Interactive Multimodal Robot Dialog Using Pointing Gesture Recognition

Stefan Constantin(✉), Fevziye Irem Eyiokur, Dogucan Yaman, Leonard Bärmann, and Alex Waibel

Interactive Systems Lab, Karlsruhe Institute of Technology, Adenauerring 2, 76131 Karlsruhe, Germany
stefan.constantin@kit.edu

Abstract. Pointing gestures are an intuitive and ubiquitous way of human communication and thus constitute a crucial aspect of human-robot interaction. However, isolated pointing recognition is not sufficient, as humans usually accompany their gestures with relevant natural language commands. As ambiguities can occur both visually and textually, an interactive dialog is required to resolve a user's intentions. In this work, we tackle this problem and present a system for interactive, multimodal, task-oriented robot dialog using pointing gesture recognition. Specifically, we propose a pipeline constituted of state-of-the-art computer vision components to recognize objects, hands, hand orientation as well as human pose, and combine this information to identify not only pointing gesture presence but also the objects which are pointed at. Furthermore, we provide a natural language understanding module which considers pointing information to distinguish unambiguous from ambiguous commands and responds accordingly. Both components are integrated into the proposed interactive and multimodal dialog system. For evaluation purposes, we introduce a challenging benchmark set for pointing recognition from human demonstration videos in unconstrained real-world scenes. Finally, we present experimental results of both the individual components as well as the overall dialog system.

Keywords: Human-robot interaction · Pointing gesture recognition · Task-oriented dialog

1 Introduction

Human communication is inherently multimodal. Specifically, pointing gestures are an intuitive yet effective way of clarifying otherwise ambiguous intentions or utterances, with developmental psychology showing that infants already acquire pointing gestures before and in parallel to their language skills [12,50]. Therefore, to accomplish a truly natural Human-Robot Interaction (HRI) [6,15], it is

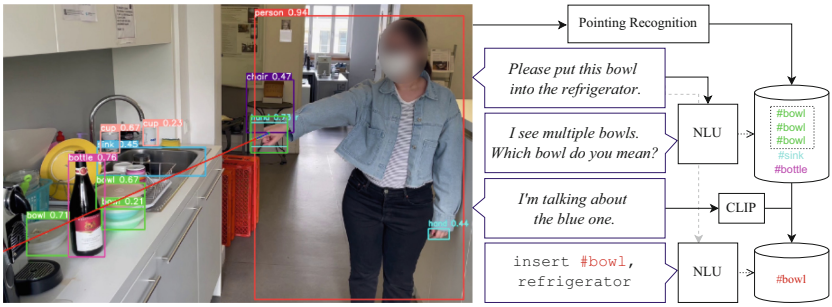

Fig. 1. An application example of the proposed interactive, multimodal dialog system

crucial for future assistive robot systems to correctly recognize and react to such gestures. Due to the tight connection between pointing and language, this must be combined with natural dialog to form a multimodal communication experience. For example, imagine an elderly care robot [55] hearing the command "please bring me that thing" and seeing the user pointing to a table with one or multiple objects. If there is only one object or the pointing gesture is very precise, the robot should immediately execute its task to fulfill the user's needs. However, if the situation is ambiguous, the robot should ask for a clarification ("which object do you mean exactly?"), that could lead to a user response like "the green one". Thus, ambiguities are resolved in a multimodal combination of pointing and dialog interactions.

In this paper, we present a system for task-oriented robot dialog that is able to achieve the aforementioned interactive, multimodal combination of pointing gesture recognition and natural language understanding (NLU). Building on top of recent advances in computer vision [7,38], we constructed a pipeline for pointing gesture recognition, with significantly less assumptions and constraints than previous work on pointing recognition [19]. Moreover, we propose a novel architecture for combining pointing recognition with a neural-network based NLU module (see Fig. 1). For evaluation purposes, we introduce a human pointing gesture dataset, particularly featuring ambiguities asking for dialog clarification, and use this to present extensive experimental results of our system.

Specifically, we present three major contributions: First, we combined a variety of state-of-the-art computer vision components including YOLOv5 object detection [20] for hand detection and object detection, human pose estimation [7], and hand pose estimation [47] into an integrated pointing recognition pipeline. The resulting system is able to analyze videos and determine whether a pointing gesture is present and if so, produce a list of pointing target candidate objects with labels and bounding boxes. While the system first detects and tracks hands to decide whether pointing happens or not, object detection and pose estimation models run only when the algorithm decides pointing occurs, thus saving computation resources. In the end, based on the estimated points on the forearm/hand, a line is interpolated and objects with intersecting bounding boxes are taken as pointing targets.

Second, we integrated the pointing recognition system into an interactive, task-oriented dialog system, useful for future human assistance devices [1] as well as robotics [2,49]. Specifically, the dialog system uses results from pointing recognition and prompts the user with a follow-up question in case of ambiguities. For training the Transformer-based [51] NLU module, an appropriate dataset was generated based on existing kitchen-related utterances [10]. All the time during dialog, the user can specify certain objects in a combination of pointing gestures and natural language, optionally using generic referring expressions like "this thing". Uncertainties in the pointing recognition are resolved using a follow-up questions, where the answer is interpreted using the CLIP model [36] or triggered a better pointing gesture of the user. To summarize, the combined task-oriented dialog system allows for inherently multimodal, interactive goal communication, resolving ambiguities using visual clues, pointing gesture recognition and natural language clarifications.

Finally, we created an evaluation dataset to quantify the performance of our pointing gesture recognition, featuring a diverse set of objects, locations, cameras and human subjects. It consists of videos of humans pointing at objects in unconstrained and cluttered real-world environments, accompanied with human-annotated natural language commands and possible clarification utterances to additionally enable the end-to-end evaluation of our interactive, multimodal dialog system. In contrast to existing datasets [8], this test bed explicitly asks for ambiguities on both the gesture and the language level, and thus provides a novel challenge to the research community. To foster future research on and unified comparison of interactive pointing dialog systems, we published our code, models, and trainings and evaluation datasets[1].

2 Related Work

2.1 Natural Language Understanding

The first approaches to natural language understanding (NLU) have been using fixed rules to map natural language to semantic representations [53,54]. However, the ability of such rule-based systems is limited as it seems impossible to cover the whole variety of natural language in rules. Therefore, neural networks were introduced to NLU [29], where the dependence on high-quality rules is replaced with a dependence on huge amounts of training data. Pre-trained, large models like BERT [13] and T5 [37] simplify the data collection as one can combine general natural language skills and knowledge from their pre-training with domain knowledge by fine-tuning them with a smaller, domain-specific dataset. More recently, even larger models are not even fine-tuned, but relevant behavior is elicited by using task-specific prompts [5,26]. For a more thorough review, we refer the reader to recent surveys like [30,52].

[1] https://github.com/msc42/dialog-using-pointing-gestures.

2.2 Pointing Recognition

Researchers utilize pointing gestures to improve HRI approaches and models. Pointing gesture recognition studies can be categorized into approaches that use a stereo camera or Kinect-based input to estimate 3D coordinates [3,11,14,17, 21,24,31,32] and approaches that use RGB camera input and estimate pointing direction based on 2D coordinates [18,27,28,34,42]. In addition, pointing gesture recognition methods can be classified by their used algorithms such as Hidden Markov Models (HMMs)-based [31,32], probabilistic approaches [9,46], and deep learning-based [3,17,27,28]. After decision of occurrence of pointing gesture, there are various ways to calculate pointing direction in terms of used body joints. The common body parts such as the hand, forearm, and face are used to calculate the line of sight between them. In [21], dense disparity maps are used to track pointing gestures. Similarly, [32] use a disparity map obtained by stereo camera input and skin-color classification to track the hand and face of the person, whereupon HMMs are trained to decide on the occurrence of a pointing gesture. For the decision of pointing gesture direction, three different approaches are compared: line of head-hand direction, forearm, and head orientation. Park and Lee [33] used stereo camera inputs and 3D particle filters to track hands, and different from the previous approach, they used HMMs as a two-stage method to decide the pointing direction more precisely. Similarly, Kehl and Van Gool [22] used multiview cameras to obtain 3D coordinates of points in the real world and used them to detect and track pointing gestures. The line of sight between the pointing fingertip and eyes is considered as pointing direction. In [34], a single RGB camera input is used to track and estimate the rotations of the hand pointing gesture and face. During fusing of face and hand inputs, the Dempster-Shafer theory is utilized. This approach benefits from prior information about the location of objects which can pointed at. In another 2D camera input based approach [43], a bottom-up saliency map in addition to the pointing gesture to identify referred-to objects is used. Then, attention to specified region is used to detect pointed objects.

In more recent studies, deep learning-based models have been becoming more useful for both 3D and 2D pointing gesture recognition. In [3,17], off-the-shelf models are used to calculate 3D vectors to obtain robust pointing gesture recognition. While Azari et al. [3] focused more on detection of face and hand areas, Hu et al. [17] estimated human body pose to obtain pointing line using eye and wrist coordinates. In [28], a pipeline that works with inputs from RGB drone camera is constructed. While the OpenPose model [7] is used to estimate human body pose, a Yolo-based detector [38] is trained to detect target objects. In a follow-up work [27], the authors extended their work using monocular simultaneous localization and mapping algorithm to estimate an unscaled point cloud. Using estimated 2D coordinates and camera calibration, they obtained estimated depth coordinates of hand and target objects.

2.3 Combination of NLU and Pointing Recognition

Bolt [4] introduced the combination of NLU and pointing gesture recognition. The system is constrained to draw shapes on specific positions on a screen, using

a rule-based natural language understanding system. If pointing gesture placeholders like "that" are recognized, pointing information is used to fill information in the placeholder. An immobile motion controller is used in the system.

In [48], a system that is more flexible is presented. Pointing gestures are recognized by cameras in different angles and light conditions. The natural language understanding system is also rule-based. The fusion of both modalities for this system is described in [16]. They rely mainly on the speech input because the pointing gesture recognition component is error-prone. If information in the speech input is missing, information of the recognized pointing gesture is added by a rule-based system. Pointing phrases like "that" are ignored because they are often misunderstood by their automatic speech recognition system. The system was evaluated with a real robot in a realistic scenario [49]. [42] also combines rule-based language parsing with line-of-sight-based pointing gesture recognition to perform saliency-based visual search.

Most of the newer systems [17,19] still use rule-based NLU and rule-based fusion of the pointing target objects with the semantic representation. In contrast, in [39] a POMDP is used for recognizing the natural language and pointing gesture as well as for learning to output the right response. [45] use verbal clues to disambiguate pointing gestures in a closed-world setting, comparing different approaches like Support Vector Machine and Decision Tree. Recently, Pozzi et al. [35] proposed an idea similar to our work, i.e. to disambiguate robot commands using pointing gestures, but provide only preliminary studies.

Most similar to our work, Chen et al. [8] introduced the task of Embodied Reference Understanding (ERU), i.e. resolving a natural language referring expression in combination with a pointing gesture. The authors also created a dataset and an end-to-end model for their task. However, they focused on single-round reference understanding, and explicitly asked annotators to give unambiguous referring expressions. In contrast, in our work, we embrace ambiguity and embed the task of ERU into a more high-level goal of performing task-oriented NLU in a robot scenario, including the ability to resolve ambiguities using interactive dialog.

3 Methodology

3.1 Pointing Recognition

We aim to recognize the pointing target object or objects with using only 2D images, namely video frames, without further information such as depth. Our approach is to detect hands, decide pointing time, and draw a line during the pointing time. Meanwhile, we run an object detector to detect the existing objects in the scene. In the end, we get the objects overlapping with the line and perform majority voting using output from each frame in the pointing time in order to decide the final label or labels. However, in order to perform this, we must overcome several challenges. First, we need to detect hand or hands precisely. Second, we have to detect almost all objects in the image. Third, we must decide how to draw the line since there are several different factors that may affect the accuracy of the line.

Last but not the least, we have to classify detected hands correctly since we need information on hand-side when we draw a line.

In order to overcome aforementioned challenges, we propose a method that is comprised of different components. We first propose to use YOLOv5 [20] to perform hand detection due to its success and robustness. We train the YOLOv5 model on a large-scale hand dataset (100DOH) [44] and use the *100k Frames* version. Since we need to classify the hand to draw the line accurately, we divide the hands into two categories —left hand and right hand— and train the YOLOv5 model to learn left hand and right hand as two different object classes. However, in our experiments, we realized that although YOLOv5 is able to accurately detect the hands, it is not accurate to assign correct labels. Therefore, we decided to use an additional classification model to classify detected hands. For this, we employ the Mobilenetv2 model [41] due to its performance in terms of accuracy and running time. We performed fine-tuning with the Mobilenetv2 model on the same hand dataset by addressing the task as a binary classification. By means of Mobilenetv2, we achieve much better classification accuracy and this boosts the performance of the overall system. Moreover, for the object detection, we benefit from the pretrained YOLOv5 that was trained on the MSCOCO dataset [25].

After the decision that there is a pointing based on the hand detection and tracking, we run an object detector. Thereafter, we utilize the OpenPose pose estimation model [7]. We use elbow and wrist points to draw the line and extend that line till the end of the image. After that, we get the objects overlapping with the line as pointing target objects. However, according to the experimental results, we realized that this method fails when the hand, and specifically the index finger, points at a direction different from the arm. In order to overcome this problem, we utilize estimated points on the forefinger to draw the line. For this, we crop detected hands and send them to the OpenPose hand pose estimation model [47] to get hand pose estimation output. We utilize two adjacent points on the forefinger to draw the line. In the end, we follow the same strategy and take the objects that overlap with the drawn line. This approach is more robust against changes in the hand direction.

3.2 Multimodal Natural Language Understanding

We first give a broad overview of the purpose and interface of the natural language understanding (NLU) component. Details to our specific implementation can be found afterwards.

The NLU component receives three inputs: First, a boolean p whether a pointing gesture was recognized. Second, a multiset $O = \{o_1, ..., o_N\}$ of recognized objects from the pointing recognizing component, where each o_i is a text label from the object recognition (e.g. “apple”). Third, the natural language utterance u provided by the user. The variable u may either contain a dialog object hint or an unspecific phrase, e.g. “give me that apple” vs. “give me that thing”.

The goal of our NLU component is to output a symbolic API call which could be executed by a robot planner or a similar receiving component. If p indicates that no pointing gesture is recognized, u is processed by the NLU component

Table 1. Ambiguous case responses, assuming a pointing gesture is provided. *obj. rec.* = $|O|$ and its content (s.c. = same class, d.c. = different classes). *obj. hint* = dialog object hint in user utterance (not rec. = hint not in recognized objects, mult. rec. = obj. recognized multiple times). X = dialog object hint, $O = \{A, B, ...\}$

Obj. rec.	Obj. hint	Response
0	✓	I cannot see the X. Please point exactly to it
0	—	I cannot see anything. Please point exactly to the object you refer to
1	Not rec.	I cannot see the X, but I see a/an A. Please point exactly to the X
$\geq$2	Not rec.	I only see A, B, ... Please point exactly to the X
$\geq$2	Mult. rec.	I see multiple X. Which X do you mean?
$\geq$2, s.c.	—	Which A do you mean?
$\geq$2, d.c.	—	Please point exactly to the object you refer to or describe it

and an API call like "carry apple" is generated, where the receiver of this API call must do the grounding to find the object by itself (not part of this work). In the case of using a natural language command and a pointing gesture, an API call like "carry #apple" is generated if the natural language command and the pointing gesture could be processed unambiguously. The "#" indicates the receiver can look for the bounding box of the object in the list of the recognized objects of the pointing recognition component. In this list, only one apple can be included because otherwise, the request would be ambiguous. If the request of the user is ambiguous, the system yields an appropriate natural language response as specified in Table 1. In this work, an API call can use up to two objects and one of them can be grounded in a pointing gesture.

To instantiate the NLU model, we use a pretrained Transformer [51] model, specifically, T5-large [37] with 737 million parameters. The three NLU component inputs as defined above are passed to the Transformer in a textual way, specifically by concatenation with pipe and comma as separators, i.e. the input is "$p \mid o_1, ..., o_N \mid c$" (e.g. "*yes |apple, bowl |Bring me the apple*"). The output is either the API call or one of the responses of Table 1. To train the model, a generated dataset as described in section Sect. 4.2 is used. We used a batch size of 32, a learning rate of 2.5e-4, and the Adam optimizer [23]. The weights of the embedding layer and the first two Transformer encoder blocks were frozen during the complete training (679 million trainable parameters).

3.3 Interactive Pointing Dialog

To realize an interactive, multimodal dialog system for instructing a robot to perform a specific task, the pointing recognition and multimodal NLU components introduced above are combined. The interface between both components is realized by a cache storing the latest pointing recognition candidates and read by

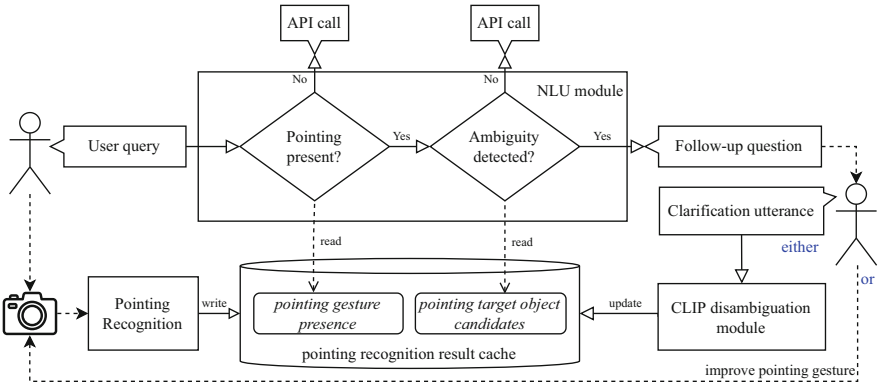

Fig. 2. Overview of the full interactive pointing dialog system, combining pointing recognition and multimodal NLU. Please note that the decisions depicted inside the NLU module are not actually hard-coded, but trained into the Transformer model. For a step-by-step application example, see Fig. 1

the NLU module. If an ambiguity is detected, the NLU module yields a follow-up question presented to the user. The pointing recognition candidates cache can now either be updated by the user improving their pointing gesture or by a clarification utterance, which is processed by a vision-language disambiguation module to filter the elements of the cache. A sketch of the resulting system is shown in Fig. 2, and a more detailed description follows below.

Without a specific trigger, the pointing recognition component runs permanently and processes the camera images. If a pointing gesture is recognized, the resulting pointing target object candidates are written into a cache, which serves both as communication buffer to the NLU module as well as a temporal smoothing of the pointing results. Specifically, each entry in the cache is a bounding box associated with a recognized object label and a numeric identifier.

The NLU module is triggered by an input user query like "Please give me that thing". We assume the input to be text, as it would be recognized by an automatic speech recognition system when deployed to a real robot. While the NLU module is a black-box Transformer model as described in Sect. 3.2, this model is trained with data leading to a two-stage decision process as depicted in Fig. 2. If p in the input indicates that there is no pointing gesture, pointing is irrelevant and an API call is produced immediately. Otherwise, the content of the pointing recognition cache (which is given as a secondary input to the model) determines whether there is an ambiguity or not. This either leads directly to an API call or a natural language follow-up question, which would be sent to a Text-to-Speech system.

In case a follow-up question was presented, the user can respond in two different ways: First, the user can answer the question with a natural language clarification utterance. In this case, we feed the input text to a CLIP [36] model that is able to predict the most related image given a text without further adaptation for the domain or task. In particular, we cut out the images of candidate

objects using the detected bounding boxes and address the task as a zero-shot classification task which means we provide one text and multiple images, and CLIP produces a similarity score for each image with respect to the semantic similarity with the text input. As a result, the pointing recognition cache is modified to only retain the candidates with a similarity score above an empirically determined threshold. Afterwards, the NLU is triggered again with the original user query, now accessing the updated pointing recognition cache and thus (hopefully) resolving the ambiguity.

The second option of the user to react to a follow-up question is to improve his pointing gesture, e.g. by moving closer to the target object. In that case, the pointing recognition component receives the new camera images and updates the pointing recognition cache. If such update happens in a reasonably short amount of time after a follow-up question was issued (3 s), we consider this as a trigger and feed the original user query to the NLU module again, similar to the above. Note that for both options of ambiguity resolution, if the ambiguity still remains, the process can be repeated. Therefore, both options end with re-triggering the NLU module with the original user query.

4 Dataset

4.1 Human Pointing Dataset

To properly evaluate the performance of our pointing recognition component as well as the overall multimodal dialog system, we collected a test dataset of human pointing demonstrations. In particular, our dataset consists of 182 videos showing 4 different human subjects pointing at specific kitchen objects of 24 different classes (referring to the classes of the MSCOCO dataset [25]). The distribution of collected videos based on selected classes are shown in the supplementary material in A.3. Each video shows a human starting the pointing gesture, pointing for a few seconds, and then ending the gesture again. Video lengths are in the range of 1.1 to 7.8 s, with an average of 3.5 and median of 3.2 s. During recording, some participants spoke out the name of the object which is pointed at to undoubtedly specify their intended pointing target. However, this audio is only for annotation purposes and thus not part of the task or usable by our system.

The dataset features a huge variety of test cases and diverse visual challenges, including varying lightning conditions, locations, cameras, camera viewpoints, specific object instances and human appearance.

Specifically, videos were recorded at six different locations (each involving various viewpoints) using four different cameras, from laptop webcam to smartphone to GoPro camera. Due to practical considerations and object availability, the distribution of pointing target object classes is not uniform. In the supplementary material in A.3, you can see a figure that depicts this. Furthermore, there are cluttered scenes involving many objects, as well as challenging cases where the pointing target is hard to be unambiguously identified even for a human viewer. For some exemplary video shots, see Fig. 3.

Fig. 3. Samples from our human pointing dataset (more in the supplementary material)

Next to the pointing videos, we also collected another set of 40 short videos showing the same human subjects in the same environments, but without performing a pointing gesture. These scenes were collected with the goal of 1) controlling for false positives of the pointing recognition, and 2) testing the natural language component also in case there is no pointing gesture.

This sets our dataset aside from previous work, specifically *YouRefIt* [8], where pointing can be assumed to be present in every sample. Furthermore, in contrast to [8], we explicitly do not ask for unambiguous references, both concerning the pointing gestures as well as the annotated natural language commands (which are explained in the next section). This is because we do neither intend to study pointing gesture recognition in isolation nor embodied reference understanding in a single-turn setting, but we aim for multimodal, task-oriented, interactive and thus multi-turn dialog for resolving ambiguities.

4.2 Natural Language Understanding Dataset

For training the NLU module as described in Sect. 3.2, we generated a dataset where each sample consists of data mimicking the pointing recognition output joined with an input utterance (source) and an output text (target). Specifically, the sources have the format "$p \mid o_1, ..., o_N \mid u$" with p indicating whether a pointing gesture was recognized, $o_1, ..., o_N$ denoting the recognized objects and u being the utterance of the user, e. g. "yes |apple |give me the apple" or "yes |apple, orange |give me the object". The target is either an API call or – in case of an ambiguous source – an appropriate response, e. g. "carry #apple" or "Please point exactly to the object you refer to or describe it". See Sect. 3.2 and Table 1 for more information. The utterances of the training and validation dataset are based on the training and validation dataset of the EPIC-KITCHENS-100 dataset [10], respectively. Details on how we adapt these utterances to the format of our dataset can be found in the supplementary material in A.1, accompanied by samples from the generated data.

For the test dataset, we collected two utterances for each of the 182 recorded pointing videos and the 40 recorded non-pointing videos (see Sect. 4.1). We used the reference pointing information (test_{ref}) or, to also simulate the ambiguous cases, the output of the pointing recognition component with the hand pose estimation model, namely forefinger without margin (test_{hyp}), to annotate the target API call or clarification question in the same way as in the training dataset. Specifically, for test_{hyp}, we used the recognized pointing target objects from our

Table 2. Statistics of the NLU dataset. More details in the supplementary material

Response	Train	Val	$Test_{ref}$	$Test_{hyp}$
API call one entity w/o pointing	28,352	239	33	31
API call one entity w/ pointing	9,143	93	257	144
API call two entities w/o pointing	4,275	154	7	9
API call two entities w/ pointing	1,313	66	147	58
Ambiguity resolution through follow-up question	22,633	296	–	202
Total	**65,716**	**848**	**444**	**444**

pointing recognition component and, if it was not possible to output a command unambiguously with the recognized pointing targets, we adapted the targets to the responses from Table 1. We additionally collected a clarification utterance for each video, which serves as the user's response to the system's clarification questions from Table 1. To illustrate this, consider the command "please cut that fruit into small pieces" with an accompanying pointing gesture. If the system is not able to determine the referred object with a high confidence, it will respond with "Which fruit do you mean?". A possible clarification utterance is then "the yellow one".

In the end, we obtained 65,716 / 848 / 444 utterances as training / validation / test set, respectively. Additionally, we have 96 actions and 7 clarification response patterns in the training, 70 actions and 7 clarification response patterns in the validation, and 71 actions in test dataset. More detailed statistics can be found in Table 2.

5 Evaluation

5.1 Pointing Recognition

The results of the pointing recognition are depicted in Table 3. While *forearm* means we employed wrist and elbow points to draw the pointing line, *forefinger* indicates that we utilized two points on the forefinger as explained in Sect. 3.1. Besides, $+$ states that we include a margin around the line to capture nearby objects as well. For each sample, we evaluate the multiset of recognized pointing target labels $O = \{o_1, ..., o_N\}$ produced by each method using the following metrics, where g is the ground truth pointing target: *exact* match means $O = \{g\}$, *include*: $g \in O \wedge |O| > 1$, *others*: $g \notin O \wedge |O| \geq 1$ and *none*: $O = \varnothing$. See supplementary material C.1 for further explanations.

The results show that the index finger approach outperforms with an F_1-score of 56.9 % the forearm approach with an F_1-score of 42.6 % on our test set. Also, we can observe that using a margin leads to more object detections, which lowers the *exact* but increases the *include* score, i. e. it leads to a higher recall while increasing ambiguity. This again motivates the need for interactive dialog, with results of the combined system shown in Sect. 5.3. 21.43 % of the cases have

Table 3. Pointing component evaluation. Numbers in percent. See the text for details

Method	$F_1 \uparrow$	Exact ↑	Include ↑	Others ↓	None ↓
Forearm	42.6	6.59	32.42	43.96	17.03
Forearm +	50.1	8.24	37.92	38.46	**15.38**
Forefinger	56.9	**9.34**	42.85	30.77	17.03
Forefinger +	63.0	3.30	**54.39**	**26.92**	**15.38**
Human	68.7	68.7	–	–	–

object detection error which represents 39.79 % of the total failures. Further, we validate the pointing recognition regarding false positives using the set of 40 videos showing no pointing gesture (see Sect. 4.1). Here, our pipeline erroneously detects a pointing in only two cases, i. e. 38 videos are correctly classified as not showing a relevant gesture.

To assess the difficulty of our dataset and to provide an upper bound for model performance, we also measured human accuracy on the pointing demonstration videos. For this, we asked several subjects not involved in the project to watch the short (muted) video clips and write down an unambiguous expression identifying the object they think the pointing gesture targets. We then compared these annotations manually with the ground truth labels (as defined by the person performing the gesture in the video), where a match is not required to have the same wording, but should unambiguously refer to the same object. In this way, we avoid costly bounding box annotation. Human evaluation leads to an F_1-score of 68.7 %.

5.2 Multimodal Natural Language Understanding

The used datasets for the evaluation of the Multimodal Natural Language Understanding (NLU) component are described in Sect. 4.2.

We trained our model described in Sect. 3.2 three times and chose the model with the second best accuracy on the complete target of the validation dataset for this evaluation, to exclude outliers due to random effects, thus making it easier to reproduce the results. A beam size of 4 was used. For clarification responses where objects are enumerated, the order is ignored for evaluation.

As metrics, we calculated the accuracy for the complete outputs and individually for the predicted actions, first objects, and second objects. Actions means the correct action like “wash” in an API call for an unambiguous input or the correct clarification response for an ambiguous input. The first object is the first argument of the API call or the first placeholder group of a clarification response. Analogously, the second object refers to the second object of the API call or the second placeholder group of a clarification response. For each possible clarification, Table 1 shows what placeholder groups (*X* or *A*, or *A, B, ...*) occur. We calculated the accuracy by dividing the number of correct samples by the number of total samples, where the later takes into account that not every

Table 4. NLU evaluation on test_{ref}/test_{hyp}. The numbers report accuracy (acc.) in percent regarding the complete output, action, first and second object slot of the output, respectively. For test_{ref}, there are no ambiguous cases and thus no clarifications

Subset\acc. of	$test_{\text{ref}}$				$test_{\text{hyp}}$			
	Complete	Action	1st obj.	2nd obj.	Complete	Action	1st obj.	2nd obj.
All	57.7	68.9	78.4	58.2	54.1	78.6	51.9	50.0
Pointing?								
– yes	58.4	69.3	80.2	59.1	55.4	80.0	51.5	53.9
– no	50.0	65.0	60.0	37.5	40.0	65.0	55.0	21.1
Target								
– API call	57.7	68.9	78.4	58.2	33.9	69.8	48.3	40.2
– Clarification	—	—	—	—	78.2	89.1	60.8	63.2

clarification response has a first or second object and not every API call has a second object.

Test results are depicted in Table 4. Additional results including the evaluation of the validation set can be found in the supplementary material C.2. We evaluated not only for each complete dataset (*all*), but divided each two times into two different disjoint subsets. The first partition separates samples based on whether a pointing gesture was used or not (*pointing?*). The second criterion differentiates based on the output target, which can either be an API call or a clarification response (*target*).

The model has an accuracy of 57.7 % on the test_{ref} dataset and an accuracy of 54.1 % on the test_{hyp} dataset. The use of pointing gestures relatively improves the performance by 16.8 % (test_{ref}) and by 38.5 % (test_{hyp}).

5.3 End-to-End Results

Finally, we evaluated the overall interactive dialog system, including pointing recognition, multimodal NLU and clarification disambiguation. For conducting these end-to-end experiments, we took the output of the hand pose estimation model without the margin approach (forefinger) from Sect. 5.1 and the NLU model from Sect. 5.2. As the final output of the system (after disambiguation) is an API call, we report the same metrics as described in Setion 5.2. The results of evaluating on the human pointing dataset with human-annotated commands and clarification utterances are depicted in Table 5. When comparing to the results of pointing recognition on its own (Table 3), we can see that the interactive dialog significantly improved the performance, with 23.3 % of the API calls generated completely correct, where the performance of an exact match of pointing recognition itself is only at about 9.34 %. The accuracies of the NLU component separately evaluated, see Sect. 5.2, are higher, but in the experiments of the NLU component either a reference label for the object which is pointed at is used or no pointing gesture is involved. In the second case, it is assumed that

Table 5. End-to-End evaluation result of our dialog system on the human-annotated human pointing dataset (with pointing gesture). Metric is accuracy in percent, more explanations in Sect. 5.2

System	Complete	Action	First object	Second object
Without CLIP	13.4	29.5	18.8	16.1
With CLIP	23.3	51.2	33.4	27.6

the object grounding challenge is solved (which is not the case [40] and would hence decrease the accuracies). Table 5 demonstrates the impact of the CLIP disambiguation model, where *with CLIP* refers to the full system and *without CLIP* is the sole combination of pointing recognition and NLU component (i. e. no multi-turn dialog). If the NLU component detects an ambiguous case and outputs a clarification question, this is counted as wrong in the *without CLIP* case since the ambiguity cannot be resolved. This point leads to the low accuracies. Comparing the resulting numbers, we can again observe that the multi-turn dialog significantly improves the results on all evaluation metrics. This underlines the significance of multimodal ambiguity resolution and indicates the potential for further research in this area.

6 Conclusion and Discussion

In this paper, we presented a system combining pointing recognition with multimodal NLU to achieve an interactive, task-oriented dialog for robot command disambiguation. Our experimental results show that the proposed approach provides a flexible and intuitive way of communicating intentions via gestures and language and thus is a valuable contribution to HRI. In contrast to previous work [8], our dataset and system can handle both cases with and without pointing gestures and embraces ambiguity in both modalities, solving them through interactive dialog. To foster future research on this important aspect of HRI, we published our code, models and evaluation dataset.

For future work, we will improve individual components, e.g. NLU with larger models like T5-11b and more elaborated fine-tuning procedures. Moreover, if the NLU component detects a pointing phrase, it could overwrite errors in the pointing detection of the pointing recognition component. Efficient methods to estimate coordinates and pointing direction in 3D could further improve the current results. Moreover, we plan to deploy the developed system to a real humanoid robot to allow for end-user feedback.

A problem of our presented pipeline system is the accumulation of errors from each component (object recognition, hand pose estimation, recognition of start and end time of pointing gesture, NLU). To mitigate that, we could pass probabilities instead of final decisions from upstream to downstream components. For instance, this could carry uncertainties from object detection (apple vs. pear) on to the NLU component, where it might be resolved due to verbal clues ("please

give me that apple"). Furthermore, to eliminate error propagation completely, an end-to-end system could be developed.

Acknowledgements. This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the project OML (01IS18040A).

References

1. Anbarasan, Lee, J.S.: Speech and gestures for smart-home control and interaction for older adults. In: Proceedings of the 3rd International Workshop on Multimedia for Personal Health and Health Care, pp. 49–57. HealthMedia 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3264996.3265002
2. Asfour, T., et al.: Armar-6. IEEE Robotics & Automation Magazine. 1070(9932/19) (2019)
3. Azari, B., Lim, A., Vaughan, R.: Commodifying pointing in HRI: simple and fast pointing gesture detection from RGB-D images. In: 2019 16th Conference on Computer and Robot Vision (CRV), pp. 174–180. IEEE (2019)
4. Bolt, R.A.: "put-that-there": voice and gesture at the graphics interface. In: Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, pp. 262–270. SIGGRAPH 1980, Association for Computing Machinery (1980)
5. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
6. Bärmann, L., Peller-Konrad, F., Constantin, S., Asfour, T., Waibel, A.: Deep episodic memory for verbalization of robot experience. IEEE Robot. Autom. Lett. **6**(3), 5808–5815 (2021). https://doi.org/10.1109/LRA.2021.3085166
7. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)
8. Chen, Y., et al.: Yourefit: embodied reference understanding with language and gesture. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1385–1395, October 2021
9. Cosgun, A., Trevor, A.J., Christensen, H.I.: Did you mean this object?: Detecting ambiguity in pointing gesture targets. In: Towards a Framework For Joint Action Workshop, HRI (2015)
10. Damen, D.: Rescaling egocentric vision. Int. J. Comput. Vision **130**(1), 33–55 (2022)
11. Das, S.S.: A data-set and a method for pointing direction estimation from depth images for human-robot interaction and VR applications. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 11485–11491. IEEE (2021)
12. Desrochers, S., Morissette, P., Ricard, M.: Two perspectives on pointing in infancy. In: Joint Attention: its Origins and Role in Development, pp. 85–101 (1995)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short

Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1423
14. Dhingra, N., Valli, E., Kunz, A.: Recognition and localisation of pointing gestures using a RGB-D camera. In: Stephanidis, C., Antona, M. (eds.) HCII 2020. CCIS, vol. 1224, pp. 205–212. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50726-8_27
15. Holzapfel, H.: A dialogue manager for multimodal human-robot interaction and learning of a humanoid robot. Ind. Robot Int. J. **35**, 528–535 (2008)
16. Holzapfel, H., Nickel, K., Stiefelhagen, R.: Implementation and evaluation of a constraint based multimodal fusion system for speech and 3d pointing gestures. In: Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI) (2004)
17. Hu, J., Jiang, Z., Ding, X., Mu, T., Hall, P.: VGPN: voice-guided pointing robot navigation for humans. In: 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1107–1112 (2018). https://doi.org/10.1109/ROBIO.2018.8664854
18. Jaiswal, S., Mishra, P., Nandi, G.: Deep learning based command pointing direction estimation using a single RGB camera. In: 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp. 1–6. IEEE (2018)
19. Jevtić, A., et al.: Personalized robot assistant for support in dressing. IEEE Trans. Cogn. Dev. Syst. **11**(3), 363–374 (2019). https://doi.org/10.1109/TCDS.2018.2817283
20. Jocher, G., et al.: ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022. https://doi.org/10.5281/zenodo.6222936
21. Jojic, N., Brumitt, B., Meyers, B., Harris, S., Huang, T.: Detection and estimation of pointing gestures in dense disparity maps. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pp. 468–475. IEEE (2000)
22. Kehl, R., Van Gool, L.: Real-time pointing gesture recognition for an immersive environment. In: Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 577–582. IEEE (2004)
23. Kingma, D.P., Ba, J.: Adam : a method for stochastic optimization. In: Proceedings of the Third International Conference on Learning Representations (ICLR) (2015)
24. Lai, Y., Wang, C., Li, Y., Ge, S.S., Huang, D.: 3d pointing gesture recognition for human-robot interaction. In: 2016 Chinese Control and Decision Conference (CCDC), pp. 4959–4964. IEEE (2016)
25. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
26. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv:2107.13586 [cs] (2021)
27. Medeiros, A., Ratsamee, P., Orlosky, J., Uranishi, Y., Higashida, M., Takemura, H.: 3d pointing gestures as target selection tools: guiding monocular UAVs during window selection in an outdoor environment. ROBOMECH J. **8**(1), 1–19 (2021)
28. Medeiros, A.C.S., Ratsamee, P., Uranishi, Y., Mashita, T., Takemura, H.: Human-drone interaction: using pointing gesture to define a target object. In: Kurosu, M. (ed.) HCII 2020. LNCS, vol. 12182, pp. 688–705. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49062-1_48

29. Mesnil, G., et al.: Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(3), 530–539 (2015). https://doi.org/10.1109/TASLP.2014.2383614
30. Ni, J., Young, T., Pandelea, V., Xue, F., Adiga, V., Cambria, E.: Recent advances in deep learning based dialogue systems: a systematic survey. CoRR abs/2105.04387 (2021)
31. Nickel, K., Scemann, E., Stiefelhagen, R.: 3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In: Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 565–570. IEEE (2004)
32. Nickel, K., Stiefelhagen, R.: Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In: Proceedings of the 5th International Conference on Multimodal Interfaces, pp. 140–146 (2003)
33. Park, C.B., Lee, S.W.: Real-time 3d pointing gesture recognition for mobile robots with cascade hmm and particle filter. Image Vision Comput. **29**(1), 51–63 (2011)
34. Pateraki, M., Baltzakis, H., Trahanias, P.: Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation. Comput. Vision Image Underst. **120**, 1–13 (2014)
35. Pozzi, L., Gandolla, M., Roveda, L.: Pointing gestures for human-robot interaction in service robotics: a feasibility study. In: Miesenberger, K., Kouroupetroglou, G., Mavrou, K., Manduchi, R., Covarrubias Rodriguez, M., Penaz, P. (eds.) Computers Helping People with Special Needs. ICCHP-AAATE 2022. LNCS, vol. 13342, pp. 461–468. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08645-8_54
36. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR, 18–24 July 2021
37. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
38. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2017)
39. Rosen, E., Whitney, D., Fishman, M., Ullman, D., Tellex, S.: Mixed reality as a bidirectional communication interface for human-robot interaction. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11431–11438 (2020)
40. Sadhu, A., Chen, K., Nevatia, R.: Video object grounding using semantic roles in language description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10417–10427 (2020)
41. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: Inverted residuals and linear bottlenecks (2018)
42. Schauerte, B., Fink, G.A.: Focusing computational visual attention in multi-modal human-robot interaction. In: International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction. ICMI-MLMI 2010, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1891903.1891912
43. Schauerte, B., Richarz, J., Fink, G.A.: Saliency-based identification and recognition of pointed-at objects. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4638–4643 (2010). https://doi.org/10.1109/IROS.2010.5649430
44. Shan, D., Geng, J., Shu, M., Fouhey, D.: Understanding human hands in contact at internet scale. In: CVPR (2020)

45. Showers, A., Si, M.: Pointing estimation for human-robot interaction using hand pose, verbal cues, and confidence heuristics. In: Meiselwitz, G. (ed.) SCSM 2018. LNCS, vol. 10914, pp. 403–412. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91485-5_31
46. Shukla, D., Erkent, O., Piater, J.: Probabilistic detection of pointing directions for human-robot interaction. In: 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8. IEEE (2015)
47. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1145–1153 (2017)
48. Stiefelhagen, R., Fugen, C., Gieselmann, R., Holzapfel, H., Nickel, K., Waibel, A.: Natural human-robot interaction using speech, head pose and gestures. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), vol. 3, pp. 2422–2427 (2004). https://doi.org/10.1109/IROS.2004.1389771
49. Stiefelhagen, R., et al.: Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. IEEE Trans. Robot. **23**(5), 840–851 (2007). https://doi.org/10.1109/TRO.2007.907484
50. Tomasello, M., Carpenter, M., Liszkowski, U.: A new look at infant pointing. Child Dev. **78**(3), 705–722 (2007)
51. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates, Inc. (2017)
52. Weld, H., Huang, X., Long, S., Poon, J., Han, S.C.: A survey of joint intent detection and slot filling models in natural language understanding. ACM Comput. Surv. **55**, 1–38 (2022). https://doi.org/10.1145/3547138
53. Winograd, T.: Understanding natural language. Cogn. Psychol. **3**(1), 1–191 (1972). https://doi.org/10.1016/0010-0285(72)90002-3
54. Woods, W., Kaplan, R., Nash-Webber, B.: The lunar sciences natural language information system. Final Report 2378, Bolt, Beranek and Newman Inc., Cambridge, MA (1974)
55. Zlatintsi, A., et al.: I-support: A robotic platform of an assistive bathing robot for the elderly population. Robot. Autonom. Syst. **126**, 103451 (2020). https://doi.org/10.1016/j.robot.2020.103451