Interpretation of Spatial Language in a Map Navigation Task

Michael Levit and Deb Roy

***Abstract*— We have developed components of an automated system that understands and follows navigational instructions. The system has prior knowledge of the geometry and landmarks of specific maps. This knowledge is exploited to infer complex paths through maps based on natural language descriptions. The approach is based on an analysis of verbal commands in terms of elementary semantic units that are composed to generate a probability distribution over possible spatial paths in a map. An integration mechanism based on dynamic programming guides this language-to-path translation process, insuring that resulting paths satisfy continuity and smoothness criteria. In the current implementation, parsing of text into semantic units is performed manually. Composition and interpretation of semantic units into spatial paths is performed automatically. In evaluations, we show that the system accurately predicts speakers' intended meanings for a range of instructions. This work provides building blocks for a complete system that, when combined with robust parsing technologies, could lead to a fully automatic spatial language interpretation system.**

***Index Terms*— navigational instructions, spatial language understanding, human-machine interaction, natural language processing**

I. Introduction

We present components of a system that converts verbal descriptions of paths produced by human instruction givers into sequence of actions that an automated agent must take in order to successfully follow paths anticipated by the instruction givers.

Many application areas including robotics, video games and geo-spatial communications analysis may benefit from automatic understanding of navigational language. In a video game scenario, for instance, players can be enabled to guide game characters throughout the virtual world of a game. This may be especially powerful when there are large numbers of computer controlled characters in which case direct control using keyboard and mouse can become cumbersome.

A number of related systems designed to operate in robotic and domestic environment have been described in the literature (e.g. [1], [2], [3], [4], [5]). In contrast to this previous work that involves sensor-derived (and thus noisy and incomplete) knowledge of the world, we consider the interpretation of relatively complex spatial language by assuming high level knowledge of the entire map and all landmarks is available to the system.

The scenario that we adopted for this work allows humans to use speech which is unconstrained from both linguistic and representation points of view. The MAP-TASK [6] corpus was selected for system development and evaluation. This corpus is a collection of transcribed human/human dialogs involving cooperative path planning using maps. To collect data, pairs of participants were given similar two-dimensional maps. One of the participants, the instruction giver, provided navigational instructions to the other participant, the instruction follower, that would guide the latter along a path drawn only on instruction giver's map. An example of a section of such a map with a reference path is depicted in Fig. 1. There were no restrictions whatsoever on language that could be used for navigation. An advantage of this non-invasive "eavesdropping" scenario is that subjects don't attune their navigation strategies to existing or presumed limitations of any automated understanding system (see [7]).

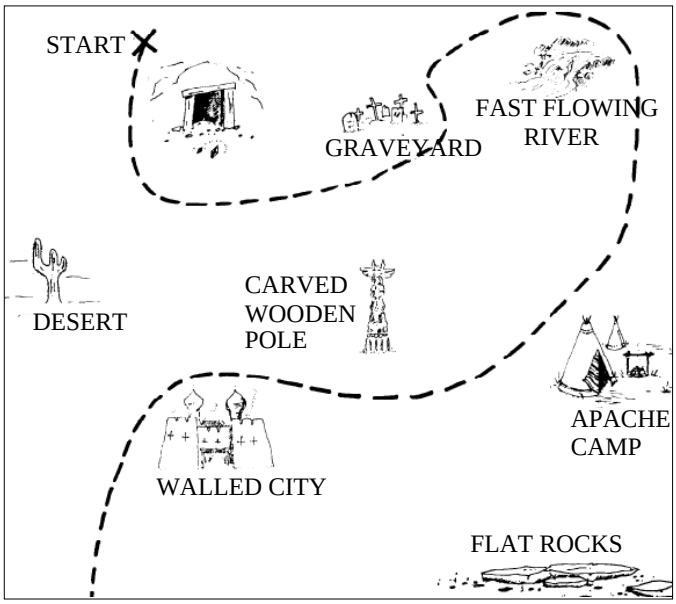

Fig. 1. Sample portion of a map from the Map-Task corpus. The path indicated by the broken line only appears on the map seen by the instruction giver. The instruction follower's goal is to recreate this path based on spoken dialog with the instruction giver.

Because of the very high complexity of spontaneous language that arose from the choice of MAP-TASK, we decided to focus on the understanding problem by initially ignoring syntactic parsing issues and turning our attention to different basic strategies people used to convey navigational information. Similar to [3], we manually extract basic instructions (which we named *Navigational Information Units*, or *NIUs*), however our units cover a much broader scope of possible instructions. Some examples of NIUs include moving around objects, moving in absolute directions (e.g., south, left), turning, and verifying closeness to a specific landmark.

One contribution of this paper is in showing that most of these NIUs can be decomposed in a number of "orthogonal" constituents (e.g. type of a move and its reference object), such that the meaning of each NIU or — following a functional approach to understanding — the realization of the path interval it describes, can be obtained as a Cartesian product of the meanings of all its constituents. Each of the NIUs can be represented as a parametrized rule with a certain degree of learned flexibility and parameter slots filled by these constituents.

A second contribution of this work is a novel algorithm that processes sequences of NIUs in order to produce coherent paths which are empirically shown to be similar to the reference paths instruction givers intended to communicate to instruction followers. This integration is possible by virtue of constraints implicated in the instructions (e.g. moving around an object pre-supposes that we must be in its vicinity even before the action can take place) and also by some common knowledge (e.g. car-objects can not be crossed, while bridges can).

II. Navigational Information Units

EXTRACTING basic instruction elements from sentences containing navigational information and grounding them in action primitives is a common strategy for understanding systems. The task environment and designer's preferences determine the choice of elements for a particular system, but generally the idea of splitting instructions in motions and referential descriptions [8] is widely accepted.

In [4], describing architecture of a system that understands verbal route instructions in a robotic environment, MacMahon uses four basic instructions: turning at a place, moving from one place to another, verifying view description against an observation and terminating current action. While perception undoubtedly plays a crucial role in human orientation and navigation abilities, in our route planning scenario geometries of all objects participating in a scene are known beforehand, and so we can reformulate the instruction categories above only in terms of this spatial knowledge, thus rendering their procedural aspect more homogeneous.

The feasibility of an automated system that translates from route descriptions to route depictions (and vice versa) is suggested by Tversky and Lee in [9]. After studying how humans describe and depict routes, the authors observe that both processes can be decomposed into equivalent sets of verbal and graphic elements respectively. The lexicon of elements used by the authors consisted of (selecting) *landmarks*, (changing) *orientations* and *actions* (such as moves), and was borrowed from [10].

In a discussion on the semantics of spatial expressions, Jackendoff [11, Chapter 9] provides linguistic evidence for a conceptual distinction between *places* and *paths*. While paths specify trajectories of a traveler, places describe his/its locations. The primary characteristic of a path is the change of location. Turns can be viewed as changes in orientation. These considerations led to four basic types of NIUs in our hierarchy:

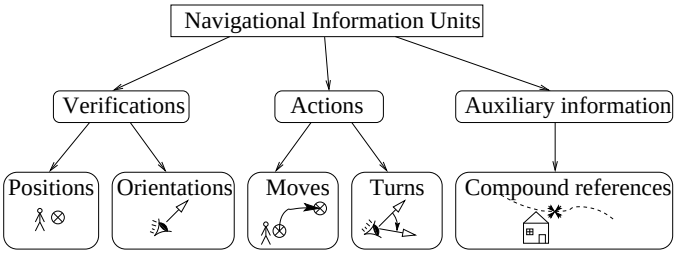

Fig. 2. Hierarchy of Navigational Informational Units (NIU's).

moves[1], *turns*, *positions* and *orientations*. The distinction is not always clear, since moving can result in change of orientation and turning in a practical setting can imply significant shift in position. We discovered however that even though different procedures are used to realize moves and turns, the overall path modeling performance doesn't suffer from the local ambiguity of such issues. Altogether, moves and turns can be subsumed under the general notion of *actions*, and positions and orientations can be viewed as *verifications*. Fig. 2 shows the full hierarchy of NIUs. The category *compound reference* of type *auxiliary information* is a special type of NIUs that we explain below.

Complex spatial instructions are decomposed into a set of NIUs. For example, *"Now could you go north past the house till you are eh right by the forest"* is decomposed into the following set of NIUs:

- *go north*;
- *go past the house*;
- *you are right by the forest*.

At present, human labelers must manually create this decomposition of complex utterances into corresponding NIUs, as well as their constituents (see below). The ultimate goal of our work is to automate this challenging process of robust parsing and semantic analysis. We do not claim that all of the navigational commands can be classified into the four categories listed above. However, in experiments we have found that most of the commands that subjects choose can be classified or decomposed into these categories, and by considering only such commands, we can replicate the paths with reasonable accuracy.

III. Constituents of NIUs

WE would like to understand the referential semantics of NIUs extracted from a sequence of sentences that instruction givers say to instruction followers, in order to execute the instructions encoded within. The type of expected system behavior depends on the category of a particular NIU, and for each category this behavior must be modeled in an appropriate machine representation. Consider the following *move*-instruction: *"move two inches toward the house"*. Its meaning μ can be decomposed into the following constituents:

[1]From now on we refrain from using the term *path* in this sense in order to avoid conflict with the notion of path as an end-to-end navigation route.

$$\begin{aligned}\mu_{\text{move}}\ (\textit{“move two inches toward the house”}) &= \\ \mu_{\text{path descriptor}}\ (\textit{“move ... toward”}) &\times \\ \mu_{\text{reference object}}\ (\textit{“the house”}) &\times \\ \mu_{\text{quantitative description}}\ (\textit{“two inches”}) &\end{aligned}$$

If there is a rule for creating a *“moving toward”*-trajectory with respect to a landmark, then we apply this rule to the object which is the meaning of the *“the house”* expression (its grounding) and follow along this trajectory as far as the meaning of *“two inches”*. In a similar way we can represent meanings of positions, turns and orientations.

To reiterate a point made earlier, the currently implemented system processes NIU-constituents, not speech signal or word transcriptions. While extracting these constituents is a separate research issue that must be addressed in the future (see Section VII), our present goal is to show the viability of this intermediate representation for the understanding task.

In this section we focus on moves because they represent the most frequent and very informative kinds of instructions, while mentioning other NIU-types whenever a particular constituent is relevant for them. Appendix II illustrates the annotation process by listing all NIUs and their constituents extracted from four consecutive (slightly modified) instruction giver sentences taken from one of the MAP-TASK dialogs.

A. Reference Objects

The first constituent type is a *reference object*, which denotes an object that serves as an anchor for identifying directions or positions [12], [13]. In our use of move descriptions, the notion of reference objects is broader than just an object with determinable location in space like the ones in [11]. Directions treated as infinitely remote locations encoded in expressions like *“north”* or *“left”*, can also be used to describe end points or entire trajectories of moves. The advantage of such an approach will become evident when we consider path descriptor constituents below. There are four major types of reference objects that we have observed in the MAP-TASK corpus: *absolute targets*, *relative targets*, *landmarks* and *compound references*:

1) *Landmarks* are the most familiar class of reference objects, they have finite size and are placed at fixed finite locations. Due to the specifics of the MAP-TASK problem where the objects on the map are drawings on a sheet of paper, we further distinguish the subcategory of *page elements* referred by expressions such as *“page center”*, *“lower edge”*, *“upper left corner”* etc. as opposed to *genuine landmarks* (or simply *landmarks*): drawings that have pre-specified names attached to them (such as *FLAT ROCKS* or *SUSPENSION BRIDGE*).
2) *Absolute targets* are infinite points in space, that are fixed at least for the time of interaction (for instance, by being tied to the coordinate system of the immobile instruction giver). In MAP-TASK this is the coordinate system of the map which is oriented in exactly the same way for both instruction giver and instruction follower[2]. Examples of expressions for absolute targets are: *“southwest”*, *“down”*, *“left”* (in the sense synonymic to “west”).
3) *Relative targets* are infinite spatial deictic references whose meaning changes as the navigation session proceeds. They are used to specify directions from the perspective of the traveler that actually moves along the path and are attached to the traveler’s coordinate system. For instance, in the instruction: *“keep moving”* an implicit relative target *FORWARD* is used which lies in an infinitely remote point along traveler’s current orientation. Another example is the expression *“left”*, however this time in the sense of the left side of traveler’s current orientation.
4) *compound references* are real or imaginary objects on the map that require an explicit specification in terms of other reference objects; we will deal with them in detail in Section III-E.

Even though the second and third categories of reference objects look more like directions than “objects”, they share a very important common aspect with the landmarks: they can be used as anchors to bind move trajectories. Before explaining how this can be done, we note that reference objects are equally important for other NIU-types as well (although not all combinations are possible), e.g. one can *“turn to face north”* or *“be above the house”*.

B. Path Descriptors

Path descriptors specify how the trajectory of a move is related to its reference object. In [12], Talmy demonstrated that spatial language is schematic insofar as it reduces the information of a scene down to a body of conceptual material assembled on a skeleton of closed-class elements such as prepositions that define spatial relations (“object dispositions”) in the scene. See [12] for a detailed explanation of possible spatial dispositions and how they are constructed using different prepositions. As far as moves are concerned, Jackendoff [11] distinguishes four categories: *directions* with the reference object on a trajectory extension (expressed by prepositions *“toward”* and *“away from”*), *bounded paths* with the reference object in an endpoint of the trajectory (e.g. *“from”* and *“to”*) and *routes* with the reference object related to some interior point of the trajectory (e.g. *“via”*). We adopt this set of categories, but also extend it to allow each category to be represented by a single rule that we call *path descriptor*. There are 10 path descriptors that are supported by our system (see the upper part of Table I); the trajectory of each of them can be modeled by a circular arc, a straight line interval or a sequence thereof. For instance, we model a TO-move as a straight line between the current traveler location and the closest point from this location that lies on the perimeter of the reference object. In addition, there is one open-end class OTHER to account for all those moves that don’t fit in any of the 10 classes.

Similarly, it is also useful to introduce *position descriptors* for position modeling. Currently our system supports three position descriptors listed with examples in the lower part of Table I.

[2]It is certainly true that the real world orientations of the two maps can be different (instruction giver’s west will be instruction follower’s east if they face each other) but the crucial fact is that both participants understand each other as long as each of them identifies herself with her map.

path/position descriptor	example(s)
TO	*"reach the house"*
FROM	*"leave the forest"*
TOWARD	*"move up"* (abs. ref. obj. NORTH) *"keep going"* (rel. ref. obj. FORWARD)
AWAY_FROM	*"go from east" (to west)*
PAST	*"pass the page center"*
PAST_DIRECTED	*"keep drawing to the left of the rocks"* *"pass right on top of the shack"*
THROUGH	*"follow over the bridge"* *"go across the fields"*
BETWEEN	*"squeeze between the ravine...* *...and the bottom of the page"*
AROUND	*"move around the mill"*
FOLLOW_BOUNDARY	*"follow the lake boundary"*
POS_AT	*"staying close to the beach"*
POS_AT_DIRECTED	*"you are just below the ranch"*
POS_BETWEEN	*"being right between them"*

TABLE I

Path and position descriptors; see the modeling rules in Appendix I.

C. Quantitative Aspect

Some path descriptors such as TOWARD or AROUND under-specify trajectories in that they encode their shape but do not encode how far the traveler should move. In other words, there is a need for a quantitative aspect in the NIU-descriptions which would eventually allow a more precise understanding of commands like *"move two inches down"*. With that in place, the traveler will know exactly what to do: select the absolute target SOUTH, extend a TOWARD-move towards it and follow it for a distance of two inches. Even when a move has an implicit distance specification as in the TO-move *"go to the house"*, the instruction giver still may provide it explicitly (*"go one inch to the house"*) in which case the instruction follower might need to make some adjustments to accommodate it.

The importance of the quantitative element for direction specifications and problems that arise from it have been addressed by many authors (see for instance [14], [15]). We distinguish two dimensions in a space of distance specifications. First of all, a distance can relate to a length of the move itself (as in the examples above) or to gaps between trajectories and reference objects (e.g. *"pass half an inch above the truck"*). Furthermore, there are three distance categories that require different knowledge to model. Modeling is the simplest when exact units are used: *"go about two centimeters to the west"*. Here one merely needs to parse the expression *"two centimeters"* as a measure equal to 2cm. Such commands are commonly observed in the MAP-TASK corpus. When relative units are used as in *"slide down half a page"* or *"move forward the length of the bridge"* (a specification preferred by many authors because it catches relational aspects of distances that define structure of the scene [16]) more situational competence is required. Finally, in the commands like *"keep going for some time"* and *"move a bit more towards page bottom"* the *intuitive* distance descriptions are used that demand significant amount of world knowledge from the interpreter.

Quantitative aspect is also relevant for other NIU types. So, the traveler could be *"three inches to the left of the grove"* in a position specification or he could *"turn forty degrees to the north"*.

D. Coordinate Systems

Many authors have observed that spatial descriptions are given in terms of a coordinate system in which the scene is taking place. For example, position-NIUs *"you are one inch below the house"* and *"the house is one inch below you"* both have reference object "the house" and position descriptor POS_AT_DIRECTED; but their meanings contrast each other clearly, because in the first case the coordinate system is centered in the house, and in the second case in the traveler.

From the perspective of cognitive psychology, the most important question about coordinate systems is whether the system is bound to the experiencer (*egocentric*[3] coordinate system) or is independent of her (*allocentric* coordinate system) [17]. These two major categories can be further subdivided according to where exactly the coordinate system is centered and what it uses as a reference object. Regarding spatial deictic references, Levelt in [13, Chapter 2] distinguishes among the following three major cases:

1) *primary deictic reference*: here the speaker is the origin of the coordinate system and also the reference object (*relatum*); example: *"the ball is in front of me"*;
2) *secondary deictic reference*: speaker is the origin of the coordinate system, but not the reference object: *"the ball is behind the tree"*;
3) *intrinsic reference*: reference object (not speaker) is also the origin of the coordinate system; here the reference object must possess its own "intrinsic" orientation with front and back: *"the ball is in front of the house"* (see also [12, Page 241]).

Similar categorization suggestions can also be found in [18] and others. By virtue of examples above we could see that orientation is indeed important when defining a coordinate system. As an arbitrary coordinate system is defined by a) its origin and b) its orientation, our approach to the spatial language in MAP-TASK is to organize all possible coordinate systems into a two-dimensional grid presented in Table II. Here, there are two possible origin placements and three different orientation types for the coordinate systems in which NIUs can be specified.

There are three possible perspectives in the MAP-TASK: one of the instruction giver, one of the map traveler (often identified with the instructions follower) and finally a perspective from some reference object on the map (landmark or page element). However, only two of them can have an origin associated with them, for instruction giver is not really located "on the map" and can only define orientation. There exists a certain redundancy in the choice of a coordinate system and specification of reference objects: for instance, whenever the reference object of a move is a relative target, the coordinate system is always placed where the traveler is and oriented according to the traveler's orientation. Besides,

[3]In reality, egocentric system itself is hypothesized to be an acquired complex coordination of several sensory-motor manifolds [17].

		orientation		
		absolute	traveler	reference object
origin	ref. object	*"pass to the left of the page center"* (move)	*"descent along your side of the rocks"* (move)	*"you are at the base of the monument"* (position)
		"turn to face north" (turn)	*"go to the other side of the lake"* (move)	*"slide a few inches down the river"* (move)
	traveler	*"go due southwest"* (move)	*"this house should be on your right"* (orientation)	?
		"the lake above you" (position)	*"turn around"* (turn)	

TABLE II
Types of coordinate systems in which navigational information units can be specified with NIU-examples.

the exact specification of a coordinate system or at least of its orientation can be irrelevant for certain NIU-types. For instance, in the position-NIU *"it's close to the barn"* orientation of the coordinate system (which is placed in "the barn") can not be determined and is in fact not needed for understanding.

E. Compound References

If the language of instruction givers were constrained to the kind of examples we have seen before, it would have only a limited expressive capacity because it wouldn't allow for a very large portion of potential reference objects to be taken into account. The set of landmarks and page elements is too sparse and lacks the needed expressive means to allow for high-precision navigation. Instructions such as *"go to the lake"* under-specify the required action, because the lake can occupy a large portion of a map. Instead something like *"go to the north-west corner of the lake"* is needed. Similarly, if the target of a particular TO-move is a point an inch above the house, there's no way to avoid an explicit specification of this point: *"move to a spot slightly above the house"* or even simpler: *"move above the house"*. All these are examples for what we call *compound references*. Nested compound references are also possible: *"continue towards [the spot an inch under [the bottom of the monument]]"* as well as compound references that have stretch (*"you are level with the springs"*).

The description language for compound references is very similar to the one of positions, except for one special case where the compound reference is a part of its own reference object as in *"you should be right under the gate of the castle"*, and so (even though their semantic role is clearly different from that of actions and verifications) we decided to include compound references in the list of supported NIU types (see Fig. 2). In the present version, we are not trying to estimate positions of compound references, instead we assume they are known and thus look them up in manual annotations.

F. Designing and Validating Rules

Up to now we have largely ignored the question of how to model individual NIUs, concentrating mainly on the possibility of such modeling. This section describes how we design and use a lexicon of action and verification primitives.

The first part of the process is a manual step of designing prototypes for each rule. Then, for each NIU type we compare the designed prototype with actually observed instances[4] in order to estimate *spatial templates* [19]. Later, when path intervals corresponding to individual NIUs are merged together to form a continuous replica of the reference path, these spatial templates will guide this process, helping find the most probable realization for each NIU that allows for such a merge.

We first consider position-NIUs of type POS_AT_DIRECTED. There is extensive prior work on modeling spatial language (e.g. expressions *"above"*, *"to the left of"* etc.) [20], [19], [21]. We chose an easily implementable model similar to the Hybrid Model of [21] to model these NIUs. Two metrics determine goodness of a particular position with respect to its reference object: the angle between the defining axis (e.g. vertical axis in case of "above") and the beam emanating from the reference object's center of mass and passing through the position; and second, the projection of the distance from the extreme point of the reference object in the given direction (e.g. the highest point for "above") to the position on the defining axis (e.g. distance of y-coordinates for "above"). For NIUs of type POS_AT there is only one metric: absolute distance from the position to the closest point of the reference object.

For moves and turns, creating spatial templates is similar: first, for each actually observed move/turn we seed its corresponding prototype into its starting point, and thus obtain its *predicted* version. Then, *radial* and *angular deviations* between the end points of the predicted and observed path intervals are computed. Prototypes and intuitive explanations of these distances for moves of types TO and FOLLOW_BOUNDARY are shown in Fig. 3. Here, we execute a FOLLOW_BOUNDARY-move by "expanding" the perimeter of the reference point to traverse the current traveler position, and moving along this expanded perimeter in that direction (of the two possible) that has the smallest angle with the orientation that the traveler had before reaching her current position[5]. For a detailed investigation of when and how people use path descriptors of this type as opposed to path descriptors of type PAST, see [22].

In short, radial distance is the difference in lengths of the observed and predicted moves, and angular deviation shows how far from the predicted trajectory the actual trajectory deviates. For some of the prototypes (like, for instance, TOWARD-moves), we need to provide not only the path descriptor but also the default distance. In our experiments this distance was estimated empirically.

[4]As we have mentioned earlier, these instances (path intervals) are part of manual annotations that we create prior to the experiments.

[5]In practice, we used a computation scheme for angular and radial deviations for FOLLOW_BOUNDARY which is slightly different from the depicted one and approximates it instead.

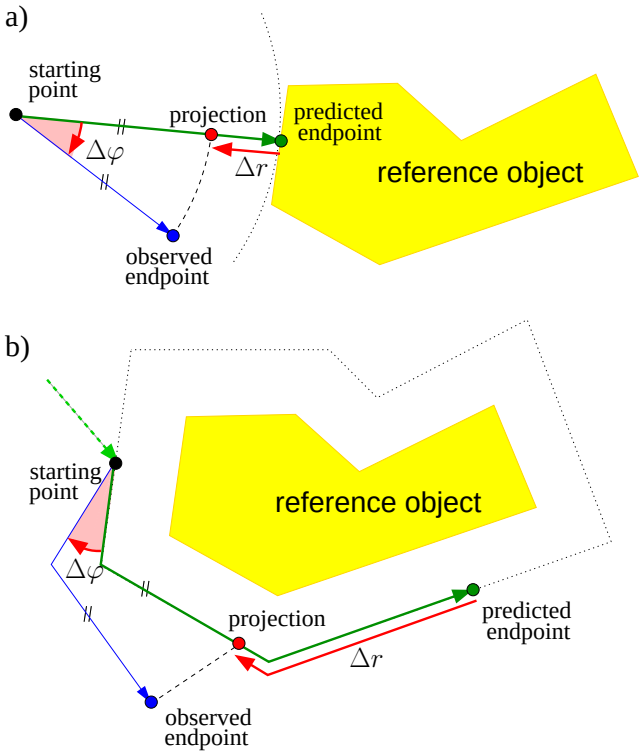

Fig. 3. Prototypes and radial and angular deviations for moves of types a) TO b) FOLLOW_BOUNDARY.

Angular and radial deviations collected for each NIU type are compiled into a two-dimensional spatial template. This template contains probabilities of all realizations of this NIU type that start in the same point but deviate from the prototype in due course. This is also why there is no need for searching for the perfect prototype for each NIU type. Indeed, the spatial template will compensate for potential mistakes.

To summarize, the models of individual NIUs are combinations of handcrafted structures combined with data-driven parameter adaptation that make these models flexible. Positions are 2D probability distributions for all locations, and moves and turns are 2D probability distributions as well, but for the end points of actions they represent (and indirectly also for different trajectories that lead to these endpoints). In experiments reported below, we ignore orientation-NIUs because they occur fairly seldom in the corpus.

IV. Combining NIUs into Contiguous Paths

NOW that we have described how NIUs are grounded in particular action and verification primitives, we turn to the issue of combining sequences thereof into a contiguous path on the map. This task can be considered as the one of route planning, where instructions are given before the actual following of the route takes place.

A. Dynamic Programming Approach

The "plan-as-communication" view on plans in [23] suggests that plans constrain possible space of actions and require some interpretative effort from the agent whenever execution of a particular action is due; in other words, it enforces considering each action in a context of the whole plan and of the environment the plan operates in. Our approach is motivated along similar lines. In our system, interpretation of instructions like *"turn left"* or *"go to the house"* takes place only when they are next to be executed, and probabilistic assessment of geometries of their reference objects with respect to traveler's current position and orientation can be made. Only then "left" and "go to" will for the first time acquire a concrete meaning attached to them. However, this meaning is by no means final, for instructions that follow can still change it later on. If we do have a map in front of us at the planning time, we can mentally follow the route right away and "rehearse" execution of all navigational instructions in the sequence one by one. If at some point in time we realize that a mistake has been made on previous stages because no consistent continuation of the path is possible, we can always back-off to the point where the mistake was made and choose other alternatives leading from there. Moreover, we can simultaneously maintain several alternative routes in the first place, scoring and extending them in parallel as we progress and dynamically preferring one of them over the others. This view of the task suggests a dynamic programming approach.

Dynamic programming, however, can not operate on a continuum of $\mathbb{R}^2$, which is the case for maps in MAP-TASK, but rather needs a set of discrete alternative states. In order to achieve that, we impose a rectangular grid on the maps and consider only cell centers as potential alternatives for traveler's locations at the end of each action.

At this point let us restrict the NIUs to moves and turns only, and assume that the entire path is split in N intervals each of which is covered by exactly one NIU[6]. Let us also assume that our map is split in I square cells c_i, $i \in \overline{1,I}$. Then, on each step $n \in \overline{0,N}$ there will be a separate probability distribution of ending up in cell c_i $\forall i \in \overline{1,I}$ after this step has been taken. Since the starting point of the path is considered given, the initial probability distribution ($n = 0$) is 0.0 for all cells except the one containing the starting point, where it is 1.0.

Assume now that we know probability distribution $p_n(j)$ on step n. Conditional probabilities $p_{\gamma_n}(i|j)$ of ending NIU γ_n in c_i given that it starts in c_j can be interpolated for all $j \in \overline{1,I}$ from spatial templates that have been discussed in the previous section (see Fig. 4).

Then, the total probability of reaching c_i on step n and passing through c_j before that is:

$$p_{n+1}(i,j) = p_n(j) \cdot p_{\gamma_n}(i|j). \quad (1)$$

With the decision-oriented approach to probabilities, we select the predecessor index j^*:

$$j^* = \operatorname*{argmax}_j p_{n+1}(i,j) \quad (2)$$

such that cell c_{j^*} is the predecessor of c_i on the optimal path and declare:

[6]We will show later how this unrealistic assumption can be removed.

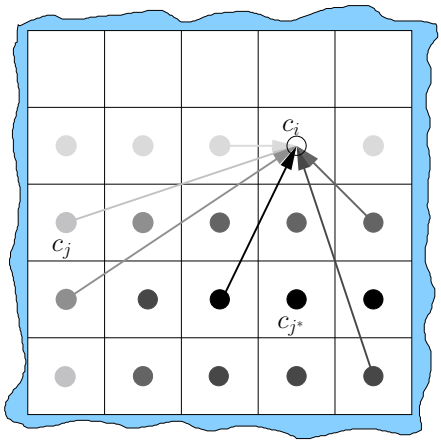

Fig. 4. Computing probabilities of endpoints for NIU: *"go north"*; for each cell c_j, probability of starting there and ending up in cell c_i is encoded as the darkness of the corresponding arrow and of the cell center; it depends on the deviation from the north direction as well as on the distance between c_j and c_i.

$$p_{n+1}(i) := p_{n+1}(i, j^*). \tag{3}$$

After all N NIUs have been processed the entire optimal path must be recovered. For that, we start in the cell c_{j^*} from the last distribution and traverse backwards the sequence of the distributions for all NIUs, following the line of predecessors all the way up to the very first distribution.

It is important to see that in this simplified task formulation, the "winning" cells describe not only the most probable position after having processed the last NIU but also the orientation of the traveler that goes along with this particular NIU realization.

B. Natural Task Constraints

At first sight it might appear unclear why spatial templates are needed at all. Indeed, if no constraints existed in the task, for all n's the n'th step of the dynamic programming algorithm would always result in selecting the realization of γ_n that possesses angular and radial deviations corresponding to the maximum in γ_n's spatial template. Fortunately, there are in fact several constraints that come along with the choice of the task domain that we call "natural" constraints of the task. In general, we can say that these constraints are task dependent, and arise from the working conditions of the system.

First, our domain expertise and intuition suggest that some of the landmarks can not be crossed. The maps designed for MAP-TASK comply with this consideration to a great extent. For example, while a path can pass through the drawing of a bridge, it will never cross a rock. This knowledge is one of the main sources of constraints that shape possible paths. In the same way we might want to prohibit self-crossings of a path and restrict all the paths to the inside of the visible map. One should keep in mind however that banning self-crossings hurts the optimality principle of the dynamic programming saying that solutions of partial problems never need to be recalculated [24], and thus can lead to not finding a good path.

Second, we propose that a reasonable bias is to favor junctions that are smooth, i.e. large changes in orientations when finishing modeling γ_{n-1} and starting γ_n should be penalized. In particular this bias will eliminate abrupt near-180° turns.

Finally, for many move types that have landmarks for reference objects, it is reasonable to presume spatial proximity of the starting point to the landmark. In fact, in the MAP-TASK corpus, there are rarely instructions to move around some landmark that is on the other end of the map, far away from our current position. This means that only those realizations of such an NIU that ended close to the landmark will be considered in the next step. Indirectly, these constraints are modeled in spatial templates; however, we found out that imposing explicit upper thresholds on maximum distances between starting point of an NIU and its reference point is helpful as well. Besides, we can require a certain degree of consistency from action trajectories and end points. For instance, the end point of a BETWEEN-move must indeed lie between its reference objects, and the PAST_DIRECTED move expects the instruction follower to be on a particular side of the reference object.

C. Integrating Positions

Another source of constraints are verification-NIUs, in particular positions. They too restrict the working space to a small area tied to a reference object, or (in the case of POS_AT_DIRECTED) to one of its sides. Consider how positions can be integrated in the framework we have developed so far. Recall that we update the distribution of locations after each action. Similarly, we can update them after each position specification as well. Here however, we can multiply the position probabilities $p_{\gamma_n}(i)$ (interpolated from the corresponding spatial template) with the distribution $p_n(i)$ obtaining a new adjusted distribution $p_{n+1}(i)$ as:

$$p_{n+1}(i) := p_n(i) \cdot p_{\gamma_n}(i). \tag{4}$$

Even though we don't model rare orientation-NIUs in the experiments of this work, they can be handled in exactly the same way as positions.

D. Dealing with Redundancy

No matter how many position specifications there are, all of them can be subsequently treated as shown in (4). This however is not true for actions. If several action NIUs compete for one path interval or even describe path intervals that only start in approximately the same location, their contributions must be considered simultaneously. Let $\Gamma_n = \{\gamma_n^k\},\ k \in \overline{1, K_n}$ be a set of NIUs competing to define next path interval. In order to compute joint probabilities $p_{n+1}(i, j)$ we average over individual NIUs in Γ_n and, assuming equal priors for all NIUs in the set, modify (1) into:

$$p_{n+1}(i, j) = p_n(j) \frac{1}{K_n} \sum_k p_{\gamma_n^k}(i|j). \tag{5}$$

After that, selection of the optimal predecessor and computation of the next distribution $p_{n+1}(i)$ is done as before.

In contrast to the remark at the end of Section IV-A, the resulting orientation after Γ_n is yet to be determined, since it is a product of several NIUs at the same time. Our approach to this problem is to select the NIU that delivers the highest conditional probability $p_{\gamma_n^k}(i|j^*)$ to represent the group, but to use the orientation computed as a weighted average over all NIUs in Γ_n to control smoothness of the path.

One issue we haven't addressed yet is how to establish a partial relation on all NIUs of a session, i.e. which NIU should be considered when, and what are the sets Γ_n of competing NIUs. For now, our system looks up this information in the annotations, looking at the starting points of all NIUs[7]. This information is usually contained in the language, and from proximity of NIUs in dialog transcriptions we can usually conclude at least on proximity of the intervals they describe. For instance, the sentence: *"go left to the creek"* contains two NIUs: *"go left"* and *"go to the creek"* that should be put in the same set Γ_n. A more sophisticated linguistic analysis is required if precedence must be determined as well (see Section VII). In other situations where the instruction giver comes back to one of the already described path intervals reiterating or rephrasing extractions given earlier, there are dialog context clues (e.g. instruction follower's feedback) to signal this fact.

We employed one general rule regarding splitting the set of NIUs with close starting points that resulted in performance improvement: if there are verification-NIUs, we first create a set out of them. Then if there are action-NIUs that are reliable in guessing directions, such as TO- and TOWARD-moves, and turns with absolute targets or landmarks as reference objects, we make a separate set out of them, and process this set only after the first one. Next, a set of other moves with such reference objects is processed. And finally, if and only if no action groups could be created, actions with relative targets as reference objects are considered.

V. Evaluation Metrics

THERE are two classes of evaluation metrics that are of interest for this work. The first class of *instruction-level metrics* concerns modeling of individual NIUs and sheds light on quality of path descriptor rules by assessing deviations of observed actions around their prototypes. The second class of *path-level metrics* evaluates entire paths and judges their overall quality by comparing them to their references on the instruction givers' maps.

On the instruction level, we can judge shapes and, in particular, compactness of spatial templates (distributions of angular and radial deviations). Visual assessment is important for the entire paths; however, we can also use criteria such as percentage of landmarks on a correct side of the path and average trajectory deviations to perform their formal evaluation. A reasonable figure of merit for the latter is the area between the observed and predicted paths. If we augment the predicted path, so that it ends in exactly the same point where the reference path ends (human objects in the original MAP-TASK experiment knew the position of the finish) then the two paths together will form a closed contour, and we can use standard filling techniques such as SCAN LINE algorithm [25] to compute the cumulative area of the "mismatch"-zones (see Fig. 5). The smaller the area the better the model.

[7]Note that we don't look up the exact positions of these starting points on the path, but rather only the fact that they are close for two or several NIUs.

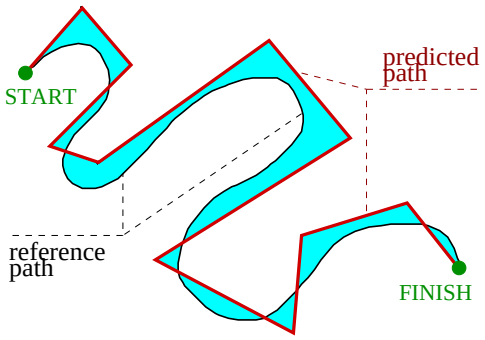

Fig. 5. Area between observed reference path and predicted paths can be used to assess quality of modeling.

VI. Results

OUR experiments were conducted on the commercially available HCRC MAP-TASK corpus [6]. This corpus consists of 128 navigation sessions with audio recordings and a number of different annotations available in XML-format for each session. We randomly selected 25 of these sessions for our experiments, and focused our analysis on instruction givers' speech. Based on a set of previously existing annotations of this speech data in terms of *moves in conversational games* [26] we selected the subset annotated as either *Instructions* or *Clarifications*. These sentences were then manually annotated with respect to the NIUs they contain. We defined the NIUs reported in this paper on the basis of analyzing only five of the 25 sessions. On average, we obtained 85 NIUs per session. Relative frequency distributions of categories of the extracted NIUs as well as of move types within the move category are shown in Fig. 6. These plots show that moves clearly dominate among all analyzed NIUs and the TOWARD type that includes instructions such as *"draw your line towards the northeast"*, *"move down"*, *"keep going"* etc., is the most frequent type among moves. Less than 5% of the 2133 annotated NIUs could not be identified as one of the five NIU categories from Fig. 2 and less than 5% of the 1526 annotated moves have been labeled with the path descriptor OTHER. The high coverage of these NIUs for the complete set of 25 sessions suggests that this set of NIU-models is well suited to the task, and perhaps also spatial language for navigation tasks more generally.

Yet another reassuring confirmation comes from the following measurements: inter-annotator agreement with respect to the extraction of NIUs and their labeling with one of the five

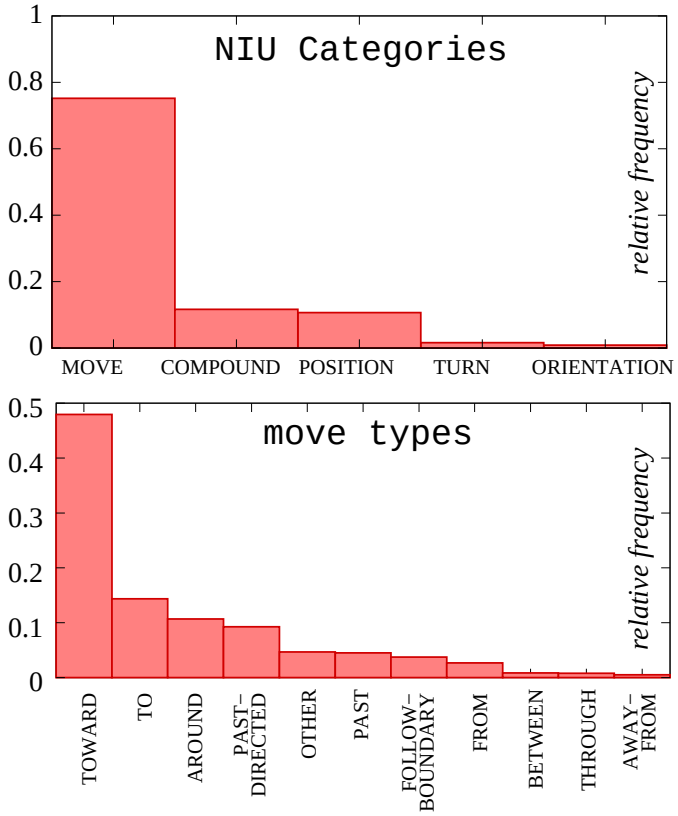

Fig. 6. Occurrence statistics of NIU-categories and move types.

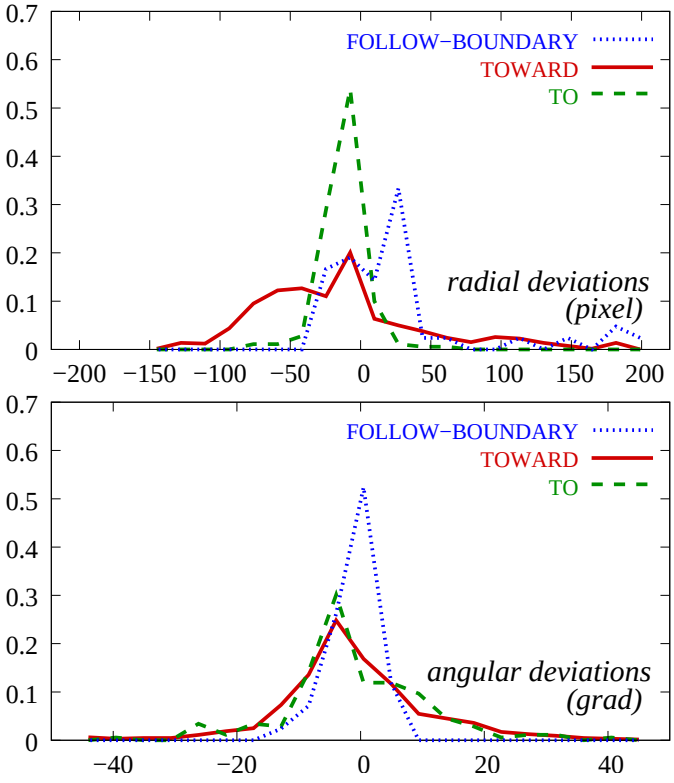

Fig. 7. Angular and radial deviations statistics estimated for moves of types FOLLOW_BOUNDARY, TOWARD and TO.

supported types as well as with respect to path descriptors of detected moves was roughly estimated for two labelers using the F-measure. It amounted to 0.86 and 0.8 respectively, with the absolute majority of mismatches due to ambiguities of cases like *"go above the house"* which can be interpreted either as a TO-move with a compound reference object, or as a PAST_DIRECTED-move. Nonetheless, our experiments showed that such ambiguities don't impair the understanding of complete paths.

We then produced discrete versions of the reference paths, representing them as sequences of many "stops" placed densely along the original curve. Each NIU was annotated with a path interval (delimited by the first and last stops on the path) that it, in labelers' view, accounts to. Based on these annotations, we estimated spatial templates for each of the move types, position types and turns expressing them in terms of radial and angular deviations from manually designed prototype rules. As expected, the main source of deviations came from the under-determined quantitative aspect of NIUs; for example, in Fig. 7 we see that while estimated angular deviations statistics possess rather compact distributions (meaning that the rules we designed to represent these move types are in fact consistent with annotations), radial deviations of the moves (variations of their stretches) along the given trajectory are flatter for those move types that intuitively require explicit stretch specification (such as TOWARD-moves).

Next, we show how the dynamic programming approach can be used to integrate models of individual NIUs in a joined consistent and smooth path. A snapshot of one of the replicated paths (in progress) is shown in Fig. 8. Several aspects of this path are noteworthy. First, the snapshot sheds light on the way the drawing is discretized. For each landmark we marked its perimeter that can not be crossed unless there is no other way to proceed with the path. This otherwise exceptional case happened to occur here at the beginning, where the instruction giver insisted on going down and the spatial template for a TOWARD move prohibits angular deviations of more than 50 degrees. Also, the rectangular grid of cells for which a new distribution is estimated after each processed group of NIUs can be seen here: the darker the cell the higher the probability of ending up there; the cell with the highest probability is chosen to determine the most probable path so far (sequence of circles and lines and arcs between them). The distribution in this snapshot takes place after one NIU that sends the traveler a specified distance towards southwest and another one that commands to go on along the same direction. From this distribution it can be seen that self-crossings are prohibited, and that perplexity of such a distribution can get very high. One of the possibilities to reduce the perplexity is to issue a verification-NIU. In the presented session, the next instruction was indeed position-NIU *"near to the abandoned cottage"*, and the new distribution with lower perplexity resulting from it is shown in the excerpt of the map in Fig. 9. It is more compact with the only allowed cells located around the landmark.

As far as the quality of the predicted path, it can be seen that it lies reasonably close to its reference path. In order to formalize the visual assessment, we computed areas of the "mismatch"-zones for each pair of reference/predicted paths

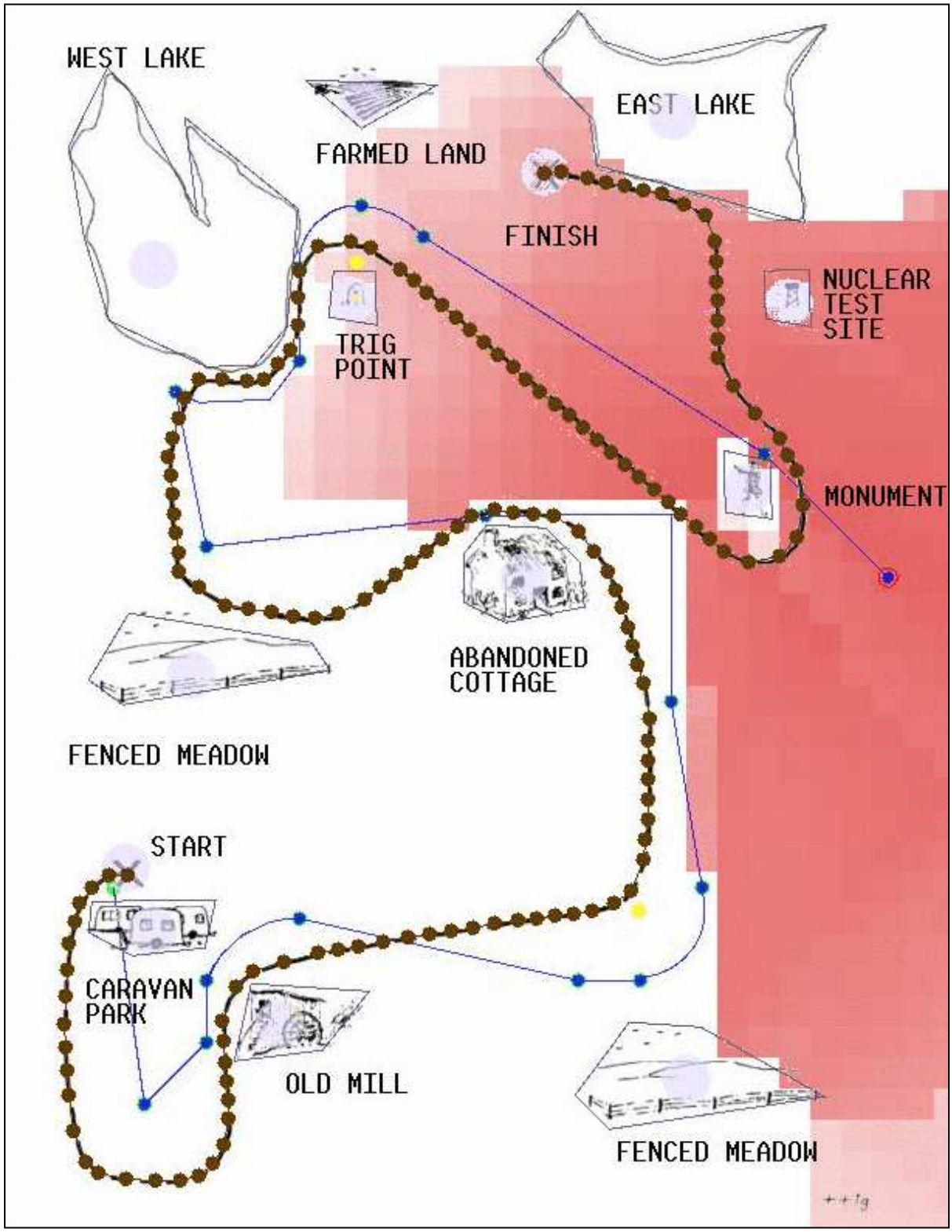

Fig. 8. Snapshot of a modeled path after processing several groups of NIUs.

(see Section V) and normalized them by the length of the corresponding reference paths. This criterion can be interpreted as the average diameter of an "error tube" of deviations that we enwrap the reference paths into. The smaller the error tube diameter, the more precise the modeling mechanism. Evaluation of predicted paths for all 25 sessions resulted in an average diameter of 19.5 pixels (one fortieth of the height of the maps) with a sample deviation of 5 pixels. For comparison, the baseline strategy of connecting start and finish landmarks with a straight line resulted in an average error tube diameter of 280 pixels. We take this as a clear sign of success for our modeling algorithm. Unfortunately, the original corpus contained no paths drawn by instruction followers on their maps. This kind of information would have provided another important baseline for our experiments.

In order to investigate the importance of the natural constraints and verifications for a successful path modeling (Section IV-A), we conducted one replicating experiment without any restrictions on landmark- and self-crossing and another one where all the position-NIUs were ignored[8]. For the first experimental set-up we obtained an average diameter of the error tube of 23 pixels (sample deviation 5.5 pixels). For the second one, one session couldn't be completed at all, and for those that could be completed, we obtained an average diameter of 21 pixels (sample deviation 5 pixels) which amounts to a relative precision loss of 18% and 7% respectively. In terms of a number of landmarks passed on the wrong side, removing all position-NIUs increased their proportion by almost 50% relative. All of the above experiments were conducted using *Leave One Out* strategy, i.e. in order to replicate each session, we trained the spatial templates on the remaining 24.

These results demonstrate the importance of natural constraints and verifications in navigational tasks. Fig. 10 shows error tube diameters for all sessions for all these experiments where sessions are arranged in such an order that the diameter increases for our final system with no landmark- and self-crossings allowed and with position-NIUs accounted for.

Yet another promising result comes from ignoring stretch specifications for moves. In the previous experiments, if the quantitative constituent of a move was specified, we would temporarily shift a corresponding spatial template to peak in this stretch. However, looking up the meaning of expressions like *"a little bit"* isn't quite fair, because it requires serious semantic analysis and a great deal of world knowledge. As it turns out, we can ignore such explicit specifications altogether, and the integration procedure will still deliver accurate models. In our experiments the average error tube diameter remained under 20 pixel.

VII. Discussion

We have reported first steps towards automatic understanding of unconstrained navigational instructions in the Map-Task domain. Clearly, substantial aspects of the problem remain unmodeled and pose significant challenges for future research. For example, we still need to extract

[8]The remaining moves still had a number of constraining elements (like PAST_DIRECTED or TO moves), so that the modeling didn't break apart.

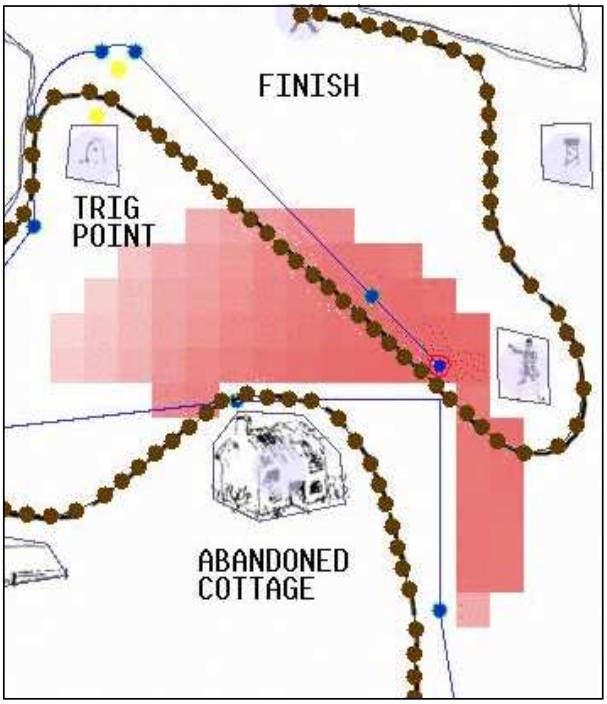

Fig. 9. Excerpt of the same path after the next instruction (position-NIU *"near to the abandoned cottage"*).

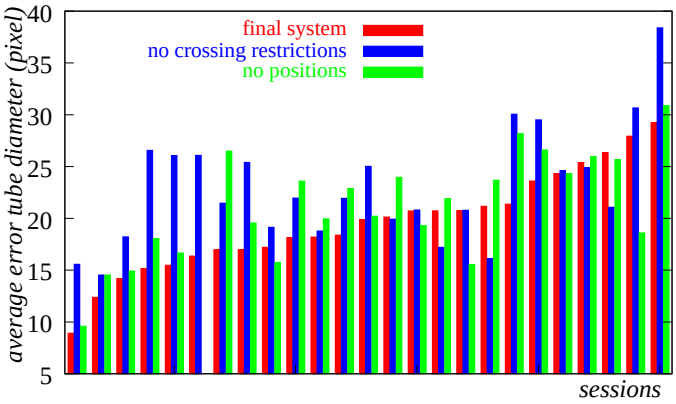

Fig. 10. Error tube diameters for all sessions computed for cases where: all available information is used; no landmark- and self-crossing restrictions are imposed; only move- and turn-NIUs are considered.

NIUs from instruction transcripts, impose a partial precedence relation on them, and understanding of meaning of distances and angles depends on manual interpretation as well. As far as the latter is concerned, we showed in the previous section that the quantitative aspect of NIUs can be ignored without significant loss in performance if we consider them in context of other NIUs. Extracting NIUs from text is a task similar in spirit to the task of named entity extraction and may be achieved using well-established tagging algorithms [27]. Our preliminary experiments in this direction produced promising results (not reported in this paper). Determining partial order of NIUs would remain a challenge. For instance, the instructions *"at the corner go left"* and *"go left till you are at the corner"* both contain one position describing being at the corner and one move describing going to the left, but in the

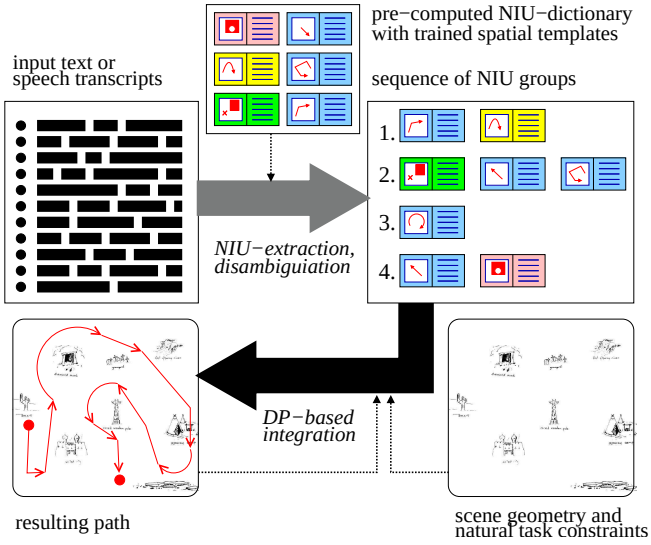

Fig. 11. Automatic path replication; system diagram.

first case the position precedes the move and in the second it's the inverse. Ultimately, more robust syntactic processing must be brought to bear on the problem. We believe our approach of propagating spatial constraints based on NIUs in a dynamic programming framework provide an extensible framework for such future investigations.

To put the work completed thus far into context from a practical perspective, we sketch how the implemented components might play a role in a larger end-to-end language understanding system. Extraction of NIUs from text or speech transcripts is the only transition that hasn't been completely automated yet; it is identified by the thick gray arrow in Fig. 11. Given a sequence of extracted groups of competing NIUs whose spatial templates are learned from a training corpus, we then use the DP-based algorithm from Section IV-A to produce a contiguous path, while also taking into account scene geometry and the scenario's natural constraints. In a video game scenario mentioned in the beginning of this paper, for instance, our system can reside inside of an action interpreter that reads multi-step natural language path descriptions submitted by the players via keyboard or microphone and sends virtual game characters to follow the complex trajectories that arise from them.

VIII. Conclusion

We have described a system that infers paths on maps by processing natural language instructions represented as Navigational Information Units (NIUs). This translation process from linguistically-derived symbolic representations to geometric spatial representations is an example of language grounding (see [28], [29]). Our focus in this effort was to automate the translation of NIUs into probability distributions over possible paths on maps. We defined four categories of NIUs: moves, turns, positions and orientations and developed an approach for composing NIUs in order to interpret the semantics of complex natural utterances that are analyzed as comprising multiple NIUs. In evaluations this approach successfully produced semantically correct interpretations for a wide range of utterances.

Appendix I
Definitions of Path and Position Descriptors

This section explains path and position descriptors that can be used to classify moves and positions, as well as turns:

1) TO:
in a straight line approach the closest point of a reference object;
2) FROM:
keeping previous direction, make sure the move goes away from a reference object;
3) TOWARD:
move in the direction of center mass of a reference object;
4) AWAY_FROM:
move in the direction opposite to center mass of a reference object;
5) PAST:
keeping previous direction proceed in a straight line up to the point where the farthest point of a reference object projects on this direction;
6) THROUGH:
in a straight line proceed through the center mass of a reference object and up to its farthest point in this direction;
7) PAST_DIRECTED:
this path descriptor can have one of the following 4 subcategories (sides): "above", "below", "to the left of" and "to the right of" a reference object. It consists of one or two straight line intervals. If the traveler is not already on the required side of a reference object, he has to take the shortest path to get there (possible directions: north, south, west, east). The second step leads from there past projection of the center mass of a reference object on the required side to a projection of the farthest point of the reference object on it;
8) AROUND:
in a circular arc move around the center mass of a reference object; among two possible initial directions select the one closest to the previous direction;
9) FOLLOW_BOUNDARY:
"expand" the perimeter of a reference object to pass through the starting point of the move. Follow this expanded perimeter; among two possible initial directions select the one closest to the previous direction;
10) BETWEEN:
this move requires two reference objects. Compute intervals of view angles not crossing any of the reference objects and consider two directions in their middles. Select the one closest to the previous direction and proceed in a straight line up to the projection of the farthest point of both reference objects on this direction;
11) TURN:
turns are modeled similar to the AROUND-moves with a small radius arc. Traveler follows the arc till needed orientation is achieved;

12) POS_AT:
this position descriptor generates score that depends on traveler's distance to the closest point of a reference object;
13) POS_AT_DIRECTED:
similar to PAST_DIRECTED, there are four possible sides for this position descriptor ("above", "below", "to the left of" and "to the right of"), each one determining its active direction (e.g. north for "above"). The score depends on traveler's distance along the active direction to the reference point's extreme in this direction, and also on angle that a beam from the reference point's center mass to the traveler creates with the active direction;
14) POS_BETWEEN:
here the score is generated based on a difference between distances from the traveler to the closest points of first and second reference objects.

APPENDIX II
ANNOTATION EXAMPLES

Consider following four (slightly modified) instructions from one of the MAP-TASK dialogs (see also Fig. 8):

- *"Continue up north slightly."*
- *"...to the tip of the lake."*
- *"...and then we're going to turn down above the trig point."*
- *"...and we're going to turn immediately to your right."*

For these sentences the following NIUs have been annotated:

1) MOVE with path descriptor TOWARD (*"continue"*), absolute reference object NORTH (*"up north"*) and intuitive stretch from start (*"slightly"*);
2) COMPOUND REFERENCE of the type PART-OF (*"the tip of"*);
3) MOVE with path descriptor TO (*"to"*) and the above compound reference as a reference object;
4) TURN with absolute reference object SOUTHEAST (*"down"*, southeast direction observed on the map);
5) POSITION with position descriptor POS_AT_DIRECTED (*"above"*) and coordinate system with a center in the reference object TRIGPOINT (*"trig point"*) and absolute orientation;
6) TURN with relative reference object RIGHT (*"your right"*);

ACKNOWLEDGMENT

The authors would like to thank all the members of the Cognitive Machines group who shared their views on the task and provided useful remarks regarding the methods and algorithms employed in this research. We also thank Stefanie Tellex, Rony Kubat, and anonymous reviewers for their comments on earlier drafts of this paper. This work was supported in part by NSF grant ITR-6891285.

REFERENCES

[1] Simmons, R., Goldberg, D., Goode, A., Montemerlo, M., Roy, N., Sellner, B., Urmson, C., Schultz, A., Abramson, M., Adams, W., Atrash, A., Bugajska, M., Coblenz, M., MacMahon, M., Perzanowski, D., Horswill, I., Zubek, R., Kortenkamp, D., Wolfe, B., Milam, T. and Maxwell, B.: *"GRACE: An Autonomous Robot for the AAAI Robot Challenge"*; AI Magazine 24(2), pp.51–72, 2003.

[2] Miranda-Palma, C. and Mayora-Ibarra, O.: *"Robotic Remote Navigation by Speech Commands with Automatic Obstacles Detection"*; in Robotics and Applications, pp.53–57, 2003.

[3] Bugmann, G., Klein, E., Lauria, S. and Kyriacou, T.: *"Corpus-Based Robotics: A Route Instruction Example"*; in Proc. of IAS-8, pp. 96–103, March 2004, Amsterdam.

[4] MacMahon, M.: *"MARCO: A Modular Architecture for Following Route Instructions"*; in AAAI-05 Workshop on Modular Construction of Human-Like Intelligence, Pittsburg, PA, July 2005.

[5] Tellex, S., Roy, D.: *"Spatial Routines for a Speech Controlled Wheelchair"*; submitted to HRI-2006.

[6] Anderson, A. H., Bader, M., Bard, E. G., Boyle, E. H., Doherty, G. M., Garrod, S. C., Isard, S. D., Kowtko, J. C., McAllister, J. M., Miller, J., Sotillo, C. F., Thompson, H. S. and Weinert, R.: *"The HCRC Map Task Corpus"*; in Language and Speech, 34(4), pp.351–366, 1991.

[7] Tenbrink, T., Fischer, K. and Moratz, R.: *"Spatial Strategies in Human-Robot Communication"*; in "Themenheft Spatial Cognition", Freksa, Ch. (ed.), KI 4/02, arenDTaP Verlag, 2002.

[8] Riesbeck, C.: *"You Can't Miss It: Judging the Clarity of Directions"*; in Cognitive Science, Vol. 4, pp.285–303, 1980.

[9] Tversky, B. and Lee, P. U.: *"Pictorial and Verbal Tools for Conveying Routes"*; in "Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science", Freksa, C. and Mark, D. M.(eds.), pp.51–64, Springer, Berlin, 1999.

[10] Denis, M.: *"The Description of Routes: A Cognitive Approach to the Production of Spatial Discourse"* in Current Psychology of Cognition, Vol.16, pp.409–458, 1997.

[11] Jackendoff, R.: *"Semantic and Cognition"*; MIT pres, Cambridge, MA, 1983.

[12] Talmy, L.: *"How Language Structures Space"*; in "Spatial Orientation: Theory, Research and Application"; Pick and Acredolo (eds.), Plenum Publishing Corp., NY, 1983.

[13] Levelt, W. J. M.: *"Speaking: From Intention to Articulation"*; MIT Press, Cambridge, Massachusetts, 1989.

[14] Montello, D.: *"The Geometry of Environmental Knowledge"*; in "Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science", Frank, A., Campari, I. and Formentini, U. (eds.), Vol.639, pp.136–152, Springer-Verlag, Berlin, Germany, 1992.

[15] Egenhofer, M. and Shariff, A.: *"Metric Details for Natural-Language Spatial Relations"*; in ACM Transactions on Information Systems, 16:(4), pp.295–321, 1998.

[16] Hernández, D: *"Qualitative Representation of Spatial Knowledge"*; Springer-Verlag, 1994. New York, NY.

[17] Grush, R.: *Self, World and Space: The Meaning and Mechanisms of Ego- and Allocentric Spatial Representation*; in "Brain and Mind", Vol. 1, No. 1, pp.59-92(34), April 2000.

[18] Levinson, S. C.: *Frames of Reference and Molyneux's Question: Crosslinguistic Evidence*; in "Language and Space", Bloom, P., Peterson, M., Nadel, L. and Garrett, M. (eds.), pp.109–169, MIT Press, Cambridge, MA, 1996.

[19] Logan, G. D. and Sadler, D. D.: *A Computational Analysis of the Apprehension of Spatial Relations*; in "Language and Space", Bloom, P., Peterson, M. A., Nadel, L., and Garrett, M. (eds.), pp.493–529, MIT Press, Cambridge, MA, 1996.

[20] Gapp, K. P.: *"Angle, Distance, Shape, and their Relationship to Projective Relations"*; in Proc. 17th Conf. of the Cognitive Science Society, pp.112–117, Mahwah, NJ, 1995.

[21] Regier, T. and Carlson, L.: *"Grounding Spatial Language in Perception: An Empirical and Computational Investigation"*; in "Journal of Experimental Psychology", Vol.130, No.2, pp.273–298, 2001.

[22] Zimmer, H., Speiser, H., Baus, J. and Krüger, A.: *"Critical features for the selection of verbal descriptions for path relations"*; in Cognitive Processing, 2001.

[23] Agre, P. E. and Chapman, D.: *"What Are Plans For?"*; in Robotics and Autonomous Systems, Vol. 6, pp. 17–34, 1990.

[24] Aho, A., Hopcroft, J. and Ullman, J.: *"The Design and Analysis of Computer Algorithms"*; Addison-Wesley, Reading, MA, 1974.

[25] Foley, J. D., van Dam, A., Feiner S. K. and Hughes, J. F.: *Computer Graphics: Principles and Practice in C*; 2nd edition, Addison Wesley, 1997.
[26] Power, R.: *The Organization of Purposeful Dialogues*; Linguistics, Vol.17, pp.107–152, 1979.
[27] Bikel D., Schwartz R., Weischedel R.: *"An Algorithm that Learns What's in a Name"*; "Machine Learning", special Issue on Natuaral Language Learning, Vol.34, no.1–3, pp.211–231, 1999.
[28] Roy, D.: *Grounding Words in Perception and Action: Computational Insights*; in "Trends in Cognitive Science", 9(8), pp.389–396, 2005.
[29] Roy, D..: *Semiotic Schemas: A Framework for Grounding Language in Action and Perception*; in Artificial Intelligence, 167(1-2), pp.170–205, 2005.

Michael Levit Michael Levit received his M.S. and Ph.D. degrees from the University of Erlangen, Germany in 2000 and 2005 respectively. He spent two years working at AT&T Laboratories where he was actively engaged in the *"How May I Help You?"* project and has authored several scientific publications in the fields of spoken language recognition and understanding. During 2005 he held position of a postdoctoral associate at MIT Media Laboratory in Cambridge, Massachusetts. Since 2006 he is with BBN Technologies and is currently with ICSI working on automatic question answering in multilingual environments.

Deb Roy Deb Roy is Associate Professor of Media Arts and Sciences at the Massachusetts Institute of Technology, and Director of the Cognitive Machines Group at the MIT Media Laboratory. He has published in the areas of knowledge representation, speech and language processing, language acquisition, robotics, information retrieval, cognitive modeling, and human-machine interaction. He has served as guest editor for the journal Artificial Intelligence, and is an associate of the journal Behavioral and Brain Sciences. He holds a BASc in computer engineering from University of Waterloo, Canada, and MS and PhD degrees in Media Arts and Sciences from MIT.