce computational burdens of resolving language instructions in large outdoor environments. The proposed system is shown in Figure 1.

First, language instructions are grounded to a structured form. Landmark and hierarchy references are extracted from the instruction using constituency parsing [4, 5] and named entity recognition [6, 5]. Second, the parsed instructions are supplied to a Seq2Seq model [7] that grounds language to Linear Temporal Logic. Third, a sequence of named locations is calculated from LTL using the Abstract-Product MDP [3]. Fourth, a multi-resolution map representation is generated for the robot. Fifth, a portion of the named locations are resolved to real-world landmarks by assessing landmark references against semantic mapping data. Finally, the robot starts to plan and navigate in the outdoor environment, updating its map and calculating new landmark resolution (as needed) during travel. These steps provide the layers of abstraction between a user’s language instruction and the robot’s low-level autonomous systems.

### 2.1 Landmarks and Hierarchy

Landmarks specify sub-goals along a path of movement [8]. They are important for self-orientation and navigation in an environment [9]. The robot needs to understand landmark references to ground natural language to a structured form. At travel-time, the robot needs to resolve these references over a territorial hierarchy. Since users may specify the hierarchy level to initialize search (for example, “cities containing landmark A”), the robot should understand references to the territorial hierarchy as well. I propose approaching landmark and hierarchy reference extraction with constituency parsing [4, 5] and named entity extraction [6, 5].

I have observed that landmark references can be captured in noun phrases. Constituency parsing [4] enables the robot to extract noun phrases from the language instruction. Noun phrase constituents can be matched to a hierarchical expression (if applicable) using a nearest-parent algorithm. In effect, an instruction such as “go to cities with charging stops” can be parsed into (“cities”, “charging stops”). Leaf noun phrase constituents that are unrelated to a hierarchical expression can be directly added to the output. However,
there are cases when leaf noun phrases do not refer to landmarks. For example, in the instruction “make your way to Providence then go to Boston”, “your way” is a leaf noun phrase. The robot can remove non-landmark noun phrases with named entity extraction [6]. Noun phrases that do not match to a hierarchical expression are assessed against named entities using cosine similarity; phrases that do not meet the similarity threshold are pruned from the output.

This approach has two benefits. First, it converts the language instruction to a more uniform format before the robot attempts to find an LTL grounding. Second, identifying hierarchical references can reduce the search space of the robot’s map, potentially enabling faster planning. Preliminary tests of landmark and hierarchy extraction have achieved an F1-score of up to 61.76% for instructions containing hierarchical references and 84.88% for instructions that do not contain hierarchical references. However, testing was conducted with a limited number of language structures and landmarks. A larger corpus of language data is necessary for more accurate evaluation. In addition, future work could explore methods of extracting semantic landmark referring expressions for landmarks without a hierarchical reference. Currently, such expressions can be pruned by named entity extraction.

2.2 Grounding Language to Linear Temporal Logic

Language instructions can be non-Markovian. For example, in the instruction “go to Providence but first travel to a city with a charging station,” the robot must know it has traveled to a city with a charging station before it can then go to Providence. Non-Markov structure presents a challenge for the robot’s planning systems, which requires a linear sequence of independent states. The robot needs a structured form to represent language instructions.

Linear Temporal Logic is a structured formalism that encodes goals and constraints of the language instruction. For example, “go to C then D” is $F(D \land F(C))$ in LTL. There is a body of work on grounding language to LTL [1, 3, 10, 11]; I propose using the Seq2Seq [7] approach similar to [1]. However, instead of training the Seq2Seq model on fully-formed language instructions, I propose training on instructions pre-processed by the landmark and hierarchy extractor. Continuing the earlier example, $X := \text{go to A but first travel to a \\_c with a B}$, $Y := F(A \land F(B))$. The mappings $A \rightarrow \text{“Providence”}, \_c \rightarrow \text{“city”},$ and (“city”, “charging station”) are applied to the Seq2Seq model’s output.

This approach has two benefits. First, it retains hierarchical information without modifying the structure of LTL. Second, the landmark extraction problem is separated from the grounding problem, translating to potentially less parameters for the Seq2Seq model to learn. Similar to section 2.1, I intend to evaluate performance with a larger corpus of language instructions. I expect the grounding accuracy to be greater than the performance of joint previous work [1], which achieved 74.5% for one-unseen-landmark instructions and 53.49% for two-unseen-landmark instructions. In this work, three-landmark instructions are tested as well.

2.3 Abstract-Product MDP and Landmark Resolution

LTL encodes goals and constraints of the user’s language instruction. Equipped with an LTL formula, the robot needs to calculate a sequence of states that satisfies it. This sequence, more formally called a trajectory, guides
I propose using an AP-MDP [3] to calculate a trajectory from LTL. The AP-MDP is designed for LTL expressions in hierarchical environments. However, AP-MDP’s initialize a complete plan for the robot. In large outdoor maps, initializing a complete path plan can be prohibitively expensive. I propose decoupling trajectory planning and path planning, so that at initialization, an AP-MDP can be used solely to calculate the robot’s trajectory. For example, (“Boston’’ → (“cities”, “charging stops”) → “Providence”). I anticipate the key benefit of decoupling trajectory planning from path planning to be increased computational efficiency, since path planning can occur in a simpler two-dimensional space (see Section 2.4) and landmark resolution can occur over a longer period of time.

Language expressions in the trajectory (eg. Boston) need to be resolved to real-world coordinates. The robot can accomplish this resolution by assessing cosine similarity between the referring expression and semantic mapping data, as in the landmark resolution approach introduced in [1]. I propose resolving only start and end referring expressions to landmarks at trajectory initialization, to decrease the computational expense between trajectory initialization and the robot starting navigation. Intermediate landmarks would be resolved as the robot navigates. A roughly expected landmark resolution performance comparison is shown in Figure 2.

### 2.4 Metric-Topological Maps and Planning

A map of a large outdoor environment can be intractably large for the robot. To address this problem, I propose using a hierarchical metric-topological map. The metric map contains higher-resolution, nearby environment features at a single level of abstraction. The topological map contains multiple levels of lower-resolution features.
The metric and topological map are in the global coordinate frame and their landmark data is downloaded from Open Street Map [12]. The map representation is built with graph-tool [13].

The proposed map representation builds on joint previous work [2] by introducing a hierarchical topological structure. An administrative hierarchy (e.g., states, cities, towns) and bounding box center points are downloaded from Open Street Map. The center point is used to represent large geographic areas as a single node in the topological map. As the robot approaches a new area, topological resolution is increased (for example, when approaching a state, city nodes are added to the map). To plan over the map, the robot uses A-star [14]. Since the map representation is in the global frame and coordinate of at least one goal landmark is known, the heuristic function is Haversine distance between the robot’s current location and the next known landmark to visit.

The hierarchical approach allows more distant landmarks to be stored in a compact map representation. In effect, I anticipate better performance for longer-range tasks. My plan is to quantify space efficiency and planning efficiency of the hierarchical map representation with a corpus of language commands. Space efficiency is measured as the average number of nodes stored in the map representation when compared to baseline representations (fully metric, non-hierarchical metric-topological). Planning efficiency is measured as the planning time for a trajectory using the map representation when compared to baseline representations.

3 Timeline

1. October: Collect a larger corpus of language instructions. Language pipeline trained, tested, and debugged on this larger corpus.
3. December/January: Integrate maps into end-to-end system. I anticipate this step reveals improvements that may need to be made across the system; start on those improvements. Writing.
5. March: Writing.

References


