Using Natural Language Feedback in a Neuro-inspired Integrated Multimodal Robotic Architecture

Johannes Twiefel, Xavier Hinaut, Marcelo Borghetti, Erik Strahl, Stefan Wermter

Using Natural Language Feedback in a Neuro-inspired Integrated Multimodal Robotic Architecture

Johannes Twiefel[1], Xavier Hinaut[1,2], Marcelo Borghetti,[1] Erik Strahl[1] and Stefan Wermter[1]

***Abstract*— In this paper we present a multi-modal human robot interaction architecture which is able to combine information coming from different sensory inputs, and can generate feedback for the user which helps to teach him/her implicitly how to interact with the robot. The system combines vision, speech and language with inference and feedback. The system environment consists of a Nao robot which has to learn objects situated on a table only by understanding absolute and relative object locations uttered by the user and afterwards points on a desired object to show what it has learned. The results of a user study and performance test show the usefulness of the feedback produced by the system and also justify the usage of the system in a real-world applications, as its classification accuracy of multi-modal input is around 80.8%. In the experiments, the system was able to detect inconsistent input coming from different sensory modules in all cases and could generate useful feedback for the user from this information.**

I. INTRODUCTION

In human robot interaction (HRI), substantial effort is spent on improving sensory classifiers which provide information to trainable knowledge databases. By employing state-of-the-art sensory classifiers it is still not guaranteed that the given input can be processed and learned, as there is still the possibility of incorrect input generated by the user. Also, when using a multimodal architecture, the inputs provided to the system can be inconsistent, possibly by incorrect input coming from the user or a misclassification performed by a classifier. To implicitly train the user how to provide consistent input to the system, inconsistent inputs have to be classified as such and feedback has to be provided to change the user's way of teaching. Inconsistent inputs are e.g. teaching a robot that an object is at a specific position while it is in fact not present. Also, the user could confuse his egocentric perspective with the intrinsic perspective of the robot, which means confusing "left" and "right", which is also a common problem in human-human interaction. We state that this kind of input information should not only be rejected, but also used to provide feedback to the user what kind of inconsistency occurred, so that the user is able to change his strategy of teaching.

In this paper we present a multi-modal architecture which employs vision, language and speech, to teach a robot in a home scenario. Our system also contains a novel rule-based inference system which can identify the source of an inconsistent situation and provide feedback to the user. Also, the robot possesses a trainable knowledge base which it can consult to perform different actions and learn different objects by referring to other objects it knows already.

The study of Vollmer et al. [1] also discusses the influence of feedback provided by a robot to a human tutor. In their scenario a robot is instructed to imitate or emulate human actions which consist of moving different objects on a table; the human tutor and the robot sit together at a table like in the given study. The results of the study indicate that the feedback provided by the robot directly influences the future way of teaching. In contrast to our study, no learning is involved, which leads to a stateless communication on the robot's side, while the feedback of our robot is also influenced by the knowledge the robot gained during the dialog. Also, in that study only a unimodal (visual) input is provided to the system, so there is no requirement for identifying inconsistent inputs coming from different modalities.

II. SCENARIO

In this paper our system is tested in an HRI scenario containing a table, which is divided into the absolute positions (*left*, *right* and *middle*). Our robot (a Nao) is situated on one side of the table, the user on the other side to create a conversational atmosphere. The goal of the user is to help the robot to attach labels to objects (*banana*,*box*,*cup*, see Fig. 1) which are positioned on the table. In the beginning, the robot can only distinguish between different objects, but has no labels (like "$banana$") for them. The user can employ semi-free speech to teach the robot by providing information about the relative or absolute position of the object on the table. To test if the robot actually learnt, the user can instruct the robot to point to a specific object it has learnt. Fig. 1 shows the scenario.

III. APPROACH

Our integrated system consists of two modalities, which are speech and vision. The task that the system can perform is mapping objects to names. At the beginning, the system is initialized with the two sensory modules being trained independently. The visual system is trained on a set of objects which it can distinguish between, but cannot map

*This work was partially supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme: EchoRob project (PIEF-GA-2013- 627156).

[1]Johannes Twiefel, Xavier Hinaut, Marcelo Borghetti, Erik Strahl, and Stefan Wermter are with Knowledge Technology Group, Department Informatik, Universität Hamburg, 22527 Hamburg, Germany {twiefel, hinaut, borghetti, strahl, wermter}@informatik.uni-hamburg.de

[2]Xavier Hinaut is with Inria Bordeaux Sud-Ouest, Talence, France; LaBRI, UMR 5800, CNRS, Bordeaux INP, Université de Bordeaux, Talence, France; and Institut des Maladies Neurodégénératives, UMR 5293, CNRS, Université de Bordeaux, Bordeaux, France xavier.hinaut@inria.fr

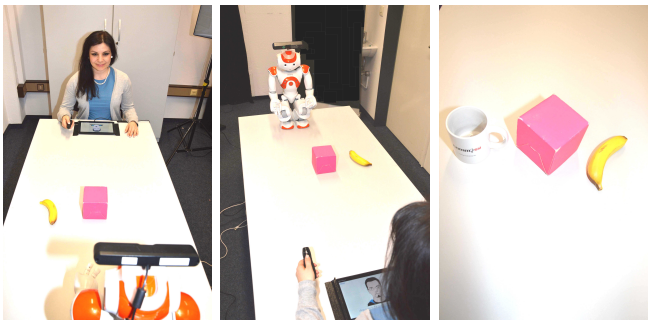

Fig. 1. The left images show a user communicating with the robot, the right image depicts the three objects (*cup*, *box*, *banana*).

these objects to lingual identifiers. Also, the visual system can determine the position of the recognized objects. The speech recognition system is able to recognize domain-specific utterance and to create word representations from them. To interpret these sentence hypotheses, a natural language processing system is trained to create machine-readable predicate representations. To process the predicates and the position of the recognized objects, a rule-based inference system was developed which is able to detect inconsistent information provided by the sensory module and so can provide detailed feedback for the user. If the input information is consistent with the context, the information is added to the knowledge base of the robot. By triggering specific commands, the robot can be encouraged to use the knowledge it has to provide information to the user by giving information about a specific object position the user asked for. Fig. 2 contains the architecture of our integrated system.

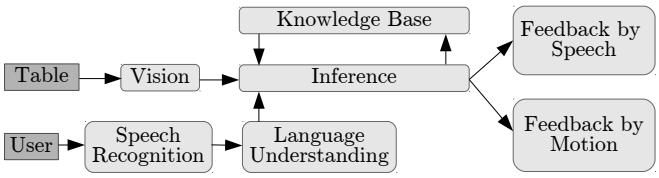

Fig. 2. The architecture of the system, showing the different modules and input modalities. Arrows indicate the flow of information. Vision and speech recognition (via language understanding) provide information to the inference module, which uses this information to verify the consistency of the inputs, while using previously learned knowledge from the knowledge base if necessary. Feedback is provided via motion or synthesized speech. While only consistent information is stored in the knowledge base, inconsistent input is rejected and the user is informed about the reasons using the feedback module.

A. Vision Module

To perform the classification of the objects, we first segment them from the environment, since this produces distinct clusters of points with reduced noise. For simplicity, we are considering that the objects are on plane surfaces such as table, floor or wall. The environment is represented as a point cloud captured by an RGB-D device. To select the objects we take into account the fact that the z axis points to the direction of the objects. According to [2], for segmentation the scene is reoriented in a way that the z axis

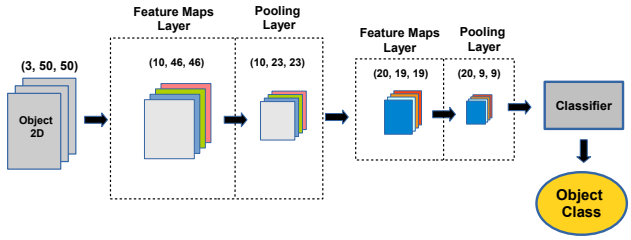

Fig. 3. Convolutional Neural Network Architecture composed of two convolutional layers, each followed by a pooling layer (max pooling). The tuple (C, w, h) on the top of each image is composed of C channels (3 for RGB), width w and height h. The convolutional layers extract features from the images and combine them in posterior layers. The max pooling layer is used for scale and translation invariance. In this figure, the convolutional filters employed have sizes $(5, 5)$ for and the pooling filters employed have sizes $(2, 2)$.

becomes orthogonal to the normal vector of the plane. Thus, we identify all objects that have y coordinate values higher than the average value of the y coordinate of the points that compose the plane. Finally, we consider only objects located within a *tolerance distance* from the center of mass of the segmented plane.

To classify the objects, a Convolutional Neural Network [3] is employed (see Fig. 3), because this model already performed well in object recognition tasks. To train the neural network, we captured samples of isolated objects and segmented them from the environment according to the approach described earlier. To increase the number of samples we rotated the objects in different orientations. We applied to each sample 5 rotations in the x axis and 10 rotations in the y axis. These numbers of rotation in x and y were defined to simulate different viewpoints from the top. The sequence of rotations around the y axis can also be seen as sequential captures around the object when the camera is moving clockwise or counter-clockwise.

All the samples produced are then projected into the x-y plane and are used to train the neural network. The output of the convolutional layers is a vector of features that are used as input to a multi-layer perceptron. We are using the same parameters described in [2]. The neural network is trained using *Backpropagation Algorithm*. The whole architecture can be seen in Figure 3. We are using two feature maps layers followed by pooling layers. The size of the input image is 50×50 pixels. For the feature maps layer we use a filter of size 5×5 and for the pooling we use a filter of size 2×2. In the first layer, 20 feature maps are generate and in the second layer 10 feature maps are generated. Different values for the number of feature maps, size of the filters, and internal parameters of the multi-layer perceptron were tested and we selected the parameters that performed best.

B. Speech Recognition Module

As we are working in a specific scenario, our speech module is based on a domain-dependent approach (DOCKS) developed by Twiefel et al. (2014) [4] which can be restricted to a scenario and improve the recognition accuracy, because less errors are possible. The approach employs

Google's speech recognition system [5], which is domain-independent and optimized for web-searches and dictation tasks and employs a post-processing step to restrict the produced text hypotheses to the given scenario. A list of possible sentences the user could utter is provided to the system, containing sentences like "this is the box", "the box is left to the cup", "show me the banana", etc. The text hypotheses from Google are converted to phoneme sequences by consulting a pre-trained grapheme-to-phoneme converter [6], as the pronunciation of the hypotheses may be similar to the given set of expectable sentences while the grapheme representation is not. To measure the distance between the Google hypotheses (we take the 10-best) and the expectable sentences, we employ the Levenshtein distance [7]:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases} \quad (1)$$

with a being a phoneme sequence from Google's hypotheses and b being a phoneme sequence from the predefined expectable sentence list. We extend this distance measure by adding a confidence value c using

$$c = \max(0, 1 - \min(lev_{a_g,b_s}(|a_g|, |b_s|)/|b_s|)) \quad (2)$$

with a_g being the g-th phoneme sequence of Google's hypothesis and b_s being the s-th phoneme sequence of the sentence list. We also developed an Android application which performs the speech recognition and replaced the PC version from before [4]. The system does not need any training, we only provide a list of sentences. To extend the list with new objects, we generate sentences containing these objects with a grammar-controlled sentence generator.

C. Language Understanding Module

The language module (*θ-RARes*) has been adapted from previous experiments on a neuro-inspired model for sentence comprehension using recurrent neural networks [8] and its application to HRI [9]. This model is based on an Echo State Network [10] (ESN) with leaky integrator neurons.

The model (see Figure 4) is trained to learn the mapping of the semantic words (SW) (e.g. nouns, verbs) of a sentence onto the different slots (the thematic roles: e.g. action, location) of a basic event structure in a predicate form like *action(object, location)*. This predicate representation enables to easily integrate this model into a robotic architecture [9].

As depicted in Figure 4, the system processes inputs as follows: from a sentence (i.e. sequence of words) as input (*upper left*) the model outputs (*middle right*) a predicate that can be post-processed by the system. The output predicate can represent a command (e.g. "Show me the banana" $\rightarrow$ $show(banana)$) or an information on the state of the world (e.g. "The banana is in the middle" $\rightarrow$ $middle(banana)$). Before entering the recurrent neural network, words are preprocessed. This preprocessing transforms the sentence

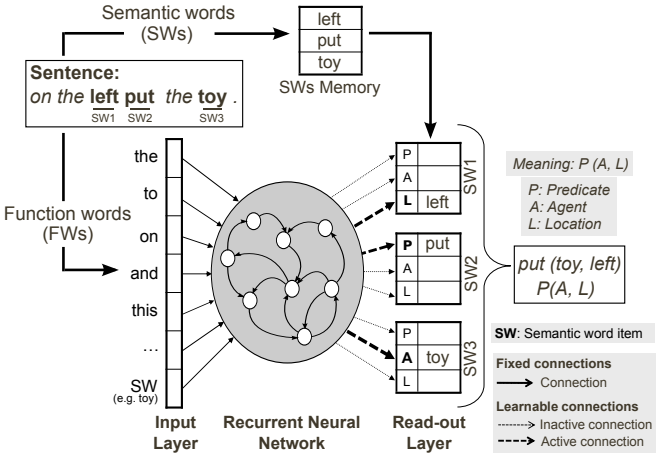

Fig. 4. The *θ-RARes* language module. Sentences are first transformed into a sentence structure, i.e. all semantic words (SW) are replaced by an SW marker. The ESN (i.e. reservoir) is given the sentence structure word by word. Each word activates a different input unit. During training, the connections to the readout layer are modified in order to learn the mapping between the sentence structure and the predicate-argument meaning. After training, the most active units are bound with the SW kept in the SWs memory to form the resulting predicate. Figure adapted from [9].

into a sentence structure (or *grammatical construction*): SW, i.e. nouns and verbs that have to be assigned a thematic role, are replaced by the *SW* item. The processing of the grammatical construction is sequential: words are given one at a time. The final estimation of the thematic roles for each SW are read-out at the end of the sentence.

The model processes grammatical constructions, not sentences *per se*, thus permitting to bind a virtually unlimited number of sentences on these sentence structures. Based only on a small training corpus this enables to process future sentences with currently unknown SW. Therefore, it is also suited for modelling developmental language acquisition [11], [12], [13]. Here are some input/output transformations that the language module performs. For the given "input sentences", we provide the corresponding $\rightarrow$ *output predicates*:

- "This is the banana" $\rightarrow$ $this(banana)$
- "Show me the banana" $\rightarrow$ $show(banana)$
- "The banana is in the middle" $\rightarrow$ $middle(banana)$
- "The banana is on the right of the cup" $\rightarrow$ $right(banana, cup)$
- "Right of the cup is the banana" $\rightarrow$ $right(banana, cup)$

In these examples one can see that the system can robustly transform different types of sentences using a simple predicate representation. The last two sentences are quite different (the order of words is different) but the system can learn to provide an identical predicate representation. Hereafter are two sentences implying the execution of two consecutive actions: the order of the predicates indicates the order in which the actions have to be performed:

- "First show me the banana and then point to the box" $\rightarrow$ $show(banana); point(box)$
- "Before pointing to the box show me the banana" $\rightarrow$ $show(banana); point(box)$

In the last sentence, one can see that the system learns to produce the correct chronological order of actions, even if the order of the SW are very different. Thus, the operations performed by the neural network could not be reduced to a simple set of rules based on the order of the words in the sentence. One major advantage of this neural network language module is that no parsing grammar has to be defined a priori: the system learns only from the examples given in the data. Another interesting aspect of the system is that the language model does not need to be retrained, it is only trained once during the initialization. Moreover, testing a new sentence is computationally fast (linear to the number of words in the sentence). The language module architecture is flexible enough to allow incremental learning (some preliminary work has been done in this direction [12]) thus enabling the system to learn during short to long periods of execution. The module has other capabilities such as learning several languages at the same time, and correctly processing sentences with out-of-vocabulary words [13]. From the HRI point of view, the aim of using this neural network based model is to gain adaptability because the system is trained on examples (no need to predefine a parser for each language), to be able to process natural language sentences instead of stereotypical sentences (i.e. "put cup left"), and to be able to generalize to unknown sentences (not in the training data set). Moreover, this model seems quite flexible when changing the output predicate representations [8]. From the computational neuroscience and developmental robotics point of view, the aim of having an architecture working with robots is to use them to model and test hypotheses about child learning processes of language acquisition [11]. Another benefit of using an ESN-based model is the constant execution time so that even long sentences can be processed in real-time.

D. Knowledge Representation and Database

For the given scenario, the knowledge representation of the system is an abstract representation of the table. For this, the table is separated into three positions, called *left*, *right* and *middle*, which is seen from the robot's intrinsic perspective. For the given scenario, the table representation looks like this: $table = (l, m, r)$ where $l, m, r \in P$ with $P = \{\text{“}banana\text{”}, \text{“}box\text{”}, \text{“}cup\text{”}, \text{“}unknown\text{”}, \text{“}empty\text{”}\}$. Also, there is a map projecting object labels to object indices: $t \rightarrow i$ with $i \in \{-1, 0, 1, 2\}$ and $t \in T$ and $T = P \setminus \{\text{“}empty\text{”}\}$. This map is filled by teaching the robot and is empty in the beginning. The representation $(\text{“}banana\text{”}, \text{“}unknown\text{”}, \text{“}empty\text{”})$ means that there is a *banana* in the *left* position, a yet *unknown* object in the *middle* and no object in the *right* position. The mapping $\text{“}banana\text{”} \rightarrow 0$ means, that the object identified with index 0 by the vision module was mapped to the label $\text{“}banana\text{”}$.

E. Inference Module

To ensure that the knowledge database is always kept consistent, we propose a rule-based inference module which is able to identify and reject inconsistent information coming from the sensory modules while accepting valid inputs. I.e. the system receives the following information from the vision module: $(-1, 0, 1)$, the module converts this information to a table representation by converting the index -1 to $\text{“}empty\text{”}$ and replacing the other indices by the labels it has in the database (in this case it has $\text{”}banana\text{”} \rightarrow 0$) and the rest with the $\text{“}unknown\text{”}$ tag which results in the following table representation: $(\text{“}empty\text{”}, \text{“}banana\text{”}, \text{“}unknown\text{”})$. In this case, it receives the predicate $right(banana, box)$, and evaluates the validity of the predicate by determining the position of the reference object (*banana*), afterwards identifying the referred relative position (the one *right* to the *banana*). Then, the module checks if there is an $\text{“}unknown\text{”}$ object in this position; if this is the case, it learns the projection $\text{“}box\text{”} \rightarrow 1$ and will label index 1 with the label $\text{“}box\text{”}$ in the future. The new table representation looks like this: $(\text{“}empty\text{”}, \text{“}banana\text{”}, \text{“}box\text{”})$. As this mapping is independent from the position of the objects, an object identified as 1 will always be labelled as a $\text{“}box\text{”}$ even if it is detected in a different position of the table in the future. If, i.e. the system receives the information that the object in the *middle* is a *cup*, and the table representation derived from the vision module looks like this: $(\text{“}banana\text{”}, \text{“}empty\text{”}, \text{“}unknown\text{”})$, the inference module will reject the information as there is no object located in the *middle* of the table. This information can be used to provide feedback to the user and encourage him to change the instruction he uttered to the system. The following list of cases among others can be identified and converted to useful feedback to the user.

- $middle(cup), (\text{“}banana\text{”}, \text{“}empty\text{”}, \text{“}unknown\text{”})$ $\rightarrow$ "no object in position"
- $middle(banana), (\text{“}banana\text{”}, \text{“}unknown\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "banana already known"
- $this(banana), (\text{“}unknown\text{”}, \text{“}unknown\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "multiple unknown objects"
- $right(cup, box), (\text{“}banana\text{”}, \text{“}cup\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "object known as banana"
- $right(banana, box), (\text{“}banana\text{”}, \text{“}cup\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "described position outside of table"
- $left(banana, box), (\text{“}unknown\text{”}, \text{“}cup\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "reference object unknown"
- $show(banana), (\text{“}unknown\text{”}, \text{“}cup\text{”}, \text{“}empty\text{”})$ $\rightarrow$ "no banana on table"

The cases of inconsistency are then transformed to natural language, i.e. "I know what a banana is but I cannot see one on the table" or "I do not know what a cup is, you have to teach me first". Also, the feedback system is supported by arm motion, as the robot is pointing to the position the new object should be located in, to help the user to correct errors he/she made by confusing his/her intrinsic perspective of the robot with his intrinsic perspective. Also, the inference module is able to perform mutual exclusion. For the given situation $this(cup), (\text{“}banana\text{”}, \text{“}box\text{”}, \text{“}unknown\text{”})$, it identifies the unknown object correctly and maps the label $\text{“}cup\text{”}$ to it, as this is the only object that could be meant. Also, the module is able to perform two commands in a row, as pointing to one object first and to another one afterwards.

IV. Study Design and Experiments

To measure the degree of usefulness of the feedback provided by our inference module, we performed a user study with 12 participants, who had to perform four tasks using the system. The following list contains the table situation and the task:

1) (“empty”,“box”,“banana”) - “Teach the robot what a banana is and make it show it to you.”
2) (“cup”,“box”,“empty”) - “Teach the robot what a cup and a box are and make it first show you the cup and then show you the box.”
3) (“empty”,“banana”,“cup”) - “Teach the robot what a cup is and make it show it to you.”
4) (“box”,“banana”,“empty”) - “Teach the robot what a box and a banana are and make it first show you the box and then show you the banana.”

The participants had 5 minutes time per task and were restricted to use a fixed list of commands:

- show me the *dog*
- this is the *dog*
- the *dog* is left of the *cat*
- the *dog* is right of the *cat*
- the *dog* is in the middle
- first show me the *dog*, then show me the *cat*

The words *cat* and *dog* were placeholders for the object names, to reduce the bias.

A. Study Design - Degree of Usefulness

As mentioned in section III-E, the inference module is able to detect inconsistent input and provide user feedback. To test the degree of usefulness, we divided the participant group into two subgroups. Two different variants of our system were used, the *verbose* (V) system, which provides feedback to the user and the *non-verbose* (NV) variant which only provides information if a given command was accepted or rejected. The V group performed task 1 and 2 with the V system, task 3 and 4 with the NV system. The NV group had to use the NV system for task 1 and 2 and the V system for task 3 and 4. Both groups had to use both systems (V and NV) to show that participants using the NV variant first are still able to learn how to use the system faster when using the V variant. Also, the V group did not receive feedback in task 3 and 4 to show that they already learned how to use the system in the previous tasks. We measured the number of commands and the time needed to perform each task. Additionally, we provided a survey containing questions which could be answered by giving numbers between 1 and 10, i.e. “The feedback of the verbose system helped me to faster understand how to instruct the robot.” “1: not at all” “10: definitely”

B. Experiment - System Performance

To be able to provide feedback for inconsistent inputs, an input has to be classified correctly, even if the outputs cannot be used to add new information to the knowledge base or perform a desired action. Accordingly, we measured the performance of all sensory modules, the inference module and also the user. The performance was measured while the participants used the system to perform their tasks.

V. Results

The participants needed 354 commands in total to perform the tasks. The speech module achieved an accuracy of 92.4%, together with the language module, which only failed when the speech module produced an incorrect hypothesis. The vision module attained a classification accuracy of 87.3%, which results only of not recognizing an object at all, but never confusing different objects. The accuracy of the multi-modal integration is 80.8%, which means that vision, speech and language module performed a correct classification. 52.5% of the instructions produced by the user were inconsistent with the context, and could be identified as this by the inference module. For all inconsistent cases, the correct feedback could be created by the inference module, although it only got provided to the user in the V variant of the system. 75% of the users confused the perspective they had to use to describe the relative position of an object, which means, that an object is i.e. left of another object from the intrinsic perspective of the robot. The results in Fig. 5 show that the V group needed less instructions and time for task 1 and 2, which is also supported by a one-tailed, paired t-test, with $p = 0.029$ for time and $p = 0.011$ for instructions. Also, the V group needed significantly less instructions ($p = 0.001$) and time ($p = 0.005$) for all tasks.

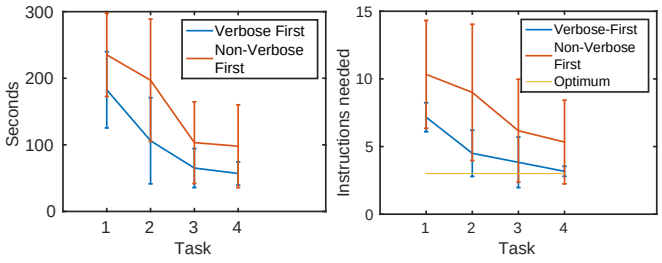

Fig. 5. The graphs show the performance of the users using the verbose and non-verbose system first. The average number of instruction (right) and the average time needed (left) to perform each task was measured.

VI. Discussion

It took the participants some time to understand that there are relative (*left*, *right*) and absolute (*middle*) descriptions of object locations. The only way to perform a task was first to teach the object in the middle, as this is the only absolute description possible using the provided commands, then relatively describing the object left or right to it. Participants often tried to refer to the object in the middle, which was not working, as the robot does not know any object labels in the beginning. Another difficulty was the correct usage of “left of” and “right of”, where the participants described the object from the wrong perspective. By providing motion feedback and pointing to the empty position, the participants immediately learned that they have to describe the scene from the robot’s intrinsic perspective. The feedback helped the user in all cases to understand immediately how to use

relative descriptions, while for the NV variant it took the user more trials to find out the correct perspective. The results show that participants using the V variant first, learned faster to use the system correctly and needed less instructions to perform the task. The minimum possible number of instructions needed to complete a task was 3, and while performing task 3, the V group already had learned how to use the system and did not need feedback anymore, while the NV group, which started to use the V variant in task 3, could use the feedback to perform the task quickly and with less instructions. The survey revealed that all participants found the feedback system useful in some way and determined an improved usability. They also noted that they were able to understand how to use the robot faster in most cases.

VII. CONCLUSION

The performance measured in the user study clearly shows that the system can be used in a real-world robot scenario. The results measured by the feedback module justify the development and integration of a feedback providing inference module. Instead of only providing information about success or failure of processing an instruction, the user can learn implicitly to use the system correctly by interacting with the robot in a natural dialog. The results of the survey emphasize the value of the given module and we suggest to include a feedback provider in other HRI scenarios, too. Also, the user could be encouraged to change his/her strategy by providing detailed feedback much faster than by only providing feedback about error or success of a given instruction. This behaviour was supported by the fact that no error occurred during the recognition but the user provided inconsistent input in a given situation. Our findings support the hypothesis of Vollmer et al. [1] (see Sec. I) that feedback coming from the robot directly influences the future way of teaching of the user, and also indicate that this hypothesis is true for multi-modal scenarios with possibly inconsistent inputs. Also, it is important to produce correct outputs for inconsistent input to be able to provide feedback, as 52.5% of the instructions provided by the user were incorrect. The feedback module could provide correct feedback in all cases if the inputs were classified correctly (80.8%), which shows that a high classification accuracy directly influences the feedback, which then influences the performance of the user. Due to the fact that all modules are loosely coupled, it is easy to extend the system with other objects, other tags or new phrase structures and predicates. At the moment, the system is not able to perform online learning in the vision module, it is only able to recognize pre-trained objects, and only the names are attached to them by the learning process. If new objects are added, only the vision module has to be retrained. To add new labels, new sentences containing new labels have to be created for the speech module, which can be performed easily by a grammar generator. As the language modules only relies on the phrase structure and is independent from the entities itself, it only has to be retrained when new phrase structures have to be detected. Also, the system could be used in a more complex scenario using continuous instead of discrete location representations. For this, the knowledge representation and the inference module have to be adjusted, which can be done easily. We plan to extend the system to process completely free speech, by using an n-gram-based post-processing model which allows unknown words [4]. Also, the vision model could be used to train the new object representations online. We also plan to employ the speech and language module for a different language, which was already shown by Hinaut et al. [13]. To illustrate how the robot interaction works a video can be seen at `https://youtu.be/FpYDco3ZgkU` [14] with additional information on the different modules.

ACKNOWLEDGMENT

We thank Francisco Cruz and Nilay Avşar for their help in the study and all participants who took part.

REFERENCES

[1] A.-L. Vollmer, M. Mühlig, J. J. Steil, K. Pitsch, J. Fritsch, K. J. Rohlfing, and B. Wrede. Robots show us how to teach them: Feedback from robots shapes tutoring behavior during action learning. *PloS one*, 9(3):e91349, 2014.

[2] M. B. Soares, P. Barros, G. I. Parisi, and S. Wermter. Learning Objects from RGB-D Sensors Using Point Cloud-based Neural Networks. In *ESANN 2015*, pages 439–444, 2015.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[4] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter. Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing. In *Twenty-Eighth AAAI. Québec City, Canada*, pages 1529–1535, 2014.

[5] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope. "your word is my command": Google search by voice: A case study. In *Advances in Speech Recognition*, pages 61–90. Springer, 2010.

[6] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

[7] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics – Doklady*, 10(8):707–710, 2 1966.

[8] X. Hinaut and P. F. Dominey. Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. *PloS one*, 8(2):e52946, 2013.

[9] X. Hinaut, M. Petit, G. Pointeau, and P. F. Dominey. Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Frontiers in Neurorobotics*, 8, 2014.

[10] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34, 2001.

[11] M. Tomasello. *Constructing a language: A usage based approach to language acquisition*. Cambridge, MA: Harvard University Press, 2003.

[12] X. Hinaut and S. Wermter. An incremental approach to language acquisition: Thematic role assignment with echo state networks. In *ICANN 2014*, pages 33–40. Springer, 2014.

[13] X. Hinaut, J. Twiefel, M. Petit, P. Dominey, and S. Wermter. A recurrent neural network for multiple language acquisition: Starting with english and french. In *NIPS 2015 Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*, 2015.

[14] X. Hinaut, J. Twiefel, M. Borghetti Soares, P. Barros, L. Mici, and S. Wermter. Humanoidly speaking – learning about the world and language with a humanoid friendly robot. In *Video competition, IJCAI, Buenos Aires, Argentina*. https://youtu.be/FpYDco3ZgkU, 2015.