Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Grounding Natural Language References to Unvisited and Hypothetical Locations

Tom Williams[1] and **Rehj Cantrell**[2] and **Gordon Briggs**[1]

Paul Schermerhorn[2] and **Matthias Scheutz**[1]

Human-Robot Interaction Laboratory
Tufts University[1] and Indiana University[2], USA
{williams, gbriggs, mscheutz}@cs.tufts.edu, {rcantrel,pscherme}@indiana.edu

Abstract

While much research exists on resolving spatial natural language references to known locations, little work deals with handling references to unknown locations. In this paper we introduce and evaluate algorithms integrated into a cognitive architecture which allow an agent to learn about its environment while resolving references to both known and unknown locations. We also describe how multiple components in the architecture jointly facilitate these capabilities.

Introduction

A recent area of research is location-based spatial reference resolution, i.e., representing and resolving references to locations in the environment such as rooms, hallways, doors and others. This is particularly important for many human-robot interaction scenarios where humans instruct robots in natural language to perform various tasks in the environment (e.g., for robotic wheelchairs to transport their users to the intended locations).

Most approaches to spatial reference resolution have focused on resolving references to *known locations* (e.g., Hemachandra et al. 2011; Kollar et al. 2010; Zender, Kruijff, and Kruijff-Korbayová 2009; Shimizu and Haas 2009; Chen and Mooney 2001; Matuszek, Fox, and Koscher 2010). Only Matuszek et al. (2012) also resolve references to *unknown locations*, parsing natural language utterances directly into action sequences which will bring the robot to the described location. However, this means that the robot will only be able to obtain information about other places relative to the locations where it received the description, rather than being able to store the description and utilize it from any location.

All previous approaches to spatial reference resolution also use some form of map to represent the robot's environment. This map can either be provided beforehand or built on the fly through exploration or a guided tour of the environment. All such maps, however, have been "static" in nature, i.e., they will not (and are not allowed to) change from the point when reference resolution is performed. This means, for example, that a robot will not be able to learn new facts about its environment while navigating to a previously referenced location (since the destination location is typically not added to the robot's map, it will also remain unknown even after having been visited).

Moreover, previous approaches for spatial reference resolution only handle natural language commands, with the exception of Zender, Kruijff, and Kruijff-Korbayová (2009) where noun phrases are resolved out of the context of declarative or interrogative utterances.

In this paper, we introduce algorithms for spatial reference resolution integrated into a cognitive robotic architecture that significantly improve previous proposals by: (1) systematically adding unknown places to the map, which allows robots to meaningfully communicate about unknown places without having to first discover their exact location (e.g., by way of navigating there); (2) updating the map as the agent discovers unknown environments, which allows robots to have natural language interactions about new environmental features discovered as part of navigating to a previously unknown place; and (3) generating action sequences only when they are actually needed to visit the referenced location (instead of going there, we proposed to store the information in a location-independent form, which affords the robot the capability of learning a map entirely through dialogue).

Algorithms, Architecture and Implementation

To be able to use algorithms for spatial reference resolution to unknown locations such that an artificial agent can converse about and perform actions involving them, we employ the Distributed Integrated Affect, Recognition and Cognition (DIARC) architecture (Scheutz et al. 2007). Figure 1 depicts the relevant DIARC components used to process natural language utterances (hereafter denoted by small caps such as NLP for "Natural Language Processing"), with a focus on the SPatial EXpert (SPEX), which contains all proposed algorithms. SPEX receives information about landmarks from perceptual components in order to build a map of its environment. NLP queries SPEX in order to perform reference resolution on utterances it receives from the speech recognition component. NLP then sends utterance semantics to the dialogue manager (DIALOGUE). DIALOGUE uses contextual information from the belief modeling component (BELIEF) to perform pragmatic analysis on received

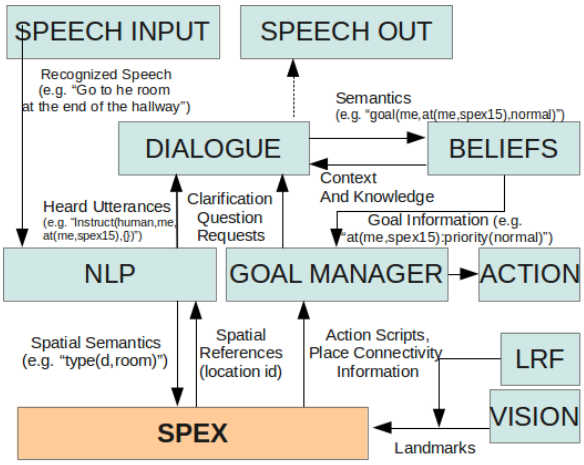

Figure 1: Partial architecture diagram isolating the components and interactions relevant to natural language spatial cognition.

semantics. BELIEF uses these semantics to inform the Goal Manager (GM) of new goals. The GM uses connectivity information from SPEX for path planning, and uses action scripts generated by SPEX when a destination has only been described to the robot and not yet visited (and thus might not have a precisely known location). It is necessary for the GM to rely on SPEX in these circumstances, as the best way to formulate a plan to these locations is by exploiting knowledge gained through dialogue interactions.

Spatial Semantics

Producing spatial descriptions entails identifying relationships between objects and modifiers, whether that is expressed in predicate semantics (as we do it), or in some other form (e.g., spatial description clauses in Tellex et al. 2011). To identify these relationships, we use a data-driven dependency parser which produces semantic predicates (for details, see Cantrell et al. 2010).

Components such as SPEX interpret these semantic predicates in ways that are typically more limited than humans would interpret them. For example, a relational term like $to\_right(x, y)$ is interpreted by SPEX in the context of a route description in the following way: when the robot is passed a landmark y (e.g., a room or hallway), x is always to the right from the robot's perspective (whereas a human might more flexibly entertain additional interpretations that are not egocentric). To illustrate, imagine that the robot is located as shown in Figure 2. C refers to the robot's current location, specifically the room in which it is located (as opposed to its location within that room). The robot is told "The room at the end of the hall is to the right", or $to\_right(r, C)$ where r is the name of the room at the end of the hall. The robot will assume that, as it is exiting C, r is to its right. So far, that is in accord with human intuition, and this description can apply to the labeled room on the map. However, imagine it is instead standing in the hallway and is given the same utterance. Now it thinks that r is to its right *as it is exiting the hallway*. This is no longer true of the labeled room, and does not match the human interpretation of the utterance.

With this in mind, we examine the utterance "go to the room at the end of the hall down to the right." *The room* is the only noun to which *at the end of the hall* could be attached; there is no ambiguity. By contrast, it is not clear whether *the hall* or *the room* is the noun to which *down to the right* should be attached. The parser chooses the most syntactically and lexically likely relationship. However, based on SPEX's interpretation of to_right, only the attachment to *room* will be successful. If it is attached to *hall*, the robot will assume that, as it is exiting C into the hallway, it will still need to find a second hallway that is to its right and which is connected to the room in question. Thus, the only acceptable semantic representation for the phrase "the room at the end of the hall down to the right" is $\iota x(endroom(x) \wedge hall(h) \wedge connected\_to(x, h) \wedge to\_right(x, C))$. If the attachment were to the hallway, the semantics would instead be $\iota x(endroom(x) \wedge hall(h) \wedge connected\_to(x, h) \wedge to\_right(h, C))$. Note that SPEX and NLP together resolve such ambiguities in fixed manner in that "higher attachments" are always preferred in the parser. In other words, because *at the end of the hall* is already attached to (is dependent on) *the room*, attachment to *the room* is preferred over attachment to *the hall*.[1]

Spatial Reference Resolution and Exploratory Route Suggestion

SPEX aggregates information about its environment in order to build a hierarchical map M stratified into two layers, similar to the mapping approaches presented in Kruijff et al. (2007) and Kuipers (2000). The *top layer* $M_{top} = (V_{top}, E_{top})$ is a graph with vertices V_{top} and edges E_{top} where each vertex $v \in V_{top}$ is a *large-scale* place such as a room or hallway, and where each edge $e \in E_{top}$ represents a means to travel between such places (i.e., through a doorway). The *bottom layer* $M_{bot} = (V_{bot}, E_{bot})$ is a graph with vertices V_{bot} and edges E_{bot} where each vertex $v \in V_{bot}$ is a *small-scale* place: a specific location in a room or hallway, or a landmark such as a door, and where each edge $e \in E_{bot}$ represents a path between these places. A vertex v in either graph is indexed by a uniquely referring identifier, and contains an adjacency list of connecting places' identifiers and a list of properties held by the represented place. The primary difference between the two levels is that M_{top} is only concerned with whether or not its vertices connect (e.g., whether or not a room is accessible from a given hallway), while M_{bot} is additionally concerned with the details of where and how its vertices connect. For this reason, the metric positions of M_{bot}'s vertices are stored when

[1] Eliminating this fixed assumption about attachment precedence is an important next step, because the robust formulation of spatial semantics is critical to the success of the whole system. Specifically, we are currently working on integrating monitoring mechanisms that will enable the correction of false semantic interpretations due to syntactic ambiguity when an interpretation can be invalidated through perceptions in the given environment.

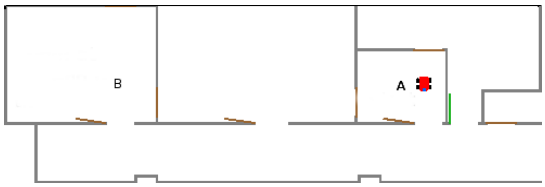

Figure 2: Simulation Environment. A: Robot's initial position, B: The room at the end of the hallway.

they can be determined, while such information is not maintained for vertices in M_{top}. Topological information, such as an ordering of places within a hallway, can be extracted from M_{bot} using the coordinates of V_{bot}. Each large-scale place in V_{top} also stores a list of places from V_{bot} that it contains. This stratification is conceptually motivated by the fact that natural language references to spatial locations are typically concerned with *large-scale* places, whereas sensing and planning systems are typically concerned with the *small-scale* places contained within.

Information used to augment this map can come from both perceptual components and dialogue. As the robot travels through its environment, SPEX actively requests information from various perceptual components, e.g, it requests information about landmarks from the Laser Range Finder (LRF) component. If LRF has detected a landmark, it returns its coordinates to SPEX, as well as coordinates necessary for establishing the landmark's orientation or for navigating through traversable landmarks such as doors. SPEX then uses the algorithm summarized in $detectLandmark$ (Algorithm 1) to process this information.

$detectLandmark$ seeks to determine whether a landmark has been seen before, and if not, to build a new representation for the landmark and any new locations its observation entails (e.g., rooms on the other sides of observed doors). To do so, it requires the aforementioned coordinates, and a list of predicates P describing any other properties the landmark might have. For example, if a camera is being used, visual characteristics of a landmark might be included. If SPEX does not know of a landmark in the given position or of a previously described but as-of-yet unobserved landmark which matches the given description, it adds the landmark and adjacent small-scale places to the map, as well as any large scale places entailed, such as a room assumed to be on the other side of an observed door.

SPEX determines whether this adjoining space is a room or hallway using the simplifying heuristic that rooms only connect to hallways and not to other rooms. A more general solution would be to postpone this decision until this property can be verified through exploration. This would require more sophisticated exploration strategies, which we will later discuss.

When NLP sends SPEX information regarding a received utterance, it is as a list of predicates P representing the semantics of the utterance that convey some information about the structure of the environment. This can be seen in $processSemantics$ (Algorithm 2). SPEX separates these predicates into three categories: *type predicates* (predicates that express the type of an entity, such as $Room(x)$),

Algorithm 1 detectLandmark(D, T, O, P)

```
1: (D,T,O are Coordinates, P is a list of Predicates)
2: for all d ∈ ν (with set of nearby doors ν ) do
3:    if dist(position(d), D) < Θ /*Θ some threshold*/ then
4:       update the position of d
5:       return
6:    end if
7: end for
8: m is a new place list
9: for all φ ∈ M_bot such that loc(φ) == ∅ do
10:   if ∀r ∈ P( r(P) ) then
11:      m ← m ∪ {φ}
12:   end if
13: end for
14: if m == ∅ then
15:   add places φ_D, φ_T, φ_O at coordinates D, T, O to M_bot
16:   for all p ∈ P do
17:      properties(φ_D) ← properties(φ_D) ∪ {p}
18:   end for
19: else
20:   c ← the room the robot is currently in
21:   t ← the place connected to n in c
22:   o ← the other place connected to n
23:   loc(m[0]) ← D; loc(t) ← T; loc(o) ← O
24:   children(c) ← children(c) ∪ {t}
25:   connect(c, parent(o))
26: end if
```

Algorithm 2 processSemantics(P)

```
27: (P is a list of Predicates)
28: i ← Map(String ↦ Identifier[ ])
29: for all p ∈ P do
30:   if p is a type predicate then
31:      i(p → type) ← i(p → arg_0) ∪ {new list n}
32:   end if
33: end for
34: for all typepredicatep ∈ P do
35:   buildBindings(p, i)
36: end for
37: if ∃ l ∈ i such that size(l) == 0 then
38:   return createPlaces(P, i)
39: else if ∃ l ∈ i such that size(l) > 1 then
40:   return ambiguous
41: end if
42: return i(P[0] → arg_0)[0]
```

descriptive predicates (predicates that describe an entity, such as $Color(x, Green)$), and *relational predicates* (predicates that describe relations between entities, such as $Connects(x, y)$). This separation is useful since each type of predicate plays a different role in the process of reference resolution. When SPEX receives a predicate list from NLP, it attempts to determine the identities of any locations referenced in the utterance (i.e., specified in a type predicate with name "Room", "Door" or "Hall"). For each landmark or large-scale location referenced by a type predicate in P, SPEX constructs a list of candidate identifiers n (Algorithm 2, 31). For example, the type predicate $Room(r)$ will result in the creation of a list which initially contains the identifiers of all known rooms (while this is acceptable when the

Algorithm 3 getScript(S, D)

(S is the id of the source, D is the id of the destination)
A is a new action script
if S, D are small scale places with the same parent **then**
  if $locationKnown(D)$ **then**
    add instruction to A to move to $position(D)$
  else
    $useCluesToPlanMotion(S, D, A)$
  end if
else if S, D are small scale places with connected parents **then**
  if $locationKnown(D)$ **then**
    add instructions to A to approach and move through the door which leads to D
  else
    $useCluesToPlanMotion(S, D, A)$
  end if
else if $\neg locationKnown(D)$ **then**
  I_0, I_1 are new lists of identifiers
  $I_1 \leftarrow reverse$(route from D to the closest point to it whose location *is* known)
  $D \leftarrow first(I_1)$
  $I_0 \leftarrow planRoute(S, D)$
  for all $i \in concatenate(I_0, I_1)$ **do**
    add instruction to A to move to i
  end for
end if
return A

number of possible locations is small, it quickly becomes intractable as the number of locations grows, and we are therefore considering various strategies to improve efficiency).

SPEX then uses the descriptive and relational predicates to eliminate bad candidate locations from these lists. Once SPEX has reduced each list to its smallest size, it examines each list (Algorithm 2, 37). The size of the lists can be used to classify the level of ambiguity in the utterance. If a list is empty, then the description corresponds to a previously unknown place. In practice, this may be incorrect since the description could actually refer to a known location in a way that is not currently determinable (e.g., if the robot knows of a certain door, but does not know that the room beyond it is the cafeteria, then it will not be able to automatically resolve a reference to "the cafeteria"), but this discrepancy will need to be resolved during exploration or during further dialogue. If the received semantics came in the context of an assertion about the world as opposed to a query regarding its structure, SPEX adds a new entry to the appropriate map level, and then adds to this new entry any relevant properties from P. In the case of the room at the end of the hall, the list corresponding with the referenced room is empty, so SPEX creates a new large-scale place to represent the room, along with small-scale places in M_{bot} representing the door which must connect it to the hallway and the places on either side of this door. SPEX then gives the newly created door the property ($end\_of\_hall(d, h)$), indicating where in the hall it is located. This is necessary since SPEX would otherwise not be able to identify the door as being at the end of the hall, as its coordinates are as of yet unknown.

If a list contains a single identifier, such as in the case of the list associated with the hallway, SPEX assumes this is the identifier of the referenced location and modifies that place's connections and properties accordingly.

This brings up one of the difficult problems which SPEX must deal with. If the robot is given a description of a place whose location it does not know, SPEX needs to create a representation of that place without specifying its metric location. If it is informed of some series of connected rooms and hallways, their topological representations should be linked to the known map only if their locations are known relative to some known place. If the robot is later able to determine the precise location of one of the rooms, its child locations in M_{bot} can then be given metric positions. In the case of the room at the end of the hall, SPEX creates place representations in M_{bot} for the door and the points of access on either side of it, but sets a property in each of these representations indicating that its coordinates are unknown. The identifiers for these places are then placed into a list of unknown places which is considered whenever a new place is seen.

If there are multiple candidate identifiers for a described place, SPEX informs NLP that more information is needed to disambiguate between the candidates (40) – determining whether and how to choose between multiple candidates is an interesting area for future research. SPEX could, for example, return a distribution representing the relative likelihoods of the various candidates, and allow NLP to decide for itself how to resolve this ambiguity, or it could create and assert the conveyed information for all candidate locations, along with some diminished confidence value. This would also be useful if NLP needed to partially assert two alternate semantic interpretations of a received utterance.

Another important capability that we address in SPEX is the generation of actions to reach locations whose metric locations are unknown. Since the semantics in the case of "the room at the end of the hall down to the right" involve clues about the location of the place (i.e., it is at the end of a hallway, in a room "to the right" of the current one), SPEX is able to produce a possible action sequence to reach the target location. This process is summarized in Algorithm 3. When SPEX is asked for an action script which, when executed, will take the robot between two locations S and D, it creates a new action script A, and adds actions to this script based on the properties and connectivity of S and D. If S and D are in the same room and D's coordinates are known, the script returned by SPEX only has to contain an instruction to travel to D. If S and D are on opposite sides of a door, SPEX adds to A additional instructions to approach and move through the separating door. If D's location is unknown, SPEX adds appropriate instructions to A using what little it *does* know about D. For example, if SPEX knows that D is to the left down the hall, from S, it will add instructions to turn left and traverse the hallway. If none of these cases hold, then S and D must be separated from each other by one or more rooms. If this is the case, then SPEX finds the shortest known path from S to c (the closest point to D whose location is known), and adds to A an instruction to navigate to each place along this path.If D's location is unknown, then similar instructions are added to A for each location along the shortest path from D to c. The distinction between these two parts of the overall route is made because they use different heuristics due to the difference in available information between them. That is, while planning the first part of the route, SPEX can use the metric positions of

locations to find the physically shortest route. While planning the second part, however, the only heuristic it is able to use while planning at this stage is the number of places the robot will need to travel through. Finally, the action script is returned. For the example sentence of the room at the end of the hall, the action sequence is formalized in this manner:

```
[moveTo, self, exitposition]
[exitRoom, self]
[moveTo, self, entryposition]
[moveTo self, currentroom]
[turnRel, self, ang]
[traverse, self]
[informSpexEnd, AtEndOfHall]
[moveTo, self, destination]
```

Note that since SPEX is unable to directly detect "end-of-hall-ness", the created script includes a request (`informSpexEnd`) to be informed when the exploration of the hallway is completed. SPEX finally alerts DIARC's Goal Manager (GM) component of any new places it learned of and the connections between them.

When the GM receives the above script and and issues execution, the robot exits the room, turns in the direction indicated, and starts driving down the hall. When the robot reaches the end of the hallway, the GM informs SPEX that the exploration has finished. SPEX checks whether any nearby location is close enough to be construed as being "at the end of the hall." It then examines all places connected to the current hallway, and checks to see if any of them have "end-of-hall-ness" listed among their properties (in this case, the previously described room does). Assuming one place fits this description (the system does not currently attempt to solve the case of multiple places being good candidates), SPEX consolidates its representations of the recently encountered place and the described place, placing into a consolidation map the identifier of the place that is consolidated away, in case the old reference is used by some other component.

Evaluation

We ran three sets of evaluations of SPEX. In the first two, SPEX alone was evaluated, and in the third, the integrated architecture was tested. In order to abstract away from problems with other parts of the architecture such as NLP, SPEX was provided with a starting location and gold standard semantics for the utterance being tested which uniquely identified a location; the robot was not asked, for example, to go to "the room at the end of the hall" in an environment in which several rooms existed at the hallway's end. If these types of requests and environments had been included in testing, performance would have decreased.

In the first evaluation, SPEX was given a full map of an environment, and 64 resolution tests, which represented all ways that a set of utterances (such as "the room to your immediate left when exiting the break room", "the room at the right end of the hallway" and "the third room on the right facing left from your current position") could be successfully resolved in the environment. For example, "the room to your immediate left" was evaluated from all starting points that had a room on their immediate left. SPEX generated the correct reference for 64/64 (100%) of the tests.

In the second evaluation, SPEX was given a partial map of the same environment; 44% of the large-scale locations were removed, along with all contained small-scale locations and any connecting doors. SPEX was then given all 34 tests from the original set of tests whose starting location was still known. Since some destinations were unknown in this set, success in the case of an unknown destination was qualified as generating a new place representation and returning a plan which would successfully take the robot from its current location to the location. SPEX passed 34/34 (100%) of these tests.

Finally, the complete architecture using SPEX was tested in a simulated environment on a set of utterances. We used a simulated MobileRobots Pioneer robot, although the remainder of the architecture ran in the same configuration that it would on the real robot. We first gave the command "Go to the room at the end of the hallway down to the right" to the robot in the simulated environment pictured in Figure 2. The robot exited the room and proceeded to the right end of the hallway. Examining SPEX's map showed that SPEX had successfully consolidated its representations of the rooms the robot had heard referenced in natural language and observed at the end of the hall. Thus, the original reference was successfully resolved to its physical location.

We also evaluated some basic exploratory functionality for resolving ambiguous statements. Consider the command "Go to the room at the end of the hallway." In an unknown environment, this will result in the GM asking SPEX for an action script, which will need to be formed using the useCluesToPlanMotion (Algorithm 3, 49) function. When this function tries to determine which end of the hallway it needs to send the robot to, it will determine that the room could be at either end of the hallway. It thus chooses one of the ends and adds the necessary instructions to the action script. It then creates a new script to return to the choice point and travel to the other end of the hallway, and stores this second script in an "alternate plan" list indexed by the destination point. SPEX then returns the first action script. When the system follows this script and travels to the first end of the hallway, the last action it will execute will be to move to the destination point. If the reference is successfully resolved, it will move to that point. If it is not, the GM will once again ask SPEX for an action script. SPEX will check its alternate plan list and see that there is a plan waiting for that destination, and will remove and return it to the GM. Assuming the robot's interlocutor did not give an instruction to go to a nonexistent location, this plan will lead it to the target location. We tested this in the manner of the first two steps of evaluation and achieved successful results, as evidenced in the produced action scripts:

```
[moveTo, self, spex12]
[turnRel, self, -1.5708]
[traverse, self]
[informSpexEnd, AtEndOfHall]
[moveTo, self, spex14]

[moveTo, self, spex12]
[traverse, self]
[informSpexEnd, AtEndOfHall]
[moveTo, self, spex14]
```

Discussion

The above evaluations showed that SPEX is able to successfully resolve spatial references to both known and unknown locations as long as the spatial semantics picks out places uniquely. Storing the information gleaned from natural language and through exploration in a location-independent format affords the robot improved capabilities. Specifically, it allows the robot to (1) travel to previously described locations, (2) describe how two unknown locations are positioned relative to each other, (3) pause an action sequence and then later resume it from another location, and (4) return to a known location after visiting an unknown one. Finally, augmenting the robot's world model based only on descriptions allows a robot to learn a map purely through dialogue if it is able to extract sufficiently accurate semantics representations, while none of the approaches mentioned in the introduction would able to learn a map of their environment without physical exploration from dialogue alone.

Despite these improvements, SPEX has several shortcomings, specifically in situations where the attempt to resolve a spatial reference produces either no candidate places, or several appropriate candidates. Consider the instruction "Go to the cafeteria": if the robot knows of no cafeterias, what heuristics should it use to determine where to explore? Clearly, unless the robot has some notion of where cafeterias are usually located (e.g., in buildings like the current one), this will be very challenging. One strategy might be to simply ask a human for help. If that is not feasible or not allowed, another strategy might be for the robot to start exploring its environment, even when it has no notion of the goal location (some strategies for this approach have been suggested, e.g., Hawes et al. 2011). Sometimes a combination of strategies may be called for – identifying the best strategy for a given situation is in itself a challenging open research problem.

Another condition in which our current integrated systems is not able to resolve references in general is when the robot is able to identify several locations that are candidate referents. For example, if the robot is told to go to the cafeteria and it knows of several cafeterias, how is it to determine the intended one? This is a problem we have not yet addressed, except for the limited fashion examined in the final part of our evaluation, an approach that could be improved through many techniques we are interested in investigating in future work; prioritization of exploration based on relative likelihood, the use of other experts (e.g., an Episodic Memory Expert), the modeling of the beliefs and knowledge of other agents, and the ability to query an interlocutor for disambiguating information. Many of these techniques could in fact be used for both of the particularly difficult situations described above.

Note that we did not include either type of situation in our evaluation. For example, SPEX was not asked to deal with underspecified descriptions of locations, which typically happen in natural interactions. A place could easily be described in a way which fails to mention important details that are necessary for determining its location. In such a case, a representation for the described place would have beeen added to the map, but SPEX would either have beeen unable to generate a plan to reach it, or it would have never been able to recognize the place when it was encountered. There are additional complications that would impact the performance of SPEX, for example, the environmental complexity (including multiple intersecting hallways with loop closure, multi-level spatial layouts with connections among the levels, and others). Finally, our evaluation also assumed reliable perceptual information, but this is rarely the case in practice. For example, if the robot is sent to the third room in a hallway but fails to notice one of these doors, many problems will arise. In the second of our evaluations, we counted a test case as successful if SPEX was able to generate an appropriate action plan, but did not check whether the robot made mistakes while carrying out those plans as this would require additional action monitoring mechanisms to detect action failures and mechanisms to recover from them.

Conclusions

We presented SPEX, an architectural component consisting of several algorithms that are jointly capable of resolving references to unknown locations in an indoor environment through delayed exploration in such a manner that the unknown location can be discussed and reasoned about without having to visit it first. We discussed how SPEX's capabilities are greatly facilitated by its interaction with other components in an integrated cognitive robotic architecture (in our case the DIARC architecture). And we reported results from several evaluations of SPEX alone as well as the integrated system.

Future work will focus on handling referential ambiguities due to syntactic constructions as well as the spatial complexity of the environment. Moreover, we will investigate ways in which SPEX can be used in answering questions about hypothetical changes to the environment, such as "If the door between the cafeteria and kitchen were unlocked, what would be the fastest way to the atrium from here?"

Acknowledgments

This work was in part funded by NSF grant 111323 and in part funded by ONR grant #N00014-11-1-0493.

References

Cantrell, R.; Scheutz, M.; Schermerhorn, P.; and Wu, X. 2010. Robust spoken instruction understanding for HRI. In *Proceedings of the 2010 Human-Robot Interaction Conference*.

Chen, D. L., and Mooney, R. J. 2001. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.

Hawes, N.; Hanheide, M.; Hargreaves, J.; Page, B.; Zender, H.; and Jensfelt, P. 2011. Home alone: Autonomous extension and correction of spatial representations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 3907 –3914.

Hemachandra, S.; Kollar, T.; Roy, N.; and Teller, S. 2011. Following and interpreting narrated guided tours. In *Pro-*

ceedings of the IEEE International Conference on Robotics and Automation.

Kollar, T.; Tellex, S.; Roy, D.; and Roy, N. 2010. Toward understanding natural language directions. In *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction*, HRI '10, 259–266. New York, NY, USA: ACM.

Kruijff, G.-J. M.; Zender, H.; Jensfelt, P.; and Christensen, H. I. 2007. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems* 4:125–138.

Kuipers, B. 2000. The spatial semantic hierarchy. *Artificial Intelligence* 119:191–233.

Matuszek, C.; Herbst, E.; Zettlemoyer, L.; and Fox, D. 2012. Learning to parse natural language commands to a robot control system. In *Proc. of the 13th Intl Symposium on Experimental Robotics (ISER)*.

Matuszek, C.; Fox, D.; and Koscher, K. 2010. Following directions using statistical machine translation. In *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction*, HRI '10, 251–258. New York, NY, USA: ACM.

Moratz, R., and Tenbrink, T. 2006. Spatial reference in linguistic human-robot interaction. *Spatial Cognition and Computation* 63 – 106.

Moratz, R.; Tenbrink, T.; Bateman, J.; and Fischer, K. 2003. Spatial knowledge representation for human-robot interaction. In Freksa, C.; Brauer, W.; Habel, C.; and Wender, K. F., eds., *Spatial cognition III*. Berlin, Heidelberg: Springer-Verlag. 263–286.

Scheutz, M.; Schermerhorn, P.; Kramer, J.; and Anderson, D. 2007. First steps toward natural human-like HRI. *Autonomous Robots* 22(4):411–423.

Shimizu, N., and Haas, A. 2009. Learning to follow navigational route instructions. In *Proceedings of the 21st international jont conference on Artifical intelligence*, 1488–1493. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Stopp, E.; Gapp, K.-P.; Herzog, G.; Laengle, T.; and Lueth, T. C. 1994. Utilizing spatial relations for natural language access to an autonomous mobile robot.

Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M. R.; Berjee, A. G.; Teller, S.; and Roy, N. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI'11*.

Zender, H.; Kruijff, G.-J. M.; and Kruijff-Korbayová, I. 2009. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, 1604–1609. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.