INTEGRATING CONTEXT INTO INTENT RECOGNITION SYSTEMS

Richard Kelley, Christopher King, Amol Ambardekar, Monica Nicolescu, Mircea Nicolescu
Department of Computer Science and Engineering, University of Nevada, Reno, 1664 N. Virginia St., Reno, U.S.A.

Alireza Tavakkoli
Department of Computer Science, University of Houston-Victoria, 3007 N. Ben Wilson, Victoria, U.S.A.

Keywords: Intent recognition, Human-robot interaction, Natural language processing.

Abstract: A precursor to social interaction is social understanding. Every day, humans observe each other and on the basis of their observations "read people's minds," correctly inferring the goals and intentions of others. Moreover, this ability is regarded not as remarkable, but as entirely ordinary and effortless. If we hope to build robots that are similarly capable of successfully interacting with people in a social setting, we must endow our robots with an ability to understand humans' intentions. In this paper, we propose a system aimed at developing those abilities in a way that exploits both an understanding of actions and the context within which those actions occur.

1 INTRODUCTION

A precursor to social interaction is social understanding. Every day, humans observe each other and on the basis of their observations "read people's minds," correctly inferring the goals and intentions of others. Moreover, this ability is regarded not as remarkable, but as entirely ordinary and effortless. If we hope to build robots that are similarly capable of successfully interacting with people in a social setting, we must endow our robots with an ability to understand humans' intentions. In this paper, we propose a system aimed at developing those abilities in a way that exploits both an understanding of actions and the context within which those actions occur.

2 RELATED WORK

Whenever one wants to perform statistical classification in a system that is evolving over time, hidden Markov models may be appropriate (Duda et. al., 2000). Such models have been very successfully used in problems involving speech recognition (Rabiner, 1989). Recently, there has been some indication that hidden Markov models may be just as useful in modeling activities and intentions. For example, HMMs have been used by robots to classify a number of manipulation tasks (Pook and Ballard, 1993)(Hovland et. al., 1996)(Ogawara et. al., 2002). These approaches all have the crucial problem that they only allow the robot to detect that a goal has been achieved *after* the activity has been performed; to the extent that intent recognition is about prediction, these systems do not use HMMs in a way that facilitates the recognition of intentions. Moreover, there are reasons to believe (see below) that without considering the disambiguation component of intent recognition, there will be unavoidable limitations on a system, regardless of whether it uses HMMs or any other classification approach.

Extending upon this use of HMMs to recognize activities, previous work (of which (Tavakkoli et. al., 2007) is representative) has examined the use of HMMs to predict intentions. The work to date has focused on training a robot to use HMMs (via theory-of-mind inspired approaches), and on the best information to represent using those models. That work has mostly ignored questions such as scalability and the role that contextual information plays in the recognition process. The present work begins to address these issues, and introduces the idea of using a digraph-based language model to provide contextual knowledge.

Kelley R., King C., Ambardekar A., Nicolescu M., Nicolescu M. and Tavakkoli A. (2010).
INTEGRATING CONTEXT INTO INTENT RECOGNITION SYSTEMS.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 315-320
DOI: 10.5220/0002953903150320

To build that language model, we use the typed dependency extraction facilities of the Stanford Parser (Marneffe et. al., 2006). In contrast with constituency grammar, which views a sentence as being made up of phrases, dependency grammar represents grammatical links between pairs of words. *Typed* dependencies explicitly label the links between words with grammatical relations (Marneffe et. al., 2006). To the best of our knowledge, such a representation has not yet been used as the basis for any human-robot interaction work; our graph-based approach is also new.

3 LEXICAL DIGRAPHS

As mentioned above, our system relies on contextual information to perform intent recognition. While there are many sources of contextual information that may be useful to infer intentions, we chose to focus primarily on the information provided by object affordances, which indicate the actions that one can perform with an object. The problem, once this choice is made, is one of training and representation: given that we wish the system to infer intentions from contextual information provided by knowledge of object affordances, how do we learn and represent those affordances? We would like, for each object our system may encounter, to build a representation that contains the likelihood of all actions that can be performed on that object.

Although there are many possible approaches to constructing such a representation, we chose to use a representation that is based heavily on a graph-theoretic approach to natural language – in particular, English. Specifically, we construct a graph in which the vertices are words and a labeled, weighted edge exists between two vertices if and only if the words corresponding to the vertices exist in some kind of grammatical relationship. The label indicates the nature of the relationship, and the edge weight is proportional to the frequency with which the pair of words exists in that particular relationship. For example, we may have vertices *drink* and *water*, along with the edge $((drink, water), direct\_object, 4)$, indicating that the word "water" appears as a direct object of the verb "drink" four times in the experience of the system. From this graph, we compute probabilities that provide the necessary context to interpret an activity.

3.1 Dependency Parsing and Graph Representation

To obtain our pairwise relations between words, we use the Stanford labeled dependency parser. The parser takes as input a sentence and produces the set of all pairs of words that are grammatically related in the sentence, along with a label for each pair, as in the "water" example above.

Using the parser, we construct a graph $G = (V, E)$, where E is the set of all labeled pairs of words returned by the parser for all sentences, and each edge is given an integer weight equal to the number of times the edge appears in the text parsed by the system. V then consists of the words that appear in the corpus processed by the system.

3.2 Graph Construction and Complexity

3.2.1 Graph Construction

Given a labeled dependency parser and a set of documents, graph construction is straightforward. Briefly, the steps are

1. Tokenize each document into sentences.
2. For each sentence, build the dependency parse of the sentence.
3. Add each edge of the resulting parse to the graph.

Each of these steps may be performed automatically with reasonably good results, using well-known language processing algorithms. The end result is a graph as described above, which the system stores for later use.

One of the greatest strengths of the dependency-grammar approach is its space efficiency: the output of the parser is either a *tree* on the words of the input sentence, or a graph made of a tree plus a (small) constant number of additional edges. This means that the number of edges in our graph is a linear function of the number of nodes in the graph, which (assuming a bounded number of words per sentence in our corpus) is linear in the number of sentences the system processes. In our experience, the digraphs our system has produced have had statistics confirming this analysis, as can be seen by considering the graph used in our recognition experiments. For our corpus, we used two sources: first, the simplified-English Wikipedia, which contains many of the same articles as the standard Wikipedia, except with a smaller vocabulary and simpler grammatical structure, and second, a collection of childrens' stories about the objects in which

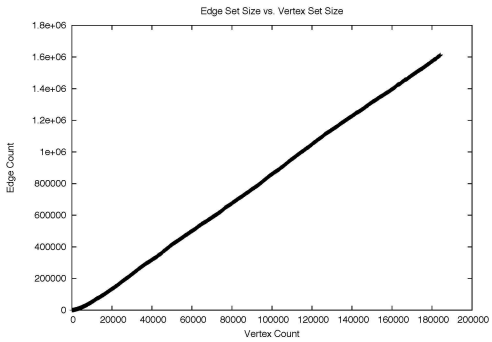

Figure 1: The number of edges in the Wikipedia graph as a function of the number of vertices during the process of graph growth.

we were interested. In Figure 1, we show the number of edges in the Wikipedia graph as a function of the number of vertices at various points during the growth of the graph. The scales on both axes are identical, and the graph shows that the number of edges for this graph does depend linearly on the number of vertices.

The final Wikipedia graph we used in our experiments consists of 244,267 vertices and 2,074,578 edges. The childrens' story graph is much smaller, being built from just a few hundred sentences: it consists of 1754 vertices and 3873 edges. This graph was built to fill in gaps in the information contained in the Wikipedia graph. The graphs were merged to create the final graph we used by taking the union of the vertex and edge sets of the graphs, adding the edge weights of any edges that appeared in both graphs.

4 VISION-BASED CAPABILITIES

In support of our intent recognition system, we require a number of visual capabilities for our robot. Among these, our system must be able to segment and track the motion of both humans and inanimate objects. Because we are interested in objects and their affordances, our system must also be able to visually identify objects and, for objects whose state can change over time, object states. Moreover, tracking should be done in three-dimensional space. To support this last requirement, we use a stereo-vision camera.

To perform segmentation and object recognition, we use a variant of maximally stable extremal regions for color images (Forssen, 2007). In our variant, we identify "strong" and "weak" edges in the image (based on our thresholding), and constrain the region-merging of color-based MSER so that region growth is inhibited across weak edges and prevented entirely across strong edges. This approach allows for increased stability, for multiple regions of different homogeneity to coexist near one another, and for more coherent segmentation of textured regions.

Having segmented a frame into regions, we perform object recognition using a mixture of Gaussians, computing probabilities at the *region* level rather than the pixel level. Because objects tend to consist of a smaller number of regions than pixels, this can lead to a substantial speedup.

Once we have segmented a frame and identified the regions of interest in that frame, we perform tracking via incremental support vector data descriptions and connected component analysis. We refer the interested reader to other, vision-specific work (Tavakkoli et. al., 2007).

5 INTENT RECOGNITION SYSTEM

5.1 Low-level Recognition via Hidden Markov Models

As mentioned above, our system uses HMMs to model activities that consist of a number of parts that have intentional significance. Recall that a hidden Markov model consists of a set of hidden states, a set of visible states, a probability distribution that describes the probability of transitioning from one hidden state to another, and a probability distribution that describes the probability of observing a particular visible state given that the model is in a particular hidden state. To apply HMMs, one must give an interpretation to both the hidden states and the visible states of the model, as well as an interpretation for the model as a whole. In our case, each model represents a single well-defined activity. The hidden states of represent the intentions underlying the parts of the activity, and the visible symbols represent changes in measurable parameters that are relevant to the activity. Notice in particular that our visible states correspond to dynamic properties of the activity, so that our system can perform recognition as the observed agents are interacting.

We train our HMMs by having our robot perform the activity that it later will recognize. As it performs the activity, it records the changes in the parameters of interest for the activity, and uses those to generate sequences of observable states representing the activity. These are then used with the Baum-Welch algorithm (Rabiner, 1989) to train the models, whose topologies have been determined by a human operator

in advance.

During recognition, the stationary robot observes a number of individuals interacting with one another and with stationary objects. It tracks those individuals using the visual capabilities described above, and takes the perspective of the agents it is observing. Based on its perspective-taking and its prior understanding of the activities it has been trained to understand, the robot infers the intention of each agent in the scene. It does this using maximum likelihood estimation, calculating the most probable intention given the observation sequence that it has recorded up to the current time for each pair of interacting agents.

5.2 Adding Context

Our system uses contextual information to infer intentions. This information is linguistic in nature, and in section 3 we show how lexical information representing objects and affordances can be learned and stored automatically. In this subsection, we outline how that lexical information can be converted to probabilities for use in intent recognition.

Context and Intentions. In general, the context for an activity may be any piece of information. For our work, we focused on two kinds of information: the location of the event being observed, and the identities of any objects being interacted with by an agent. Context of the first kind was useful for basic experiments testing the performance of our system against a system that uses no contextual information, but did not use lexical digraphs at all; contexts and intentions were defined by entirely by hand. Our other source of context, object identities, relied entirely on lexical digraphs. In experiments using this source of information, objects become the context and their affordances – represented by verbs in the digraph – become the intentions. As explained below, if s is an intention and c is a piece of contextual information, our system requires the probability $p(s \mid c)$, or in other words the probability of an affordance given an object identity. This is exactly what is provided by our digraphs. If "water" appears as a direct object of "drink" four times in the robot's linguistic experience, then we can obtain a proper probability of "drink" given "water" by dividing four by the sum of the weights of all edges which have "water" as the direct object of some word. In general, we may use this process to obtain a table of probabilities of affordances or intentions for every object in which our system might be interested, as long as the relevant words appear in the corpus. Note that this may be done without human intervention.

Inference Algorithm. Suppose that we have an activity model (*i.e.* an HMM) denoted by w. Let s denote an intention, let c denote a context, and let v denote a sequence of visible states from the activity model w. If we are given a context and a sequence of observation, we would like to find the intention that is maximally likely. Mathematically, we would like to find

$$\arg\max_s p(s \mid v, c),$$

where the probability structure is determined by the activity model w.

To find the correct s, we start by observing that by Bayes' rule we have

$$\max_s p(s \mid v, c) = \max_s \frac{p(v \mid s, c)p(s \mid c)}{p(v \mid c)}. \quad (1)$$

We can further simplify matters by noting that the denominator is independent of our choice of s. Moreover, we assume without loss of generality that the possible observable symbols are independent of the current context. Based on these observations, we can write

$$\max_s p(s \mid v, c) \approx \max_s p(v \mid s)p(s \mid c). \quad (2)$$

This approximation suggests an algorithm for determining the most likely intention given a series of observations and a context: for each possible intention s for which $p(s \mid c) > 0$, we compute the probability $p(v \mid s)p(s \mid c)$ and choose as our intention that s whose probability is greatest. The probability $p(s \mid c)$ is available, either by assumption or from our linguistic model, and if the HMM w represents the activity model associated with intention s, then we assume that $p(v \mid s) = p(v \mid w)$. This assumption may be made in the case of location-based context for simplicity, or in the case of object affordances because we focus on simple activities such as reaching, where the same HMM w is used for multiple intentions s. Of course a perfectly general system would have to choose an appropriate HMM dynamically given the context; we leave the task of designing such a system as future work for now, and focus on dynamically deciding on the context to use, based on the digraph information.

5.3 Intention-based Control

In robotics applications, simply determining an observed agent's intentions may not be enough. Once a robot knows what another's intentions are, the robot should be able to act on its knowledge to achieve a goal. With this in mind, we developed a simple method to allow a robot to dispatch a behavior based on its intent recognition capabilities. The robot first

infers the global intentions of all the agents it is tracking, and for the activity corresponding to the inferred global intention determines the most likely local intention. If the robot determines over multiple time steps that a certain local intention has the largest probability, it can dispatch a behavior in response to the situation it believes is taking place.

6 EXPERIMENTAL VALIDATION

We performed a series of experiments to test different aspects of our system. In particular, we wished to test the following claims: first, that contextual information could improve performance over a context-agnostic system; second, that intention-based control as described above can be used to solve basic, but still realistic, problems; and last, we wished to test lexical digraphs as a source of contextual information for inferring intentions related to objects and their affordances.

6.1 Setup

To validate our contextual approach, we performed a set of experiments using a Pioneer 3DX mobile robot, with an on-board computer, a laser rangefinder, and a stereo camera. We trained our robot to understand three basic activities: *following*, in which one agent trails behind another; *meeting*, in which two agents approach one another directly; and *passing*, in which two agents move past each other without otherwise directly interacting. We also built a model of *reaching* for use in the object-based tests.

We placed our trained robot in several indoor environments and had it observe the interactions of multiple human agents with each other, and with multiple static objects. In our experiments, we considered both the case where the robot acts as a passive observer and the case where the robot executes an action on the basis of the intentions it infers in the agents under its watch.

The first set of experiments was performed in a lobby, and had agents meeting each other and passing each other both with and without contextual information about which of these two activities is more likely in the context of the lobby. To the extent that meeting and passing appear to be similar, we would expect that the use of context would help to disambiguate the activities. Although contrived, these scenarios do facilitate direct comparison between a context-aware and a context-agnostic system.

To test our intention-based control, we set up two scenarios. In the first scenario (the "theft" scenario), a human enters his office carrying a bag. As he enters, he sets his bag down by the entrance. Another human enters the room, takes the bag and leaves. Our robot was set up to observe these actions and send a signal to a "patrol robot" in the hall that a theft had occurred. The patrol robot is then supposed to follow the thief as long as possible.

In the second scenario, our robot is waiting in the hall, and observes a human leaving the bag in the hallway. The robot is supposed to recognize this as a suspicious activity and follow the human who dropped the bag for as long as possible.

Lastly, to test the lexical-digraph-based system, we had the robot observe an individual as he performed a number of activities involving various objects. These included books, glasses of soda, computers, bags of candy, and a fire extinguisher.

Table 1: Quantitative evaluation - meet versus pass.

Scenario (with Context)	Correct Duration [%]
Meet (No context) - Agent 1	65.8
Meet (No context) - Agent 2	72.4
Meet (Context) - Agent 1	97.8
Meet (Context) - Agent 2	100.0

6.2 Results

To provide a quantitative evaluation of intent recognition performance, we use two measures:

- *Accuracy rate* = the ratio of the number of observation sequences, of which the winning intentional state matches the ground truth, to the total number of test sequences.
- *Correct Duration* = C/T, where C is the total time during which the intentional state with the highest probability matches the ground truth and T is the number of observations.

6.2.1 Similar-looking Activities

As we can see from Table 1, the system performs substantially better when using context than it does without contextual information. Because *meeting* and *passing* can, depending on the position of the observer, appear very similar, without context it may be hard to decide what two agents are trying to do. With the proper contextual information, though, it becomes much easier to determine the intentions of the agents in the scene.

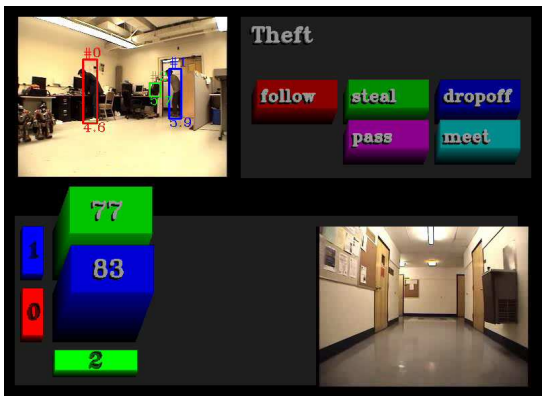

Figure 2: An observer robot catches an agent stealing a bag. The top left video is the observer's viewpoint, the top left bars represent possible intentions, the bottom right bars are the robot's inferred intentions for each agent (with corresponding probabilities), and the bottom right video is the patrol robot's viewpoint.

6.2.2 Intention-based Control

In both the scenarios we developed to test our intention-based control, our robot correctly inferred the ground-truth intention, and correctly responded the inferred intention. In the theft scenario, the robot correctly recognized the theft and reported it to the patrol robot in the hallway, which was able to track the thief. In the bag drop scenario, the robot correctly recognized that dropping a bag off in a hallway is a suspicious activity, and was able to follow the suspicious agent through the hall. Both examples indicate that intention-based control using context and hidden Markov models is a feasible approach.

6.2.3 Lexical-digraph-based System

To test the lexically-informed system, we considered three different scenarios. In the first, the robot observed a human during a meal, eating and drinking. In the second, the human was doing homework, reading a book and taking notes on a computer. In the last scenario, the robot observed a person sitting on a couch, eating candy. A trashcan in the scene then catches on fire, and the robot observes the human using a fire extinguisher to put the fire out.

Defining a ground truth for these scenarios is slightly more difficult than in the previous scenarios, since in these scenarios the observed agent performs multiple activities and the boundaries between activities in sequence are not clearly defined. However, we can still make the interesting observation that, except on the boundary between two activities, the correct duration of the system is 100%. Performance on the boundary is more variable, but it isn't clear that this is an avoidable phenomenon. We are currently working on carefully ground-truthed videos to allow us to better compute the accuracy rate and the correct duration for these sorts of scenarios.

7 CONCLUSIONS

In this paper, we proposed an approach to intent recognition that combines visual tracking and recognition with contextual awareness in a mobile robot. Understanding intentions in context is an essential human activity, and with high likelihood will be just as essential in any robot that must function in social domains. Our approach is based on the view that to be effective, an intent recognition system should process information from the system's sensors, as well as relevant social information. To encode that information, we introduced the lexical digraph data structure, and showed how such a structure can be built and used. We discussed the visual capabilities necessary to implement our framework, and validated our approach in simulation and on a physical robot.

REFERENCES

R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience (2000)

L. R. Rabiner, A tutorial on hidden-Markov models and selected applications in speech recognition, in Proc. IEEE 77(2) (1989)

P. Pook and D. Ballard, Recognizing teleoperating manipulations, in Int. Conf. Robotics and Automation (1993), pp. 578585.

G. Hovland, P. Sikka and B. McCarragher, Skill acquisition from human demon- stration using a hidden Markov model, Int. Conf. Robotics and Automation (1996), pp. 27062711.

K. Ogawara, J. Takamatsu, H. Kimura and K. Ikeuchi, Modeling manipulation inter- actions by hidden Markov models, Int. Conf. Intelligent Robots and Systems (2002), pp. 10961101.

A. Tavakkoli, R. Kelley, C. King, M. Nicolescu, M. Nicolescu, and G. Bebis, "A Vision-Based Architecture for Intent Recognition," *Proc. of the International Symposium on Visual Computing*, pp. 173-182 (2007)

P. Forssen, "Maximally Stable Colour Regions for Recognition and Matching," *CVPR 2007*.

M. Marneffe, B. MacCartney, and C. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses," *LREC 2006*.