Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions *

Matt MacMahon
Dept. of Electrical and Computer Engineering
adastra@mail.utexas.edu

Brian Stankiewicz
Department of Psychology
bstankie@psy.utexas.edu

Benjamin Kuipers
Department of Computer Sciences
kuipers@cs.utexas.edu

The University of Texas at Austin
1 University Station
Austin, Texas 78712 USA

Abstract

Following verbal route instructions requires knowledge of language, space, action and perception. We present MARCO, an agent that follows free-form, natural language route instructions by representing and executing a sequence of *compound action specifications* that model which actions to take under which conditions. MARCO infers implicit actions from knowledge of both linguistic conditional phrases and from spatial action and local configurations. Thus, MARCO performs explicit actions, implicit actions necessary to achieve the stated conditions, and exploratory actions to learn about the world.

We gathered a corpus of 786 route instructions from six people in three large-scale virtual indoor environments. Thirty-six other people followed these instructions and rated them for quality. These human participants finished at the intended destination on 69% of the trials. MARCO followed the same instructions in the same environments, with a success rate of 61%. We measured the efficacy of action inference with MARCO variants lacking action inference: executing only explicit actions, MARCO succeeded on just 28% of the trials. For this task, inferring implicit actions is essential to follow poor instructions, but is also crucial for many highly-rated route instructions.

Introduction

Imagine you have an appointment in a large building you do not know. Your host sent instructions describing how to reach her office. You pull out the paper, read through and interpret the text, and proceed down corridors, taking the necessary actions. Upon finishing the instructions, you come to an unmarked, closed door. Is your appointment behind this door? Though the instructions were fairly clear, in a few places, such as the end, you had to infer what to do. How does an agent interpret an under-specified instruction text in the environment to infer the correct course of action?

Verbal *route instructions* are explanations given by a *director*, intended to guide a mobile agent, the *follower*, toward a specific spatial destination. When following route instructions, the follower must parse and interpret the text, model the instruction's actions and descriptions, and enact the instructions in the world, by performing these actions and recognizing the descriptions.

Typically, a follower cannot simply execute instructions without inference, since the necessary actions are not completely specified. Instructions often provide just a skeletal plan of action (Agre & Chapman 1990). A follower can resolve the ambiguities and omissions by using knowledge of language, an understanding of spatial actions and relations, and a model of the environment.

This paper presents a system that interprets human-written route instructions and follows the inferred model of the described route. Our approach builds on a rich literature studying different aspects of route instructions. Some work presents a model of route instructions, but does not apply the model to navigate (Vanetti & Allen 1988; Daniel *et al.* 2003; Tversky & Lee 1999; Klippel *et al.* 2005; Anderson *et al.* 1991). Other work concentrates on understanding single spatial commands in the small-scale space of a room (Skubic *et al.* 2004) or tabletop (Roy 2005). Finally, other work follows instruction sequences in a large-scale space, but does not use spatial and linguistic knowledge to recover from instruction errors or to infer implicit actions (Bugmann *et al.* 2004; Simmons *et al.* 2003).

Inferring and executing implicit actions from route instructions requires knowledge of both language and spatial actions. Some implicit actions are explicitly stated as conditions to achieve (e.g. "`With the wall on your left, walk forward,`") while other actions are implicit as preconditions (e.g. "`Go two intersections down the pink hallway`"). Some unstated actions are necessary to match the description to the environment. When told to "`Walk to the further end of the hall,`" a follower must turn to see the hall in both directions, estimate which end is most distant, possibly turn again to face the longer end, and only then move forward.

The core measure of a set of instructions for a route is simple – did the follower end up at the intended destination? Likewise, the minimal measure of an instruction follower is simple – how often does the follower successfully complete instructions? To evaluate MARCO and human followers, we use a large corpus of natural language route instructions,

*This work was supported by AFOSR grants FA9550-04-1-0236, FA9550-05-1-0321 and NIH grant EY016089 to BJS, by NSF grant IIS-0413257 to BJK, and by support for MM under ONR work order N0001405WX30001 for the NRL Research Option, Coordinated Teams of Autonomous Systems.

written by people over a variety of routes. The agent follows the routes by navigating through complex, large-scale environments without any prior spatial knowledge of the environment's layout.

This paper introduces the MARCO architecture for understanding and executing natural language route instructions. We measured how often MARCO reaches the destination of route instructions written by people for other people in large-scale, indoor, virtual environments. We compared MARCO's performance with people's performance following the same instruction texts in the same layouts. To better understand MARCO's performance and the behavior of human directors and followers, we compared the performance of MARCO with and without the ability to infer implicit actions. Running MARCO without action inference provides a measure of how often spatial and linguistic inference are necessary to follow route instructions successfully.

MARCO Architecture

MARCO is composed of six primary modules: three to interpret the route instruction text linguistically and three to interpret the instructions spatially in the context of the environment. The MARCO architecture for understanding and following natural language route instructions builds on ideas from the Nautilus natural language understanding system and the GRACE system (Simmons *et al.* 2003).

The linguistic stack parses and models raw text. The *syntax parser* models the surface structure of an utterance. The *content framer* interprets the surface meaning of the utterance. The *instruction modeler* applies spatial and linguistic knowledge to combine information across phrases and sentences. Figure 1 shows the representations MARCO uses to model route instructions.

The *executor* reactively interleaves action and perception, acting to gain knowledge of the environment and execute the instructions in the context of this spatial model. The *robot controller* is an interface to the particular follower's motor and sensory capabilities. The *view description matcher* checks symbolic view descriptions against sensory observations and world models, checking the expected model against the observed model.

Modeling natural language route instructions

The *syntax parser* parses the raw route instruction text. Our implementation uses a probabilistic context-free grammar built with the Python Natural Language Toolkit (Bird & Loper 2004). Instead of modeling part-of-speech syntax, our grammar directly models verb-argument structure, similarly to (Bindiganavale *et al.* 2000; Chang, Narayanan, & Petruck 2002). An example parse tree is at the top of Figure 1. We used the parser to help annotate of the treebank of parses for the corpus, but do not test the parser in the evaluation below.

The *content framer* translates the surface structure of an utterance to a model of the surface meaning as a nested attribute-value matrix. The matrix representation makes the content readily accessible. The resulting *content frame* (see middle of Figure 1) models the nested structure and sense of an utterance by dropping punctuation, arbitrary text ordering, inflectional suffixes, and spelling variations. The

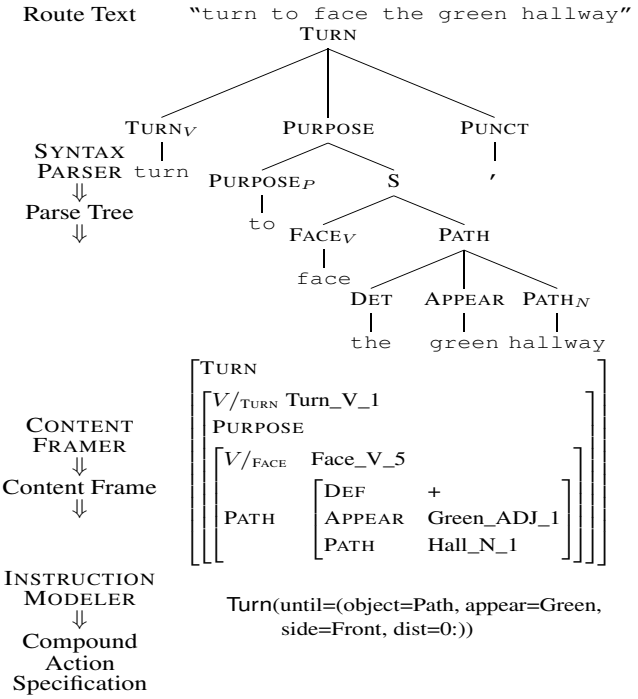


Figure 1: MARCO linguistic modules modeling a route instruction text (Top) through the syntactic verb argument and phrase structure (Mid-Top), the surface semantics frame (Mid-Bottom), and the imperative semantics of which action to take under a minimal model of the context (Bottom).

content framer draws word senses from WordNet (Fellbaum 1998), an ontology for English.

The *instruction modeler* translates the *content frame*'s representation of the surface meaning of an instruction element to an imperative model of what to do under which conditions – the *compound action specification*. The instruction modeler infers the imperative model from the instructions by applying linguistic knowledge of the verbs and prepositions of the route instructions and spatial knowledge of how perception and action depend on the local spatial configuration in similar environments.

Representing expected views and actions

The follower needs to model the actions and observations described in route instructions. However, instructions rarely specify exactly what the follower will see, but rather describe some distinctive attributes of some of the scenes along the route. The follower takes actions depending on how its observations, while navigating, match its expectations from the route instructions.

A *view description* represents what the follower expects at a pose in the environment, given the descriptions in the instructions. For each expected object, the view description models the object's type, the object's location within the view relative to the observer (angle and distance), and any description of the object's appearance and other attributes. The view description is a minimal model of what the fol-

lower expects: it neither over-commits to unspecified details nor enumerates possible worlds. Instead, it models no more than what was said. For example, the *until* condition at bottom of Figure 1 models the post-condition of the Turn: the follower expects a *Path* with a *Green* appearance in front of it, but it may be immediate in the view or off in the distance.

Route instructions require at least four low-level *simple actions*. Turn changes the agent's orientation (pose) while remaining in the same location. Travel changes the agent's location without changing orientation along a path. Verify checks an observation against a description of an expected view. Declare-goal terminates instruction following by declaring the agent is at the destination. Route instructions may contain other action types, such as "`open the door`" or "`take the elevator to the 2nd floor`". However, these four simple actions are both necessary to follow almost all route instructions and sufficient for many route instructions.

The *compound action specification* captures the commands in route instructions by modeling which simple actions to take (i.e., Turn, Travel, Verify and Declare-goal) under which perceptual (e.g. seeing a view) or cognitive conditions (e.g. estimating a distance). Resolving some ambiguities is deferred until the follower observes the environmental context as it proceeds along the route. These compound action specifications are similar to the "minimal units of information" (Daniel *et al.* 2003) or Higher-Order Route Instruction Elements (Klippel *et al.* 2005). Figure 1 shows the transformation from text to the imperative instruction model.

Each clause is interpreted as a compound action specification depending on the verb or a heuristic match based on the other constituents. Adverbs, verb objects, and prepositional phrases translate to pre-conditions, while-conditions, and post-conditions in compound action specifications. For instance, constituents may describe which path to take, how far to travel, or the view that will be seen. This is similar in intent to work on combining the lexical semantics resource FrameNet with action schemas, allowing inference (Chang, Narayanan, & Petruck 2002). The modeler also recognizes termination conditions stated as purpose clauses (Di Eugenio 1992), like "`Turn so that you see a chair in front.`"

Interleaving Action, Perception, and Modeling

The *executor* sequences simple actions given the environmental context and the state of following the route instructions. For example, given a "`face`" command, the executor Turns until a Verify signals that the observations have matched the view description. Currently, MARCO uses a simple executor that attempts to execute each compound action specifications fully before moving to the next. This algorithm may be replaced with a full action sequencer (e.g. RAPs as by (Bonnasso *et al.* 1997) or TDL as by (Simmons *et al.* 2003)) or an algorithm reasoning on inferred route topology (Kuipers *et al.* 2004).

The *robot controller* executes the Turn, Travel, Verify, and Declare-Goal actions. Robot controllers present a common interface to the executor, abstracting domain-dependent control implementation to simple actions.

The *view description matcher* checks the symbolic view descriptions against sensory observations. The view description matcher treats the view description as constraints that the observation stream must meet. This defers handling many forms of ambiguity until the environment can provide some disambiguating context. For instance, given the instruction "`Turn to face the blue path`," the view description would be Path(distance=*'0:'*, side=*Front*, appear=*blue*). The colon indicates an unbounded distance in the view. The blue path may run forward from the agent (distance=*'0'*, side=*Front*) or may be visible crossing this path in the distance (distance=*'1:'*, side=*Sides*). MARCO checks for both cases while turning.

The view description matcher will use whatever perceptual abilities the robot has available. On a hardware robot, the concept of an intersection can be linked to the code that segments intersections in the laser scan and classifies the local path topology (e.g. as a dead end, "T", or corner intersections (Kuipers *et al.* 2004)). With the simulation in this paper, MARCO cannot directly observe intersection type, but must model it through the relative positions of the observed paths (see Figure 3).

Robustness to errors and ambiguities

When MARCO comes across a word that it does not have in its concept base, it searches for the nearest known synonym or more abstract hypernym using the WordNet ontology. For instance, when instructed to "`face the futon,`" MARCO will discover *futon* is not in its concept base, look it up in WordNet, find the broader concept of *couch* in its concept base, and stop turning when the view description matcher observes a *couch*.

MARCO is also robust to unexpected input. If the content framer encounters an constituents that it cannot model, it will ignore it while modeling the remainder of the clause. Likewise, if the parser cannot parse one sentence from a set of route instructions, it will parse the others. These techniques work well for two reasons. First, route instructions often contain a lot of redundant information, so neglecting to understand a phrase in one sentence is often not critical. Second, the essential information in route instructions is usually stated using a relatively small variety of content frames for directing movements. Most of the novel sentence frames occur in the declarative descriptions between movement commands, so understanding these is often not necessary if the imperative sentences are correct, complete, properly understood, and properly applied.

Inferring actions implicit in instructions

The instruction modeler recognizes some linguistic conditional clauses (e.g., "`when`," "`at`," and "`so that`"). These conditionals are modeled as possibly requiring an action to achieve. For instance, "`At the corner, turn left`" is modeled as Turn(direction=*Left*, precondition =Travel(until=Corner(*dist*='0'))).

Implicit actions are inferred using both linguistic and spatial knowledge and reasoning. For instance, reading "`Go down the hall to the chair,`" the language model interprets the phrase structure as *along* and *un-*

"Take the blue path to the chair."

Travel(along =Path(*appear* =Blue, *side* =Front),
until =Corner(*distance* ='0', *side* =At))

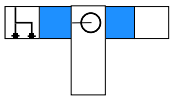
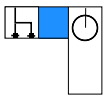
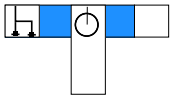
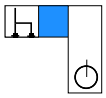
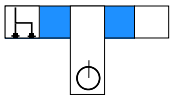
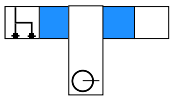
	Map, Robot Pose	Implicit Actions	Worst-Case Actions Taken
a			Travel
b		Precond Turn	Turn Travel
c		Explore Turn Precond Turn	Turn Turn Turn Travel
d		Precond Travel Precond Turn	Travel Turn Travel
e		Precond Travel Explore Turn Precond Turn	Travel Turn Turn Turn Travel
f		Explore Turn Precond Travel Explore Turn Precond Turn	Turn Travel Turn Turn Turn Travel

Figure 2: These simple scenarios illustrate how interpreting an utterance depends on the follower's pose in the environment and its cognitive map. The circle represents the follower, with the line indicating its front. The follower can see hallways to its side, but not down the side hallway.

til parameters of a Travel action. Using spatial knowledge and the Travel action model, MARCO infers the conditions of the Travel action:

Pre The path should be immediately in front and the chair should be in the front in the distance.

Post The chair will be local to the agent.

Though the executor primarily performs the actions explicitly stated in the route instructions, the executor also plans sequences to gain information and to achieve pre- and post-conditions of actions. Exploratory actions may be necessary to determine where a reference object is: e.g. in "Go towards the chair", the follower may Turn to locate the chair. If the pre- and post-conditions of actions are not met, the executor plans to achieve them. The actions the follower takes depend on both the route instruction text and the text's correspondence to the environment.

Figure 2 shows how this instruction is applied to navigate given different maps and starting poses. Figure 2(a) shows the default assumption, that the previous instruction elements have moved the follower into position. If a blue path is immediately in front of the agent, it will execute the explicit Travel action. In Figure 2(b), the blue path is visible immediately to one side, so it will Turn to meet the precondition of Travel along a path, though this action is not stated in the instructions. In Figure 2(c), the blue path is visible to both sides, but the follower does not know which way the chair is. The follower must make an exploratory Turn to look down the blue hall in one direction, then if it does not see the chair, Turn around to face the chair.

If the follower does not see a blue path in its immediate surround, but does see one off in the distance (Figure 2(d,e)), it will Travel to the distant path, then Turn onto it before proceeding. Figure 2(f) shows the agent making an exploratory Turn to find the blue hall, a Travel to reach it, another exploratory Turn to find the chair, and only then the explicit Travel command. If it does not see a blue path from any pose at its current location, it will move through the environment until it finds a match. This search behavior improves performance on poor instructions, while not significantly reducing the success rate of highly-rated instructions (MacMahon & Stankiewicz 2006).

Evaluation

Route instructions represent knowledge about spatial actions and spatial layouts. A route instruction set is useful if it reliably guides followers to the intended destination. Conversely, the navigation and understanding skills of a follower mediate how well route instructions are followed.

Heuristic linguistic or spatial methods can suggest that the syntax, the semantics, and even the pragmatics of a route instruction text are incoherent or potentially inadequate (Riesbeck 1980). However, without situating the route instructions in a spatial environment, these methods cannot determine if missing information is necessary or extraneous. Other knowledge of the structure of the environment may constrain the possible actions or resolve linguistic ambiguity. Additionally, as seen in Figure 2, even apparently complete instructions may still require thought and action to understand how they correspond to the environment. Finally, heuristic methods cannot catch explicit mistakes in description.

Human Route Instruction Directing and Following

We evaluated MARCO in three environments with a large corpus of route instructions written by six human directors. The corpus consists of 786 natural language route instruction texts from 6 subjects (3 M, 3 F) in three virtual reality environments. 36 subjects (21 M, 15 F) followed these route instructions. For details of the human study procedure, see (MacMahon & Stankiewicz 2006).

Using desktop virtual reality environments had several benefits: (1) all route directors had similar exposure to the environments; (2) all pertinent aspects of the environments were known and repeatable across subjects; (3) directors learn the environment by exploring from the same first-

Figure 3: Human Participants' first-person view from pose of the simulated robot (blue circle) at the easel ('E') in the map. MARCO experienced the view as the text token list:

[(Cement, Easel, Cement, Butterfly, Wood, Butterfly),
(Wall, Empty, Wall, Butterfly, Wood, Butterfly),
(Cement, Empty, Wall, End, Wall, End)]

person perspective as the followers; and (4) MARCO can navigate the same environments as people.

Each of the three large-scale spaces used had forty locations, seven long paths with distinct textured flooring, seven to twelve short paths with a common cement floor, and numerous visual and structural features. Each environment contained seven named locations that were the start and end points of the routes that the directors were asked to describe. The layouts are difficult for people to learn and navigate, so they provide challenges for both the directors and followers. Figure 3 shows an example human view of the environment (Top) and the textual view of the simulator MARCO sees (Bottom). Figure 4 shows the overhead layout map (not seen by participants) of this environment, with the follower's movement trace marked.

The director's task in each environment is split into three phases. In the first phase, a director freely explores the environment. Second, the director is quizzed for navigation competency in the environment. Once able to pass the competency test by navigating efficiently among the named locations, the director is queried for directions between all pairs of named places in the environment. For each route, the director types a set of instructions, then navigates to the goal, and then self-rates his(her) belief that (s)he has reached the goal and the quality of his(her) own instructions.

To gauge the quality of the route instructions, another group of people evaluated the route instructions. Thirty-six participants (15 female, 21 male) read the route instructions and attempted to follow the routes described in the virtual environments. While navigating, the follower could re-examine the route instructions by pressing a key, which covered the navigation screen with a pop-up window showing the instruction text. Each route instruction text was evaluated independently by six people. The destination positions were not marked in the environments; the followers had to explicitly end the navigation and indicate whether they believed they had reached the described goal.

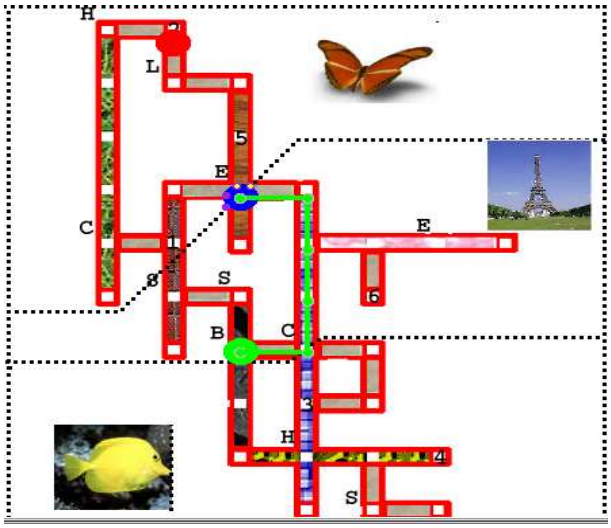


Figure 4: **Bottom**: Map of one of three virtual environments (not seen by participants). Three regions share a wall hanging of a fish, butterfly, or Eiffel Tower. Each long hallway has a unique flooring. Letters above mark objects (e.g. 'C' is a chair), numbers indicate named positions.

Route Instruction Corpus Statistics

For some routes, the director either did not enter any text or only entered a comment, e.g. "I don't know." For this evaluation of MARCO, we omit training routes, duplicated routes, and the empty route descriptions, leaving 682 route instruction texts that MARCO and people followed. The route instructions had a mean of 34.5 words from a lexicon of 587 words and, as modeled, had means of 4.7 context frames and 5.1 compound action specifications.

The six directors in this study vary significantly in writing style as a group and across different route instructions. Across directors, style varies significantly in length of the instructions (m=36.4, sd=16.5 words), size of the lexicon used (m=213, sd=55 words), number of frames used (m=5.0, sd=2.0 frames), efficiency of the routes (m=55, sd=21 percentage points), human success rate (m=63, sd=19 percentage points), and human subjective rating (m=4.0, sd=1.0 of 1–6 scale).

Route Instruction Situated Testbed

To test how well an agent (either human or MARCO) follows route instructions, we gave the agent a route instruction text, placed it at the starting location, monitored how it navigates through the environment, and observed whether it reaches and identifies the destination. We performed this experiment with people navigating computer-rendered VRML models of the three indoor environments. We provided the same instruction texts to a software agent, MARCO, which navigated through symbolic representations of the same environments. MARCO's input was from the hand-verified 'gold-standard' parse treebank, not the parser, but all other modeling was done autonomously.

In these experiments, MARCO perceives the world as an

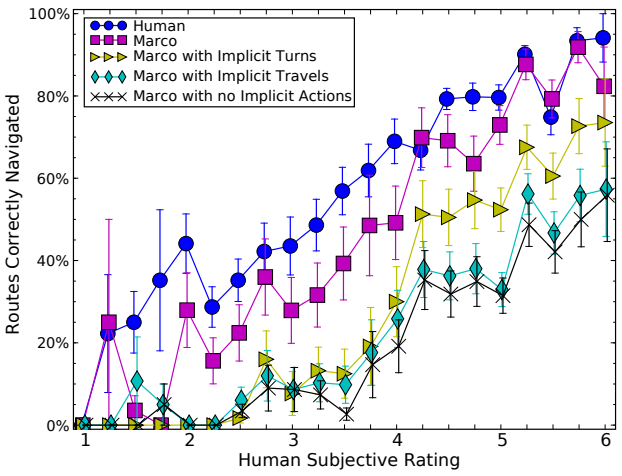


Figure 5: Human and MARCO success rates, with standard error bars, versus human instruction rating. The rating of 1 indicates extremely poor instructions, while 6 is excellent. Success rate is how often, on average, the human followers (circles) and MARCO (squares) finished navigating at the intended position for all the instruction texts with mean human rating of $r \pm 0.125$. Data as of April 21, 2006.

ordered list of symbols corresponding to any visible walls, pictures, furniture, and path segments. However, MARCO must model the world to fill the significant disconnect between the symbolic observations and the concepts mentioned in the instructions. For instance, MARCO must match a described `corner` with a model with spatial knowledge of a corner as the termination of two paths in intersection. Additionally, the instructions contain compositional restrictions, e.g. "`the intersection with the chair where the flowered hallway goes to the left.`"

We tested the full MARCO model and MARCO variants missing spatial and linguistic action inference abilities by running each agent against a large corpus of instructions. The test set consisted of the 682 instructions with some descriptive text, which were followed by the human participants in the three full-sized virtual environments. Comparing the performance of MARCO with people following the same route instructions tells us how well MARCO is performing on a wide variety of route instructions. Comparing the performance of an ablated MARCO model against the full MARCO model shows how much impact action inference has on navigation success, given any redundant information in the environment, in the instructions, or from the follower's spatial reasoning.

Full-Corpus Implicit Action Inference Experiment

We present results for five types of followers: (1) human participants, (2) the full MARCO model, (3) MARCO without Turn inference, (4) MARCO without Travel inference, and (5) MARCO without either Turn or Travel inference. For people, the results are the mean over runs from 6 participants following each instruction set, each beginning at the start location facing a random direction. For the MARCO cases, the presented results are the mean over four runs, facing each of the four directions at the start.

Figure 5 shows the evaluation results. Human participants were able to successfully find and identify the desired destination with an overall mean success rate of 69% of 682 instruction sets in the three environments. With full action inference, MARCO successfully followed 61% of the route instruction texts. Further, MARCO increases in performance as the human instruction rating increases and as human performance increases. While MARCO does not yet match human performance across route instructions of all qualities, the correlations from MARCO's performance to human performance and to human ratings are strong (MacMahon & Stankiewicz 2006).

Without inferring Travel actions, MARCO's performance drops to 42%. Some implicit Travels are stated in the text as preconditions, for instance, "`At the end of the hall, turn right.`" Others are implicit in the preconditions of the stated action, for instance, "`Take the green path,`" when the green path is distant. Finally, some actions may be implicit in how the command is expressed. "`Take the second left,`" implies Travel forward two intersections with a path to the left.

If MARCO does not execute implicit Turn actions, performance slips to 32% of the instruction corpus. One type of implicit Turn is in the text as a condition to achieve instead of an explicit command: "`With your back to the wall, walk forward`" implies a possible Turn if a wall is not immediately to the rear. Turns can also be implicit in an action's preconditions, e.g. a Turn to face the path in "`Go down the brick hallway.`" Implicit actions may be unnecessary, depending on the starting conditions and how MARCO interpreted and executed any previous instruction elements.

Following purely explicit instructions, without inferring either Turn or Travel actions, MARCO can successfully follow just 28% of the routes in the corpus. The effects of Turn or Travel action inference are neither fully independent nor fully dependent, both are critical for route instructions.

Implicit Action Experiment Results by Rating

Action inference is essential for following the lowest rated instructions in this corpus, but merely important for following the highest rated instructions. Table 1 summarizes the results graphed in Figure 5 across broad classes of human *post-hoc* subject instruction ratings. In this discussion, r will denote the mean rating on an instruction set from the six human followers.

For poor instructions, $r \leq 3.5$ out of 6, MARCO is effectively crippled without action inference skills. On good but not excellent instruction, MARCO can follow a significant number of instructions without action inference, but performs much better by inferring actions, especially Turns. Making an implicit turn puts the follower on the correct path, revealing a view down the path which shares very little information with views facing in other directions. A Travel

Quality Range	All	1:2.5	2.5:3.5	3.5:5	5:6
Human	69%	33%	46%	75%	85%
Full MARCO	61%	16%	36%	64%	84%
No implicit Travel	42%	0%	12%	46%	66%
No implicit Turn	32%	2%	11%	32%	52%
No implicit acts	28%	1%	7%	30%	46%

Table 1: Performance on instructions with mean human rating r, s.t. $x < r \leq y$. All differences from Full MARCO are significant at $p < 0.001$, except there is no significant difference between people and Full MARCO for the instructions rated $5 < r \leq 6$.

moves the agent to a new place, but does not bring as much new information into the view for agents able to see distant objects.

On the best instructions, those rated $r > 5$, the full MARCO system and people had no significant difference in performance. Without action inference, MARCO had significant decreases in performance even on the best sets. Action inference accounts for nearly all the success on poor instructions and about half the success on good instructions.

Instruction-Based Learning (IBL) Comparison

This work is similar in intention to the Instruction-Based Learning (IBL) for Mobile Robots project (Bugmann *et al.* 2001; 2004). (Bugmann *et al.* 2001) presented a corpus of 96 spoken route instruction sets from participants guiding a human operator, who had remote control of a robot navigating through a tabletop model of a town center. They modeled the instructions as action schemas, called "functional primitives," such as MOVE FORWARD UNTIL ¡COND¿, TURN ¡DIR¿ ¡LOC¿, ¡LANDMARK¿ IS LOCATED ¡WHERE¿, and GO TO ¡LANDMARK¿.

(Bugmann *et al.* 2004) implemented a robotic system capable of following programs of functional primitives from this corpus, expanded to 144 route instructions. The 15 functional primitives take a fixed parameter list, so their action model is less expressive than our Compound Action Specifications. Effectively, their functional primitive are curried versions of our actions with some parameters fixed, matching common sentence argument structures rather than allowing any combination of conditions. For instance, they model go_until, exit_roundabout, follow_road_until, and take_road, all of which would be modeled with our Travel action with various keyword parameters.

(Bugmann *et al.* 2004) also compared human performance with the performance for a robot navigating through the tabletop model environment given hand-translated or automatically-translated programs of functional primitives from the corpus. People were able to reach the destination on 83% of the instructions, the robot followed hand-translated programs on 63% of the routes, and 14% of the routes automatically translated into programs "would actually lead the robot to the goal."

Though our success rates are not directly comparable, since they start with raw speech and control a physical robot, our automated success rates are much more similar to our human rates. Their environment had fewer places, paths, and strong visual features than ours, but had more diverse intersections in a realistic town street layout. Their basic instruction-following method is similar to our work, but seems less robust to errors and omissions in the instructions, due to the spatial and linguistic knowledge we model.

The work in this paper is more easily and less expensively replicated, since no special robotic equipment or physical town model is needed. More importantly, our subjects learned the environments from the same first-person perspective as the human and software agents following the instructions and wrote instructions from memory. Bugmann's participants only saw an outside, panoramic perspective of the town model while directing. This difference in how environments are learned and perceived between the directors and followers leads to a class of errors not present in our approach. Specifically, directors may refer to information unavailable to followers. Conversely, while our directors may make errors while learning the map through navigation or recalling the map while directing, these errors are cognitively interesting and prevalent in the real world.

Conclusions

This paper examines the role of using knowledge about language and space to infer implicit actions in following natural language route instructions through large-scale spaces. The MARCO agent can parse, model, and reactively enact route instructions. MARCO approaches human levels of performance in applying instruction texts to navigate from a starting place to the destination and declare when the goal is reached. Our evaluation testbed ties together a large instruction corpus, navigable environments, and human and artificial embedded agents with linguistic and spatial reasoning abilities. Comparing the performance of MARCO model variants, we find implicit actions are essential to following poorly-rated instructions and are often important to following even highly-rated instructions.

This testbed of a large route instruction text corpus tied to simulated environments presents a challenge task for researchers in natural language understanding and spatial reasoning. The methodology emphasizes understanding the gist of route instructions over some details: the essential linguistic and spatial details separate navigation success from failure. However, to be tested, components must be integrated into a complete agent that can read the instructions and apply the understanding to act in the world.

This paper contributes an assessment of human performance for communicating route information through unfamiliar large-scale spaces. By comparing the performance of a computational model with and without the ability to infer implicit actions, we measure how often understanding the unstated is necessary to succeed in this task. Though this ratio will change for other tasks and domains, the methodology of comparing human and automated systems on large corpora of problems will generalize.

Successfully following natural language route instructions requires both linguistic and spatial reasoning skills. Linguistic syntax parsing and surface semantics are not sufficient; a system must be able to ground semantic concepts in

actions and observations. Moreover, the system must be able to apply pragmatic reasoning skills to infer the director's intentions of *what* to do and *where* to go. The follower should move to a pose matching the precondition of the next instruction, even when the text does not state the step. In fact, inferred actions may violate or override explicitly stated actions, such as when the instructions lead the follower to face a dead end and indicate forward travel.

The ability to follow route instructions is useful: every day, people use route instructions to travel along previously unknown routes and there is a large industry devoted to generating driving instructions. Spatial route instructions are an interesting combination of robotics, artificial intelligence, cognitive psychology, and natural language processing. Route instructions are easily evaluated, despite the complexity of integrating modules doing linguistic modeling, abstract spatial reasoning, and moving a robot through a world – does the follower reach the destination?

We have demonstrated how linguistic and spatial knowledge, along with exploratory action in the environment, are jointly necessary for successful applying route instructions. We believe that the natural language understanding methods described here will generalize to the larger domain of understanding instructions about complex sequential tasks, including cooking, first aid, furniture assembly, automobile repair, and many others. We also believe these tasks should be similarly evaluated, with a testbed that demonstrates sufficient understanding by achieving a complex, situated task given diverse natural language instructions.

References

Agre, P. E., and Chapman, D. 1990. What are plans for? *Robotics & Autonomous Systems* 6:17–34.

Anderson, A.; Bader, M.; Bard, E. G.; Boyle, E.; Doherty, G.; Garrod, S.; Isard, S.; Kowtko, J.; McAllister, J.; Miller, J.; Sotillo, C.; Thompson, H. S.; and Weiniert, R. 1991. The HCRC map task corpus. *Language and Speech* 34(4):351–366.

Bindiganavale, R.; Schuler, W.; Allbeck, J. M.; Badler, N. I.; Joshi, A. K.; and Palmer, M. 2000. Dynamically altering agent behaviors using natural language instructions. In *Proc. of 4th Intl. Conf. on Autonomous Agents*, 293–300.

Bird, S., and Loper, E. 2004. NLTK: The natural language toolkit. In *Proc. of 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*.

Bonnasso, R. P.; Firby, R. J.; Gat, E.; Kortenkamp, D.; Miller, D. P.; and Slack, M. G. 1997. Experiences with an archtitecture for intelligent, reactive agents. *J. of Experimental & Theoretical Artificial Intelligence* 9(1):237–256.

Bugmann, G.; Lauria, S.; Kyriacou, T.; Klein, E.; Bos, J.; and Coventry, K. 2001. Using verbal instructions for route learning : Instruction analysis. In *Proc. of Torwards Intelligent Mobile Robots Conf.* .

Bugmann, G.; Klein, E.; Lauria, S.; and Kyriacou, T. 2004. Corpus-based robotics : A route instruction example. In *Proc. of Intelligent Autonomous System*, 96–103.

Chang, N.; Narayanan, S.; and Petruck, M. R. 2002. Putting frames in perspective. In *Proc. of 19th Intl. Conf. on Computational Linguistics (COLING-02)*.

Daniel, M.-P.; Tom, A.; Manghi, E.; and Denis, M. 2003. Testing the value of route directions through navigational performance. *Spatial Cognition and Computation* 3(4):269–289.

Di Eugenio, B. 1992. Understanding natural language instructions : the case of purpose clauses. In *Proc. of 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*, 120–127.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Freksa, C., and Mark, D. M., eds. 1999. *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science (COSIT '99)*. Stade, Germany: Springer.

Klippel, A.; Tappe, H.; Kulik, L.; and Lee, P. U. 2005. Wayfinding choremes - a language for modeling conceptual route knowledge. *J. of Visual Languages & Computing* 16(4):311–329.

Kuipers, B.; Modayil, J.; Beeson, P.; MacMahon, M.; and Savelli, F. 2004. Local metrical and global topological maps in the hybrid Spatial Semantic Hierarchy. In *Proc. of IEEE Intl. Conf. on Robotics & Automation (ICRA-04)*.

Lovelace, K. L.; Hegarty, M.; and Montello, D. R. 1999. Elements of good route directions in familiar and unfamiliar environments. In Freksa and Mark (1999), 56–82.

MacMahon, M., and Stankiewicz, B. 2006. Human and automated indoor route instruction following. In *Proc. of 28th Annual Meeting of the Cognitive Science Society*.

Riesbeck, C. 1980. "You can't miss it!" : Judging the clarity of directions. *Cognitive Science* 4:285–303.

Roy, D. 2005. Semiotic schemas: a framework for grounding language in action and perception. *Artificial Intelligence* 167(1–2):170–205.

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. 2003. GRACE: An autonomous robot for the AAAI Robot Challenge. *AI Magazine* 24(2):51–72.

Skubic, M.; Perzanowski, D.; Blisard, S.; Schultz, A.; Adams, W.; Bugajska, M.; and Brock, D. 2004. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man & Cybernetics – Part C* 34(2):154–167.

Tversky, B., and Lee, P. U. 1999. Pictorial and verbal tools for conveying routes. In Freksa and Mark (1999), 51–64.

Vanetti, E. J., and Allen, G. L. 1988. Communicating environmental knowledge : The impact of verbal and spatial abilities on the production and comprehension of route directions. *Environment & Behavior* 20:667–682.