2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Interpreting Multimodal Referring Expressions in Real Time

David Whitney* Miles Eldon* John Oberlin Stefanie Tellex

Abstract— **Humans communicate about objects using language, gesture, and context, fusing information from multiple modalities over time. Robots need to interpret this communication in order to collaborate with humans on shared tasks. Processing communicative input incrementally has the potential to increase the speed and accuracy of a robot's reaction. It also enables the robot to incorporate the relative timing of words and gestures into the understanding process. To address this problem, we define a multimodal Bayes filter for interpreting a person's referential expressions to objects. Our approach outputs a distribution over the referent object at 14Hz, updating dynamically as it receives new observations of the person's spoken words and gestures. We collected a new dataset of people referring to one of four objects in a tabletop setting and demonstrate that our approach is able to infer the correct object with 90% accuracy. Additionally, we augment and improve our filter in a simulated home kitchen domain by learning contextual knowledge in an unsupervised manner from existing written text, increasing our maximum accuracy to 96%, even with an increase in the number of objects from four to seventy.**

I. INTRODUCTION

In order for humans and robots to collaborate in complex tasks, robots should be able to understand people's references to objects in the external world. People provide these signals continuously using language and gesture, and exploit contextual background information to disambiguate requests. Cognitive science experiments have shown that highly successful teams rarely make explicit requests from one another and instead infer correct actions as needs arise [16]. Responding quickly and incorporating the relative timing of speech and gesture is critical for accurate understanding in human-human interaction [4].

To provide a foundation for these capabilities, we propose a Bayes filtering approach for interpreting multimodal information from language and gesture [26]. Our framework relies on a factored observation probability that fuses information from language, context, and gesture in real time to continuously estimate the object a person is referring to in the real world. We demonstrate that our approach quickly and accurately fuses multimodal information in real time to continuously estimate the object a person is referencing.

We also show that our approach can use contextual information, such as the knowledge of which ingredients or tools are likely to be used together, along with language and gesture to disambiguate requests. In this paper we focus on the home kitchen domain, generating contextual information in an unsupervised manner by processing an online repository of recipes. Recipes provide semi-structured data that can be automatically mined for contextual information and then combined with the person's language and gesture to interpret a request.

We evaluate our model in simulation and in the real world. In simulation, we use Amazon Mechanical Turk to collect referring expressions. We then test those expressions against our system in a simulated kitchen of seventy items. In the real world, we run trials with and without the robot. In trials without the robot, the user refers to several objects in a row, switching objects on a fixed schedule. In the robot trials, the user asks the robot to hand over one of several items on a table, and only switches once the robot has completed the hand-off. In simulation, we have an accuracy of 95.55%. In our non-robot trials, we have an accuracy of 90.77%. In our robot study, the robot hands over the correct item on the first try 80% of the time.

II. RELATED WORK

Language understanding for robots is a very active research topic. We can divide the field into two domains: continuous and batched interpretation. Batched interpretation is highly applicable in written communication [15, 6, 14, 21], but in recent years continuous interpretation has proved more valuable in the real-time domain. Here we will focus largely on works that have provided a method of continuous language interpretation.

Kennington and Schlangen [13] created a discriminative model of incremental reference resolution. In their work, the authors use wizarded trials of reference resolution to collect training sentences, which they use to train a logistic regression model. Their work is quite successful, but requires data collection and hand-crafting features. We found our more simple unigram model to be sufficient once combined with gesture. In a more complex domain, a more complex model may be required.

Funakoshi et al. [9] also created a model of incremental reference resolution. Like us, their model is based on a Bayesian network design, and is able to consider different domains for words. In "Bring me the red box, and the blue one, too," their model would understand "one" refers to the general concept of box. We felt that work focused more on depth in a single modality, where our goal was breadth across multiple modalities.

The majority of gesture systems today focus on gesture recognition [20] which is a classification task that does not require the location or orientation of the gesture [27, 18]. Often, this recognition is performed in batch, and has a

*First two authors contributed equally, and ordering is randomized.

All authors are with the Department of Computer Science, Brown University. {dwhitney,meldon,joberlin,stefie10}@cs.brown.edu

slightly different goal, namely to identify times in a video clip in which certain gestures occur. Many approaches to recognition use discriminative models [28, 23], which have been shown to be more accurate than their generative counterparts. The regression-like nature of pointing, however, makes a discriminative approach more difficult. Pointing is not a classification problem, as the goal is a real-valued set of numbers, namely the coordinates in space (x,y,z) the user is pointing to. Our solution is to extend a cone from the wrist of the user, with objects closer to the center ray of the cone considered more likely targets. This approach has been successful before, as shown by Schauerte et al. [24]. In that work they identify an object from a still image of a person pointing. They extend a cone with a Gaussian distribution from the tip of the arm.

Other work with collaborative robots exist. Foster et al. [8] have done research with a bar-tending robot. This research used a rule-based state estimator, and delivered drinks from fixed positions behind the bar to multiple users based on their speech and torso position. Similarly, we combine input from multiple modalities, but our work uses a probabilistic approach, allowing for smooth incorporation of dynamic gestures such as pointing. Bohus and Horvitz [3] have worked with a robotic guide that directs users searching for specific rooms in a building. Again, our work differs in that their model has prior knowledge about the location of the desired rooms, whereas our does not know the location of the desired objects. This system combines the modalities of head direction and speech, but not any other form of gesture.

Matuszek et al. [22] present a multimodal framework for interpreting references to tabletop objects using language and gesture. Our approach similarly focuses on tabletop objects but integrates language and gesture continuously. Additionally, their work has the user sitting at a table, meaning their pointing gestures occur several inches from the referent. In our work, the user stands several feet from the table, making our pointing gestures less easy to parse.

III. METHODOLOGICAL APPROACH

Our aim is to estimate a distribution over the set of objects that a person could refer to, given language and gesture inputs. We frame the problem as a Bayes filter [26], where the hidden state, $x \in \mathcal{X}$, is the the object in the scene that the person is currently referencing. The robot observes the person's actions and speech, $\mathcal{Z}$, and at each time step estimates a distribution over the current state, x_t:

$$p(x_t|z_{0:t}) \tag{1}$$

To estimate this distribution, we alternate performing a time update and a measurement update. The time update updates the belief that the user is referring to a specific object given previous information:

$$p(x_t|z_{0:t-1}) = \sum_{x_{t-1}\in\mathcal{X}} p(x_t|x_{t-1}) \times p(x_{t-1}|z_{0:t-1}) \tag{2}$$

The time update includes the transition probability from the previous state to the current state. Our various models for this probability are illustrated in Figure 1 and Section III-B.

The measurement update combines the previous belief with the newest observation to update each belief state:

$$p(x_t|z_{0:t}) = \frac{p(z_t|x_t) \times p(x_t|z_{0:t-1})}{p(z_t|z_{0:t-1})} \tag{3}$$

$$\propto p(z_t|x_t) \times p(x_t|z_{0:t-1}) \tag{4}$$

A. Observation Model

The observation model calculates the probability of the observation given the state. Each observation is a set of the user's arm position and speech, $\langle l, r, s\rangle$ where:

- l represents a vector from the elbow (l_o) to the wrist (l_v) of the left arm.
- r represents a vector from the elbow (r_o) to the wrist (r_v) of the right arm.
- s represents the observed speech from the user, consisting of a list of words.

Formally, we have an observation model of the form:

$$p(z_t|x_t) = p(l, r, s|x_t) \tag{5}$$

We factor our observations assuming that each modality is independent of the others given the state. Namely, we are assuming that if we know the true object, the probabilities of the user pointing at that object with their left hand or right hand are independent:

$$p(z_t|x_t) = p(l|x_t) \times p(r|x_t) \times p(s|x_t) \tag{6}$$

The following sections describe how we model each type of input from the person.

1) Gesture: We model pointing gestures as a vector through three dimensional space. First, we calculate a gesture vector using the skeleton pose returned by NITE [1]. We compute a vector from the elbow to the wrist, then project this vector so that the origin is at the wrist. Next, we calculate the angle between the gesture vector and the vector from the elbow to the center of each object, and then use the PDF of a Gaussian ($\mathcal{N}$) with variance (σ) to determine the weight that should be assigned to that object. We define a function $\mathrm{A}(o, p, x)$ as the angle between the point p and the center of mass of object x with the given origin, o. Then

$$p(l|x_t) \propto \mathcal{N}(\mu_l = 0, \sigma_l)[A(l_o, l_v, x_t)] \tag{7}$$

$$p(r|x_t) \propto \mathcal{N}(\mu_r = 0, \sigma_r)[A(r_o, r_v, x_t)] \tag{8}$$

While gesture remains a continuous input throughout the entire interaction, many gestures have little or no meaning, such as scratching your nose or crossing your arms. To allow for these without overloading the model with noise, we treat any gesture observation that is greater than some angle θ away from all objects as applying uniform probability to all objects. Mathematically:

$$p(l|x_t) \propto \begin{cases} \frac{1}{|\mathcal{X}|} & \text{if } A(l_o, l_v, x') > \theta, \forall x' \in \mathcal{X} \\ \mathcal{N}(\mu_l = 0, \sigma_l)[A(l_o, l_v, x_t)] & \text{otherwise} \end{cases} \tag{9}$$

The observation for the right arm is calculated in the same manner.

Note that at each time step, a single pair of arm positions are observed. Full gesture information results from the fusing of positions over time.

2) Speech: We model speech with a unigram model, namely we take each word w in a given speech input s and calculate the probability that, given the state x_t, that word would have been spoken.

$$p(s|x_t) = \prod_{w \in s} p(w|x_t) \tag{10}$$

To account for words that don't appear in the corpus, we incorporate an epsilon probability for all words that would otherwise have zero probability and then normalize the distribution. When no words are spoken, we assume a null word which has a uniform distribution over the objects. This effect means that spoken words cause a discrete bump in probability according to the language model, which then decays over time.

B. Transition Model

Context is incorporated in our transition model, using learned knowledge of related ingredients to better predict future requests. In our home kitchen domain, the user requests ingredients for a recipe. Therefore the desired ingredient is the hidden state, and transitions are nonuniform. Recipes generally use ingredients in similar orders. For example, dry ingredients are used in sequence, or peanut butter follows white bread and grape jelly. With this knowledge we are able to estimate transition probabilities. In other domains, estimates will be more difficult to generate, so we developed a context-free transition model as well.

1) Modeling Non-Contextual Information: When contextual information is not available, we assume that a person is likely to continue referring to the same object, and at each timestep has a large probability, c, of transitioning to the same state:

$$p(x_t|x_{t-1}) = \begin{cases} c & \text{if } x_t = x_{t-1} \\ \frac{1-c}{|\mathcal{X}|-1} & \text{otherwise} \end{cases} \tag{11}$$

This assumption means that the robot's certainty slowly decays over time, in the absence of corroborating information, converging to a uniform distribution. It enables our framework to integrate past language and gesture information but also quickly adapt to new, conflicting information because it assumes the person has changed objects.

2) Modeling Contextual Information: To model contextual information, we assume that the next object that a person requests depends on the previous object, as well as the information the robot can observe from language and gesture. We empirically calculate transition probabilities by applying language modeling techniques to a large corpus of recipes, C. We consider each recipe as a document, $d_0 \dots d_n \in C$ which contains an ordered list of ingredients, $d_i^0 \dots d_i^k$. We treat this list as an ordered list of states in our model and use it to calculate transition probabilities by mining co-occurrence statistics. Figure 1 provides the graphical models for the four approaches we compare in this paper, using increasing amounts of context to interpret a person's language and gesture.

Our first approach uses a unigram estimator based on individual ingredient frequencies, but does not take the history of past states into account [25, 19, 17]. For example, one of the most frequent ingredients is salt, occurring 3.15% of the time in our dataset. The unigram model makes it more likely the robot will fetch the salt but does not incorporate information about previous ingredients that were used. To compute this model, we count the number of times we observe state x_t compared to the total number of observed states ($x_{0:n}$). Formally:

$$p(x_t|x_{t-1}) = \frac{|\{\forall d_i^k | d_i^k = x_t \in C\}|}{|\{\forall d_i^k \in C\}|} \tag{12}$$

This model gives higher probabilities to more common ingredients, and does not consider past states. An estimator using a purely unigram model would always predict salt as the most likely ingredient.

To incorporate more context we use a bigram model to incorporate one previous state to inform the robot's decision. Formally, we model the probability of the next state, x_t given the previous state, x_{t-1} by counting bigram co-occurrence statistics in the corpus:

$$p(x_t|x_{t-1}) = \frac{|\{\forall d_i^k, d_i^{k+1} \in C | d_i^k = x_{t-1} \wedge d_i^{k+1} = x_t\}|}{|\{\forall d_i^k, d_i^{k+1} \in C\}|} \tag{13}$$

The graphical model for the bigram approach appears in Figure 1(c). Similarly, we can use two previous states to create a trigram model:

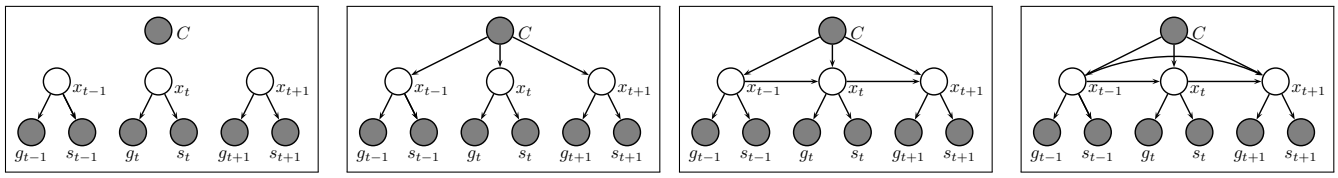
$$p(x_t|x_{t-1}, x_{t-2}) = \frac{|\{\forall d_i^k, d_i^{k+1}, d_i^{k+2} | d_i^k = x_{t-2} \wedge d_i^{k+1} = x_{t-1} \wedge d_i^{k+2} = x_t\}|}{|\{\forall d_i^k, d_i^{k+1}, d_i^{k+2} \in C\}|} \tag{14}$$

The graphical model for the trigram approach appears in Figure 1(d). While increasing the size of the history adds contextual information, it causes issues with sparseness and compute time, with diminishing returns on accuracy. In our research we found a plateauing of accuracy after trigrams.

3) Training: Our corpus consists of 42,212 recipes collected from `www.allrecipes.com` using a web crawler. We chose the website for its large collection, varied cuisine, and most importantly, ingredient ordering. All ingredients are listed in the order they are used in the recipe. Each recipe includes a title, the ingredients, the steps, and an end of recipe tag.

The following algorithm applies equations (13) and (14) to our corpus.

- Given the previously used ingredient (or past two ingredients), for each recipe, iterate through the list of ingredients.



(a) Uninformed Transitions (no dependency on corpus or previous states).

(b) Unigram model (dependency on corpus, but not previous states).

(c) Bigram model (dependency on corpus as well as one previous state).

(d) Trigram model (dependency on corpus as well as two previous states).

Fig. 1. Different models for our approach, using increasing amounts of context. Shaded variables are observed.

- If a match is found between the input and the current ingredient(s), record the next ingredient in the recipe.
- After scanning all recipes, return the list of ingredients used after the given input, ranked by the number of times they occurred.
- A numerical probability can be constructed by dividing each count by the sum of the counts.

Examples of the top ten unigrams, bigrams and trigrams appear in Figure 2.

C. Model Parameters

We tuned model parameters by hand. We generated the language model from hand-crafted data combined with the results of our pilot studies. After our initial tuning, we fixed model parameters, and results reported in the paper all use the same fixed set of parameters. We expect that as we add larger sets of objects, a language model trained using data from Amazon Mechanical Turk or other corpora will be necessary to increase robustness over a larger set of objects.

In our experiments, we had the following parameters: the uniform transition probability, c, was 0.9995. We set this parameter to give an object that has 100% confidence an approximately 10% drop in confidence per second with all null observations. Standard deviation for the Gaussian used to model probability of gesture, σ_l, σ_r, and σ_h was 1.0 radians. We found that this standard deviation allowed for accurate pointing, without skewing the probabilities during an arm swing. The language model consisted of 16 unique words, containing common descriptors for the objects such as "bowl," "spoon," "metal," "shiny," etc. It also included words that were commonly misinterpreted by the speech recognition system, such as "bull" when the user was requesting a bowl.

D. Algorithm

Algorithm 1 shows pseudocode for our approach, generating a belief distribution over the possible current states $bel(x_t)$, while Figure 3 shows an example of the system's execution. The person's speech is ambiguous, and the system initially infers an approximately bimodal distribution between the two bowls. The robot does not hand over any object, which elicits a disambiguating response from the person, who points at the appropriate object. The model incorporates information from language and infers the person is referring to the blue bowl.

Input: $bel(\mathcal{X}_{t-1}), z_t$

Output: $bel(\mathcal{X}_t)$

for $x_t \in \mathcal{X}_t$ **do**

$$\bar{bel}(x_t) = \sum_{x_{t-1} \in \mathcal{X}_{t-1}} p(x_t|x_{t-1}) * bel(x_{t-1})$$

if not is_null_gesture(l)

$$\bar{bel}(x_t) = p(l|x_t) * \bar{bel}(x_t)$$

if not is_null_gesture(r)

$$\bar{bel}(x_t) = p(r|x_t) * \bar{bel}(x_t)$$

for $w \in s$ **do**

$$\bar{bel}(x_t) = p(w|x_t) * \bar{bel}(x_t)$$

end

$$bel(x_t) = \bar{bel}(x_t)$$

end

Algorithm 1: Interactive Bayes Filtering Algorithm

Although in this example we are demonstrating the approach at two specific timesteps, the system updates its distribution at 14Hz, enabling it to fuse language and gesture as it occurs and quickly updating in response to new input from the person, verbal or nonverbal. Our approach runs on an Asus machine with 8 2.4 GHz Intel Cores that is also performing all perceptual and network processing. This system is used in conjunction with the Baxter Robot and a Kinect V1.

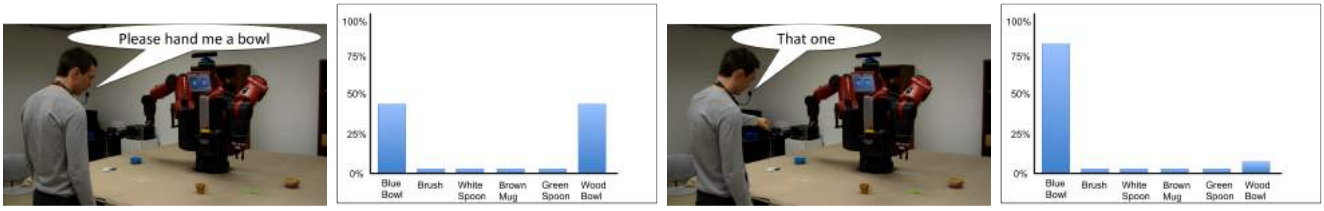
IV. EVALUATION

We evaluated our model through several methods. We first ran simulated trials in our home kitchen domain to detect the efficacy of using contextual information in specific domains. We then ran a system comprehension user study without contextual information to ensure the system's reliability in interpreting referring expressions in a closed environment. Finally, to both show the effectiveness of our model in the real world, as well as demonstrate the ways in which social feedback can play into the model in the future, we ran real world experiments with a robot using this system interacting with human users asking for common kitchenware[1].

[1]Unfortunately, we were unable to test our contextual model in real world. Our contextual simulation study had 70 items in the pantry, and we currently do not have access to a system that can identify and interact with 70 items at once.

(a) Unigrams.	(b) Bigrams.	(c) Trigrams.
salt	salt, black pepper	all-purpose flour, baking powder, salt
white sugar	all-purpose flour, baking powder	all-purpose flour, baking soda, salt
butter	baking soda, salt	baking powder, baking soda, salt
all-purpose flour	baking powder, salt	white sugar, eggs, vanilla extract
water	butter, white sugar	all-purpose flour, baking powder, baking soda
eggs	white sugar, eggs	butter, white sugar, eggs
garlic	all-purpose flour, salt	eggs, vanilla extract, all-purpose flour
olive oil	onion, garlic	all-purpose flour, white sugar, baking powder
vanilla extract	all-purpose flour, baking soda	vanilla extract, all-purpose flour, baking powder
	eggs, vanilla extract	olive oil, onion, garlic

Fig. 2. Top ten ingredient unigrams, bigrams, and trigrams from our training procedure.



(a) Ambiguous speech. (b) State estimate during ambiguous speech. (c) Clarification with gesture. (d) State estimate after clarification.

Fig. 3. After an ambiguous spoken request (a), the model has a uniform distribution between two objects (b). The robot responds by indicating confusion. Clarification with gesture (c) causes a probabilistic update leaving the model highly confident it has inferred the correct object (d). The robot responds by smiling and handing the user the object they referenced.

A. Simulation Results

Next we assess our model's accuracy at inferring ingredients based on a person's requests. Context is most valuable when there are many possible objects that the robot could hand to the person, and we wanted to evaluate our model on a large set of recipes and varied natural language input so we conducted this evaluation using Amazon Mechanical Turk data along with simulated gesture input.

As the number of ingredients the robot interacts with increases, it needs more information to pick the correct one. For example, in a small kitchen there may only be white sugar. The request "hand me the sugar" is unambiguous and the robot easily identifies the correct ingredient. A larger kitchen may have white sugar, brown sugar, and powdered sugar. The request has now become ambiguous, and contextual information becomes necessary to infer the correct object that the user desires.

For our study, we presented a series of photos to AMT workers. Each photo contained all the ingredients needed for a recipe in a kitchen setting. The workers typed requests to the robot. Each worker typed two requests for each ingredient: an ambiguous request, and an unambiguous request. Once the data was collected, the requests were fed as simulated speech to our system. We assessed accuracy by recording whether the system inferred the correct ingredient for each request. We collected a total of 1640 commands over 5 recipes not used in the training set.

Our system had a simulated 'pantry' of objects. The set of ingredients were taken from the cookbook *How to Cook Everything*, under the sections "Kitchen Basics", "Everyday Herbs", and "Everyday Spices" [2]. The ingredients are described as staple ingredients.

Each ingredient in the pantry had several words associated with it. These words were the singular and plural forms of the ingredient's name, and allowed for the observation update to link speech to specific ingredients. For instance, lemon had two words associated with it: lemon and lemons. We did not add more descriptive words, like yellow, or round, but we are eager to explore more expressive observation models.

Due to the difficulty of collecting multimodal data for our large dataset, we augmented our system with simulated gesture. We created gesture observations by assuming that a person produced pointing gestures which identified a subset of ingredients, one of which was the one they were asking the robot to fetch. To simulate different amounts of ambiguity in gesture, we varied the size of the cluster, d, between 3, 5, 10, and ∞; here ∞ corresponds to using language only and no gesture.

Tables I(a), I(b), and I(c) show an evaluation of our system using uniform, unigram, bigram and trigram models. We report the model's accuracy at identifying the correct object to fetch for each request after the person's natural language input using 90% confidence intervals. First we observe that more specific gestures (with smaller cluster size d) leads to higher model performance. This result is unsurprising because the system has access to significantly more information when augmented with simulated gesture.

As a high-level trend, we observed a significant increase in performance comparing uniform to unigram. In our unigram model, the robot generates a prior distribution based on common ingredients learned from text, but does not consider objects previously used in the recipe. This model lets us infer

TABLE I

SIMULATED CONTEXT, LANGUAGE, AND GESTURE

(a) Results using Gesture without Language

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	23.41% ± 1.73%	15.49% ± 1.46%	8.84% ± 1.15%	0.67% ± 0.329%
Unigram	34.82% ± 1.94%	27.74% ± 1.83%	19.21% ± 1.60%	5.43% ± 0.92%
Bigram	42.74% ± 2.01%	35.73% ± 1.94%	28.23% ± 1.83%	12.68% ± 1.34%
Trigram	41.04% ± 1.99%	32.50% ± 1.91%	27.38% ± 1.81%	12.74% ± 1.35%

(b) Results Using Gesture with Ambiguous Language Requests

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	74.39% ± 1.78%	70.91% ± 1.84%	67.13% ± 1.91%	47.99% ± 2.03%
Unigram	75.61% ± 1.74%	72.56% ± 1.81%	70.61% ± 1.84%	52.74% ± 2.03%
Bigram	77.80% ± 1.69%	76.22% ± 1.73%	72.56% ± 1.81%	53.11% ± 2.03%
Trigram	77.38% ± 1.69%	75.12% ± 1.76%	72.68% ± 1.81%	53.72% ± 2.03%

(c) Results Using Gesture with Unambiguous Language Requests

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	94.63% ± 0.92%	93.96% ± 0.97%	93.41% ± 1.00%	87.50% ± 1.35%
Unigram	95.12% ± 0.87%	94.27% ± 0.94%	94.39% ± 0.94%	89.09% ± 1.27%
Bigram	95.67% ± 0.82%	95.00% ± 0.89%	94.27% ± 0.94%	88.66% ± 1.28%
Trigram	95.55% ± 0.84%	94.70% ± 0.90%	94.39% ± 0.94%	88.41% ± 1.30%

the correct action for ambiguous commands such as "fetch the sugar," which most often refers to white sugar rather than brown. This result demonstrates improved performance using information from text in all conditions, but does not integrate contextual information.

Third, we observed a further improvement using the bigram model and trigram model, which use the previous state as context. This performance gain is present under all language conditions, but is increased when commands are ambiguous and decreased for unambiguous commands. Table I(c), which uses unambiguous language, shows good performance by all models, including the uniform model which uses no information from text, and a very small positive effect from context. In contrast, Table I(a) shows results using gestures only, with increasing amounts of ambiguity; here there is a very large improvement from context, going from 23% correct with uniform to 42.7% with the bigram model. In this scenario, gesture provides a strong signal but also contains a large amount of noise; combining this information with context from previous requests significantly improves system accuracy.

Finally, Table I(b) shows a modest improvement from context. We expect to see a larger gain with more ambiguous language. In our data, many requests were ambiguous because of spatial language not capable of being understood by our approach. For example, a request such as "Please hand me the onion beside the garlic" would be ambiguous to our system because it cannot process spatial referring expressions. This provides an opportunity for context to disambiguate, but since both ingredients are used similarly, the contextual models are unable to determine what the user desires. In our data, many such examples occurred because images showed all ingredients for the same recipe. In the future we plan to explore language collected from environments where the ingredients were not laid out, and also over speech recognition errors; we expect contextual input would matter more in these scenarios. Despite the limitations of the language data collected on AMT, we still observed a modest improvement from context in this type of language. For instance, in one trial a user requested soy sauce by stating "get me the soy sauce it is next to the garlic." The unigram model estimated the user wanted garlic, as garlic is used more often than soy sauce, but the bigram model looked at the last used ingredient, coconut milk, and calculated soy sauce was used more often than garlic in that context. The trigram model plateaued relative to the bigram model, most likely due to issues of sparsity in the training data.

B. System Comprehension User Study Results

Our real-world experiments measured our algorithm's performance when a person referred to an object visually and with gesture. The subject stood in front of a table with four objects placed approximately one foot apart, forming four corners of a square. We instructed subjects to ask for the indicated object in the most natural way possible, using whatever combination of gesture and language they felt was appropriate. We indicated the object to refer to using a laser pointer, and we periodically shifted to a different object on a predetermined schedule. They wore a microphone, and we used the HTML5 Speech Recognition package in conjunction with Google Chrome to recognize speech. This package reported incremental output as recognition proceeds, and we performed a model update each time a new word was perceived. We used 13 subjects, and each subject participated in five trials, for a total of 65 trials.

Results showing the percent of the time the estimated most likely object was the true object appear in Table II with 95% confidence intervals. During a typical trial, the model starts out approximately uniform or unimodal on the previous object (we did not reset the model between trials.) As the subject points and talks, the model quickly converges to the correct object. Our first set of results give a sense of how quickly the model converges.

To assess overall accuracy, we report the system's accuracy at the end of a trial in Table III. Multimodal accuracy with language and gesture is more than 90%, demonstrating that our approach is able to quickly and accurately interpret unscripted language and gesture produced by a person.

The difference in accuracy between gesture alone and the multimodal output is not as large as one might expect. This is in part caused by the small delay in speech recognition software as opposed to the instantaneous gesture input. Additionally, many subjects leaned towards ambiguous speech, such as "Hand me that" while pointing, causing the speech accuracy for those trials to be 0%. There were some users, however, who relied on an equal mix of both, and showed large leaps in accuracy between arms and multimodal. The most extreme example is of a user who, over their five trials, achieved only 45.5% accuracy with gesture alone and 42.2% with speech alone, yet managed to achieve 85.7% multimodal accuracy, only 2 percentage points away from the sum of the two probabilities, showing the ease at which

TABLE II

REAL-WORLD RESULTS

Random	25%
Language only	32.4% +/- 10%
Gesture only	73.12% +/- 9%
Multimodal (Language and Gesture)	**81.99% +/- 5.5%**

TABLE III

REAL-WORLD RESULTS (END OF INTERACTION)

Random	25%
Language only	46.15%
Gesture only	80.0%
Multimodal (Language and Gesture)	**90.77%**

alternating speech and gesture can give incredibly accurate results overall. While a combination of ambiguous speech and gesture such as "that spoon" followed by a gesture would be more accurate than just a gesture, we found that most test subjects either spoke with complete ambiguity or none, using phrases either of the form "hand me that thing" or "hand me the silver spoon". Therefore we were unable to fully test this hypothesis.

C. System Interaction User Study Results

After successfully demonstrating our system in a closed environment, we ran trials involving a human user interacting with a robot. Whenever the system placed more than 70% confidence in any single object, the robot handed the person that object. We ran 40 trials, each with four objects on a table, two on each side of the robot. Users were instructed to pick an object and continue requesting it until the robot handed them the correct object. In 80% of the trials the robot handed over the correct object on the first try. In 65% of the trials the robot handed over the desired object after a single referring expression. These trials had an average latency of 1.2 seconds between the end of the referring expression and the beginning of the robot's reaction. On average, it took 15.8 seconds from the end of the referring expression to the time the user received the object they had requested.

In these trials, we calculate the latency in robot reaction when the robot correctly inferred the desired object as a result of the user's first referring expression. 65% of our trials fall into this category, resulting in an average 1.2 second delay between the end of the user's referring expression and the beginning of the robot's reaction. Since the robot was able to react before the user finishes their referring expression, some of the delays were slightly negative. This resulted in an average of 15.8 seconds between the time the user finished their request and the time they received the object they had referenced.

In the remaining 35% of the trials that the robot did not correctly infer the desired object from the first referring expression, 15% were failures where the robot simply didn't respond to the first referring expression and 20% were failures where the robot handed the user an object other than the one they requested. The former failure type can be largely attributed to rapid gestures and speech that were missed by our system. Mistranscription also played a role, but less of one. The latter failure type appears largely due to some quirks of NITE, in which the generated skeleton is actually superimposed slightly above the actual location on the body. As a result, the calculated vector came closer to the object behind the desired one, causing a failure.

V. CONCLUSION

We have demonstrated a Bayes filtering approach to interpreting object references. Our approach incorporated learned contextual dependencies, and ran in real time. This paper demonstrates steps toward continuous language understanding and more effective human-robot interaction.

In the future we plan to expand our language model to incorporate models of compositional semantics and lower-level visual features so that the robot is not limited to prespecified object models. Additionally we aim to enable the robot to generate social feedback based on its model using a POMDP framework [12], ultimately aiming to demonstrate that by providing appropriate social feedback, the robot elicits disambiguating responses from the person, increasing overall speed and accuracy of the interaction. Dragan and Srinivasa [5] created a framework for generating legible gesture, and we anticipate that enabling a robot to respond using gestures as in Holladay et al. [10] will further increase the efficacy of this system. We also plan to extend our language model so that it supports models of compositional semantics by embedding a parsing chart into the state [11, 7]. These methods will enable the robot to understand nested referring expressions such as "the bowl on the table" incrementally. Finally, we aim to extend our approach beyond just object references; a similar modeling approach could be used to understand references to locations in the environment, and ultimately general command interpretation.

REFERENCES

[1] Openni tracker. http://wiki.ros.org/openni_tracker, 2014.

[2] Mark Bittman. *How to Cook Everything*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.

[3] Dan Bohus and Eric Horvitz. Managing human-robot engagement with forecasts and... um... hesitations. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 2–9, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2885-2. doi: 10.1145/2663204.2663241. URL `http://doi.acm.org/10.1145/2663204.2663241`.

[4] Herbert H Clark and Meredyth A Krych. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50(1):62–81, 2004.

[5] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Robotics: Science and Systems*, June 2013.

[6] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. What to do and how to do it: Translating natural language directives into temporal

and dynamic logic representation for goal management and action execution. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 3768–3773, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

[7] Jay Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102, 1970.

[8] Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald P. A. Petrick. Two people walk into a bar: dynamic multi-party social interaction with a robot agent. In *International Conference on Multimodal Interaction, ICMI '12, Santa Monica, CA, USA, October 22-26, 2012*, pages 3–10, 2012. doi: 10.1145/2388676.2388680. URL `http://doi.acm.org/10.1145/2388676.2388680`.

[9] Kotaro Funakoshi, Mikio Nakano, Takenobu Tokunaga, and Ryu Iida. A unified probabilistic approach to referring expressions. In *Proceedings of the SIGDIAL 2012 Conference, The 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 5-6 July 2012, Seoul National University, Seoul, South Korea*, pages 237–246, 2012. URL `http://www.aclweb.org/anthology/W12-1633`.

[10] Rachel M Holladay, Anca D Dragan, and Siddhartha S Srinivasa. Legible robot pointing. RO-MAN, 2014.

[11] Daniel Jurafsky, Chuck Wooters, Jonathan Segal, Andreas Stolcke, Eric Fosler, G Tajchaman, and Nelson Morgan. Using a stochastic context-free grammar as a language model for speech recognition. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 189–192. IEEE, 1995.

[12] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2), 1998.

[13] Casey Kennington and David Schlangen. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1029`.

[14] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of HRI-2010*, 2010.

[15] Matthew MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 2 of *AAAI '06*, pages 1475–1482. AAAI Press, 2006. ISBN 978-1-57735-281-5.

[16] Jean MacMillan, Elliot E Entin, and Daniel Serfaty. Communication overhead: The hidden cost of team cognition. *Team cognition: Process and performance at the interand intra-individual level. American Psychological Association, Washington, DC. Available at http://www. aptima. com/publications/2004_MacMillan_EntinEE_Serfaty. pdf*, 2004.

[17] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.

[18] Matthew Marge, Aaron Powers, Jonathan Brookshire, Trevor Jay, Odest C Jenkins, and Christopher Geyer. Comparing heads-up, hands-free operation of ground robots to teleoperation. *Robotics: Science and Systems VII*, 2011.

[19] U-V Marti and Horst Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International journal of Pattern Recognition and Artificial intelligence*, 15(01):65–90, 2001.

[20] P. Matikainen, P. Pillai, L. Mummert, R. Sukthankar, and M. Hebert. Prop-free pointing detection in dynamic cluttered environments. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 374–381, March 2011. doi: 10.1109/FG.2011.5771428.

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

[22] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. Learning from unscripted deictic gesture and language for human-robot interactions. 2014.

[23] L. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383299.

[24] Boris Schauerte, Jan Richarz, Gernot Fink, et al. Saliency-based identification and recognition of pointed-at objects. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4638–4643. IEEE, 2010.

[25] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

[26] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2008.

[27] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[28] Sy Bor Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527, 2006. doi: 10.1109/CVPR.2006.132.