Intent Recognition From Speech and Plan Recognition

Michele Persiani and Thomas Hellström

Umeå University, Umeå, Sweden
michelep,thomash@cs.umu.se

Abstract. In multi-agent systems, the ability to infer intentions allows artificial agents to act proactively and with partial information. In this paper we propose an algorithm to infer a speakers intentions with natural language analysis combined with plan recognition. We define a Natural Language Understanding component to classify semantic roles from sentences into partially instantiated actions, that are interpreted as the intention of the speaker. These actions are grounded to arbitrary, hand-defined task domains. Intent recognition with partial actions is statistically evaluated with several planning domains. We then define a Human-Robot Interaction setting where both utterance classification and plan recognition are tested using a Pepper robot. We further address the issue of missing parameters in declared intentions and robot commands by leveraging the *Principle of Rational Action*, which is embedded in the plan recognition phase.

Keywords: Intent Recognition, Plan Recognition, Natural Language Understanding, Semantic Role Labeling, Algorithms

1 Introduction

Intent recognition has been recognized as a crucial task in past and recent research in cybernetic systems [15][7][14], especially when humans are teaming along with artificial agents [5]. The ability to predict other agents' future goals and plans allows for proactive decisions, and relates to several system requirements, such as the need of an enhanced collaboration mechanism in human-machine interactions, the need for adversarial technology in competitive scenarios, ambient intelligence, or predictive security systems [15][5]. In this paper we focus on intent recognition for robotics, in scenarios where a person and a robot are present, yet the results have a broader applicability.

In robotics, the ability to predict users enables proactive behavior, ultimately giving the robots the ability to understand and coordinate actions with their users, even when only partial information is given [19][5]. In this paper we propose a method to infer user intent from speech, which is often a preferred mode of interaction in human-robot interaction [16]. Firstly, a series of utterances by the user are classified into partially instantiated PDDL [10] actions by using

Semantic Role Labeling [8]. The actions are then grounded into PDDL planning instances, and the user's intent is inferred using a plan recognition algorithm constrained to consider only plans containing the classified actions. The proposed method allows for discovery of intents beyond the scope of single sentences (achieved through, for example, a shallow classification of a sentence), by computing intents contextually to the task domain, in the form of a goal and a plan. Being able to reason on goals and plans using also context variables is necessary when attempting to describe or infer an agent behavior [14][5].

The rest of the paper is organized as follows. In Section 2 we introduce give background and related work for our proposed algorithm. Section 3 describes the intent recognition algorithm, followed by Section 4 in which we evaluate an implementation of the algorithm, both statistically by testing it in different planning domains, and experimentally using a Pepper robot. Finally, in Section 5 we give some conclusive remarks.

2 Background

In robotics, intent recognition can be performed using several modalities, such as video [9], gestures [12], eyeball movements, affect information in speech [4], and speech. When inferring intentions, raw input must be first transformed into data structures that are suitable for inference, such as action frames [2][18]. We refer to this as the process of grounding to the task domain. After grounding, various inference tools can be applied. For example, in [3], utterances are processed by mapping semantic roles into ad-hoc action frames using machine learning techniques. Semantic frames, such as the ones described in FrameNet [1], can be transformed to robot actions using sets of lexical units [18]. These units connect grammatical relations found in sentences to the different frame elements. With this approach, all core arguments must be present for the frames to be utilizable.

Pre-trained language models can also be utilized when inferring intentions from speech. Chen et al. [6] map semantic frames to robot action frames by using a language model trained on semantic roles, showing how large language models can be used to obtain the likelihood for the frames arguments. Their proposed *Language-Model-based Commonsense Reasoning* (LMCR) assigns a higher probability to the instruction *"Pour the water in the glass."* than to *"Pour the water in the plate."*. Thus, when the planning component is searching for an object to pour water into, it will prefer a glass rather then other objects. The LMCR is used to rank candidates for complete action frames by testing the different combinations of the available objects.

Inferred actions typically must have all arguments specified before they can be part of an executable plan. However, we can usually not expect all parameters to be fully specified in user utterances. In this regard our approach stands in contrast with other solutions (e.g. [6][17]) where the possible combinations of objects are exhausted or searched to retain only the most likely combination as candidate arguments. We instead allow for missing arguments to be present in the action frames, leveraging then the planner to infer them as the arguments

that would allow the whole inferred intention to be the least costly. Intentions are thus infused at parameter level with the *principle of rational action* i.e. intentional agents prefer optimal plans when evaluating different alternatives [19].

3 Method

We formally define an agent's intention as a goal $\hat{g}$ together with an action plan $\hat{\pi}$ the agent is committed to while pursuing $\hat{g}$ [19]. The sequence of actions $\hat{\pi}$ can either be a complete plan achieving $\hat{g}$ or a partial plan directed towards it. Intent recognition thus becomes the task of inferring $\hat{g}$ and $\hat{\pi}$ from a set of observations $o \in O$:

$$\hat{g}, \hat{\pi} = \underset{g \in G, \pi \in \Pi}{\operatorname{argmax}} P(g, \pi | o), \tag{1}$$

where G and Π are the set of possible goals and the set of possible partial plans respectively, O is the set of possible sets of observations. $\hat{g}$ and $\hat{\pi}$ are the arguments that maximizes the likelyhood of the intent recognition model $P(G, \Pi | O)$.

We additionally introduce an explicit grounding model $P(A|O)$ that is used to map raw observations to the task space as grounded actions. Furthermore, we add the assumption that the inferred plan is independent of the observations given the set of grounded actions $a \in A$. The formulation of the intent recognition model becomes:

$$P(G, \Pi | O) = \sum_{A} P(G|\Pi) P(\Pi|A) P(A|O) P(O). \tag{2}$$

Hence, a partial plan for the agent is first inferred from the grounded observations. Then, the plan is used to infer the agent's goal. Note that if the plan inference always infers complete plans, no inference of the goals is needed. Assuming that the agent behaves rationally, the inferred plan is the optimal plan achieving $\hat{g}$, and that contains the set of grounded actions $a \in A$.

We designed a method to infer the user's intention by grounding the utterances to sets of actions defined in a PDDL domain [10]. Semantic role labeling is used to extract semantic frames from the utterances. Each frame is then classified into a partially instantiated PDDL action to form the set $a \in A$. Inferred actions are then used to infer the speaker's intent $\hat{g}, \hat{\pi}$ using plan recognition.

Missing parameters in classified PDDL actions are automatically inferred by the planner as the ones that would make the speaker's inferred plan $\hat{\pi}$ least costly. For example, if the user utters "Give me something to drink" without specifying which glass to use, plan recognition will select the one that is most convenient to reach. The following example illustrates the process in more detail.

Parsing the utterance "Give me something to drink" may yield the following semantic parsing:

- *verb:* give, *patient:* something to drink, *recipient:* me

```
(define (domain cups)                (define (problem cups-3-cups)
    (:requirements                       (:domain cups)
        :strips :typing :equality)       (:objects
    (:types cup - object)                    blue-cup yellow-cup red-cup)
    (:predicates                     ;;tag e:blue-cup   bow:blue,cup
        (finish ?c - cup))           ;;tag e:yellow-cup bow:yellow,cup
;;tag   e:drink bow:drink            ;;tag e:red-cup    bow:red,cup
;;roles e:drink role:ARG2                (:init )
    (:action drink                   ;;goal (finish blue-cup)
        :parameters (?c - cup)       ;;goal (finish yellow-cup)
        :precondition ()             ;;goal (finish red-cup)
        :effect (finish ?c))         )
)
```

Fig. 1. Example of specification of a PDDL domain and problem instances. In green the annotations performed on the entities $e \in E$. The annotations *tag* and *roles* allows to map bag of words into entities, while every *goal* annotation specifies a possible goal for plan recognition.

– *verb:* drink *patient:* something

Assuming that the PDDL domain description contains the actions

– **(give ?to - agent ?i - item)**
– **(drink ?a - agent ?what - beverage ?from - item)**

the utterance may be classified as the partially instantiated actions

$$a = \{(\textbf{give me } \textbf{\textit{None}}), (\textbf{drink } \textbf{\textit{None None None}})\}, \tag{3}$$

with the semantic roles of type *verb* mapped to the action names, and semantic roles *me* mapped to the first argument of **give**. Suppose that G contains two possible user goals: to be served food or to be served a drink. Then, the inferred plan $\hat{\pi}$ will have as goal to drink, as it is the least costly goal achieved with a plan constrained to contain a. Furthermore, when using partially instantiated actions the planner will select as the parameters that were set as ***None*** the objects belonging to the planning instance that would make the plan least costly.

3.1 Utterance classification

For a given PDDL domain and problem definition, we define Act as the set of unique action names, and Obj as the set of all unique objects names. $E \subseteq (Act \cup Obj)$ is the selected subset of entities that are usable to instantiate PDDL actions from semantic roles. In order to map the semantic roles to an action parameter list in the correct order, we specify for every action $a \in (Act \cap E)$ a mapping between semantic roles and parameter indices:

$$M : A \times roles \rightarrow index \cup None. \tag{4}$$

For example, we can define that for the action **drink ?c - cup**, in the simplified drinking domain shown in Figure 1, the semantic role *instrument* is associated to the 1st parameter. The mapping M allows to map semantic roles to the parameters of the annotated actions. M is manually created by annotating the PDDL action descriptions.

Additionally, for finding the correct entities mentioned in the utterance we classify the semantic roles into entities by using a bag of words classifier. The training set for the classifier is obtained by manually annotating the PDDL domain. Figure 1 shows how a drinking domain is potentially annotated. Table 3.1 is the corresponding obtained dataset. Additional data is generated by data augmentation techniques (see Section 3.1) to improve generalization and robustness of classification. The dataset resulting from the annotation process contains records for the entities $e \in E$ only.

X_0 = **Bag of words**	X_1 = **Type**	E = **Id**
blue, cup	cup	blue-cup
red, cup	cup	red-cup
yellow, cup	cup	yellow-cup
drink	action	drink

Table 1. Every action or object in the set of entities E is annotated with a bag of words that are used together with the object type as input for the entity classifier. E, the classifier's target label set, contains the PDDL unique names of the annotated entities.

For every record in the dataset, every word in $x \in X_0$ is encoded into its corresponding word-vector. $x \in X_1$ and $e \in E$ are categorical features encoded using one-hot-vectors. The target classes for the classifier are the unique PDDL labels of the entities in E. The described dataset is used to train a softmax classifier $P_e(E|X_0, X_1)$ that is used to instantiate PDDL actions from semantic roles by the following algorithm:

$$\hat{a} = \underset{e \in E}{\operatorname{argmax}} P_e(e|b_{\mathbf{verb}}, \mathbf{action})$$

$$\forall i, \hat{e_i} = \underset{e \in E}{\operatorname{argmax}} P_e(e|\{w\}_i, type_i), e_i \neq \hat{a}, M(\hat{a}, type_i) \neq None. \tag{5}$$

This sequence of classifications results in an action identifier $\hat{a}$ and a list of associated parameters $\{\hat{e}\}$. For a given action, not all of its semantic roles present in $M(\hat{a}, .)$ might be mentioned in the utterance and the missing ones will appear as *None* in the partially instantiated action. Additionally, semantic roles for which $M(\hat{a}, .) = None$ are discarded.

Notice that SRL could return multiple parsing for a given sentence, one for every verb it contains. In this case we run Algorithm 5 for every different parsing. This also allows to have multiple action declarations in the same sentence, such as in the case of *I'll go to the supermarket and buy maccaroni*, where SRL would produce a parsing for the verbs *go* and *buy*.

Data Augmentation Data augmentation refers to a synthetic increase of the training data in order to increase the size of the dataset and thus the generalization capabilities for the trained model. For every entry in the original dataset we create $N = 1000$ synthetic entries by replacing, in every new record, the words in X_0 with random synonyms found using WordNet. Additionally, for every bag of word, N random words are added. Thus, the description of every object is expanded to the neighboring regions in word vector space by synonyms, while the injected random words increase the robustness of classification [20].

Negative action class As described above, Algorithm 5 will always attempt to match bag of words with entities belonging to the problem. This is not always desirable, especially for auxiliary verbs such as *am* in phrases like *I am repairing my skateboard*, where SRL might label *am* as a verb and Algorithm 5 would thus return the action with similiar name (e.g. **eat**), resulting in a spurious action for the subsequent computations. For this reason, we allow for semantic roles to be classified as *None*. To detect such cases, the classifier is modified to allow the detection of outliers in its hidden layer, by a combination of regularization and Radial Basis Functions (RBF). In the case an input is detected as an outlier, the corresponding computation of the PDDL action or parameter is not performed.

In order to detect outliers, during training the classifier's hidden layer is regularized such that $\mathbf{h} \sim N(\mathbf{0}, \mathbf{1})$, as this helps in giving the data points a silhouette suitable for RBF when evaluated at the hidden layer of the classifier.

After training the regularized classifier, for every target class $e_i \in E$ a centroid c_i (and associated variance σ_i) is computed by averaging the vectors $\mathbf{h}$ generated by the training set. For every c_i only the rows with $e = e_i$ are taken. A Gaussian RBF network is then created with activation

$$\mathbf{a} = e^{\frac{-\|\mathbf{h}-\mathbf{c}\|^2}{\boldsymbol{\sigma}^2}}, \tag{6}$$

with $\|.\|^2$ being the euclidean distance. Using the above defined RBF network, a bag of word is detected as outlier if $\max \mathbf{a} < T$, with T being a threshold hyper-parameter of the model.

3.2 Intent recognition through plan recognition

We apply a method similar to [11] that explicitly allows for partially instantiated actions to be present in the set of observations O, rather than allowing only fully instantiated ones. As the set of observations O we use the trajectory of past actions together with the partially instantiated actions gathered from sentence classification $a \in A$. We treat past observations and uttered actions in different ways, therefore splitting the set O into two parts, O_p and O_f. O_p is constrained to appear in a given sequence, as past observations are gathered in a specific order. For the uttered (possibly) future actions O_f no order is enforced instead.

From an instance $P = (G, I, A)$ (G: goal, I: initial conditions, A: available actions), a sequence of observed past actions O_p, and a set of partially instantiated

future actions O_f, we obtain two modified planning instances $P' = (G', I, A')$ and $P'' = (G'', I, A')$ that are used to compute $C[G+O]$ and $C[G+\neg O]$ respectively, where:

- $A' = A$ with action effects modified as:
  $\forall a \in A'$
  - effects(a') = effects$(a) \cup p_a \rightarrow e_0$ if $a \in O_p$ and is the first of the list (i.e. $n = 0$)
  - effects(a') = effects$(a) \cup p_a \wedge e_{n-1} \rightarrow e_n$ if $a \in O_p$ and $n \geq 1$
  - effects(a') = effects$(a) \cup p_a \rightarrow f_a$ if $a \in O_f$
  - effects(a') = effects(a) otherwise.
  - $p_a = \wedge_i (x_{ai} = arg_{ai})$ if arg_{ai} is specified for action i
  - $p_f = \cup_i f_i$
- $G' = G + O = G \cup e_n \cup p_f$, where e_n is the effect predicate of the last action in O_p, and p_f the conjunction of all of the effect predicates of the actions in O_f.
- $G'' = G + \neg O = G \cup \neg e_n \cup \neg p_f$

Every classified action $\hat{a}$ coming from the Natural Language Understanding component is inserted into the set of future observations O_f. Due to how partially instantiated actions are treated inside P' and P'', these actions receives an additional effect of the type

$$\wedge_i (x_{\hat{a}i} = arg_{\hat{a}i}) \rightarrow f_{\hat{a}}, \tag{7}$$

with $f_{\hat{a}}$ entering the set of goal predicates when computing $C[G+O]$. In this way, when computing this cost, the planner will also attempt to satisfy the actions $\hat{a}$ with the generated plan. For $C[G + \neg O]$ instead, the planner will be asked to not take actions $\hat{a}$. Notice that Eq. 7 is applied only to the parameters that are being specified in the action $\hat{a}$, and for which a valid semantic role was classified.

To compute the probability distribution for the goals, and hence of the intents, we pass the cost difference through a softmax layer obtaining $P(G_i|O) = \gamma e^{-\theta \Delta C_i} P(G)$, being $\Delta C_i = C[G_i + \neg O] - C[G_i + O]$, γ the normalizing factor and θ an hyper-parameter of the model, $P(G)$ the prior probabilities of the goals.

4 Evaluation

The evaluation of our proposed system is divided into two parts. Firstly, the developed plan recognition algorithm is evaluated statistically on different planning instances of high complexity. Statistical evaluation is done to quantify how partially instantiated actions alone contribute in the recognition of the correct goal. Then, we implement speech recognition together with image recognition on a Pepper robot, and evaluate intent recognition in human-robot interaction trials.

4.1 Evaluation of plan recognition with partially instantiated actions

We evaluate our modified plan recognition algorithm, using only partially instantiated actions as observations, on the following planning domains. Our goal is to show how partially instantiated actions scale (i.e. how many specifications the user should give) when inferring goals in complex domains.

Logistics In this well-known domain, a fleet of trucks and airplanes has to deliver packages from starting locations to destination ones. There exists different roads or flight routes in which subsets of trucks or airplanes belong to, and trucks and airplanes can move only in between nodes belonging their corresponding route system. The domain has 10 goals, each of them requiring to deliver 2 packages randomly picked from a set of 10 packages. There are 6 possible actions: *load-truck, load-airplane, unload-truck, unload-airplane, drive-truck, fly-airplane*, each of them having 3 arguments.

Blocks World In this domain there is a table and several blocks on it. Blocks can be stacked on top of each other with the help of a gripper. There are 5 possible goals each of them being a set of towers of blocks. Only one action is possible, *stack-from-to*, that has 3 arguments.

Hospital In the hospital domain a nurse has to inject drugs to the patients admitted at the hospital. Several rooms are dedicated for the patients and are spread over 3 floors. A set of elevators allow the nurse to change floor. The drugs are all initially stored in a storage room, and every patient requires a specific mixture of drugs. In addition, time constraints determine at which hour of the day the patients should receive their injections. The domain has 12 goals, each of them being the treatment of 2 patients. Patients, rooms, drugs and hours are chosen randomly when the domain is generated. There are 5 possible actions: *take-medicine*, *wait-for-hour*, *inject-drug*, *move*, *take-elevator*, with a mean number of arguments of 3.2.

For the three domains, each trial is carried as follow: a random goal is selected and an optimal plan A for it is generated. With a parameters $\alpha \in [0, 1]$ we selected the percentage of actions in A to keep and use for O_f (always at minimum one action was kept), with another parameter $\beta \in [0, 1]$ we specified the percentage of parameters to keep for every action. Retained parameters and actions are randomly selected at every trial. Every goal was tested in equal measure, and for every possible combination of α and β 10 trials were averaged.

Statistical results (Figure 2) show how both α and β are important in plan recognition. When no parameter is specified ($\beta = 0$) the recognition gives the lowest accuracy values independently of α. In blocks world, being only one action present, this results in random guess performance; in logistics, this performance is slightly above random guess. Given that at least a parameter is specified ($\beta \geq \frac{1}{3}$ in our proposed scenarios), α becomes the dominating factor for the recognition accuracy, as better shown in the right column of Figure 2.

For practical scenarios, a relevant case is when only one action is specified together with few parameters (e.g. $\alpha = 0$, $\beta \geq \frac{1}{3}$). In this case in the obtained accuracy is in the 20-60% range. Thus, if we expect a limited amount of uttered

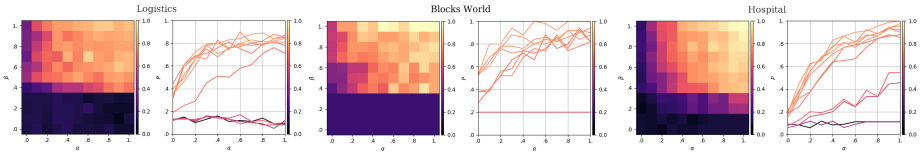

Fig. 2. Results of the statistical evaluation. The matrix on the left shows the tested combinations of values for α and β. Color, from black to white, indicates the obtained accuracy for every combination. On the right is plotted the accuracy in finding the correct goal using different values of α. Every different line correspond to a different value of β.

commitments, the introduction of the set of ordered observations O_p is an important factor for achieving high accuracy. Nevertheless, notice that this is a pessimistic measure as in the benchmarks, actions and parameters are chosen randomly, while during real interactions we can expect the observed agent more likely to communicate informatively rather than randomly. Additionally, having the possibility of selecting the classifiable actions, we can ensure that only the actions that are pivotal for the plan recognition problem are expressible an utterances. No such constraint was present in the benchmarks.

4.2 Evaluation in an HRI setting

In order to test intent recognition in an interaction with a robot, we implemented the described system in an HRI setting using a Pepper robot. In the proposed scenarios, an experimenter stands in front of Pepper and interacts with it using speech. Utterances are detected through the Google Speech API, and classified into PDDL actions using Algorithm 5. Additionally, based on the presence of different objects in the current visual scene, the truth value of selected predicates inside the planning instance is modified. Visual objects are detected using a classifier pre-trained on the YOLO dataset [13]. Figure 3 shows the full developed architecture.

Two different scenarios are evaluated: a *Groceries* scenario where inference on contextual elements is used to discriminate between the user intentions buying food or buying cigarettes. The second *Cups* scenario is created to verify how, given an utterance with partial specifications, missing parameters in the corresponding PDDL action are correctly inferred.

Groceries setting In this hypothetical setting the planning instance is programmed to detect whether the speaker in going to buy groceries or cigarettes. Through every trial, the user is asked to state what he is going to do. The possible choices are to buy from the grocery store, to eat food, or to smoke. The possible goals are to eat or to smoke.

Depending on the presence of food on the table in front of Pepper, the corresponding predicates expressing availability for that particular food are set inside the planning instance.

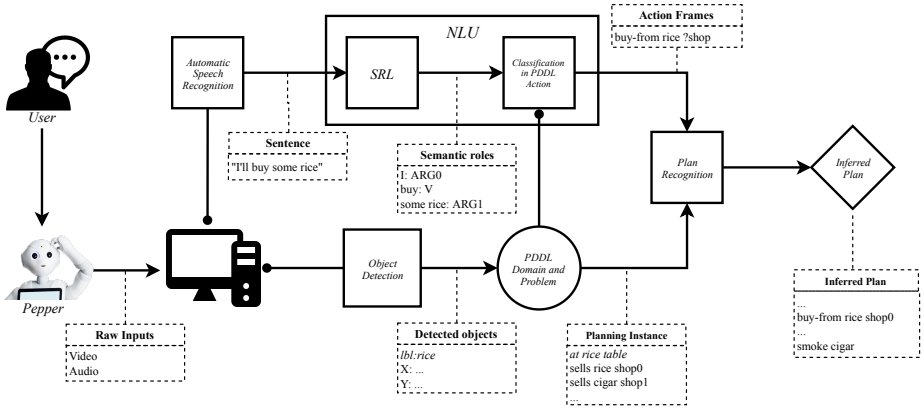

Fig. 3. Main architecture of the implemented system. Audio and video from the Pepper robot are streamed to a workstation where visual objects are identified and audio converted to text. Detected objects are used to modify the planning instance, while speech is classified into partially instantiated actions. The result is used to infer the speaker's intent through plan recognition.

The annotation of the PDDL domain and problem with semantic roles and bag of words is performed in a similar fashion as the one shown in Figure 1. The expected outcomes of the trials are:

- If the user utters that he wants to go to the supermarket or buy food, the inferred goal depends on the predicate **(at rice fridge)**, which is set to true if a visual object of type *cup* or *bowl* is detected. In such case, the inferred goal is set to smoke, and otherwise to eat.
- If the user utters that he wants to cook or eat, the inferred goal is to eat, expressed by the predicate **(consumed rice)**.
- If the user utters that he wants to smoke, the inferred goal is to smoke, expressed by the predicate **(consumed cigar)**.

Cups setting In this setting the user can ask for a drink from three different cups on the table, each one with a different associated cost to reach. The only action that is accessible through speech is *drink*, with one optional parameter specifying which cup to use. There are three possible goals, achieved by the drink action using the different cups. The expected outcomes of the trials are:

- If the user says that he wants to drink, without specifying a cup, the goals have equal probabilities as no discriminating information is present. The inferred goal is returned as to drink from the blue cup.
- If the user specifies any cup for drinking, the inferred goal is to drink with the mentioned cup.

During the experiments the algorithm behaved as expected, and the robot inferred different intentions based on the perceived contextual variable.

A video showing the different experimental trials for both scenarios is available at *https://youtu.be/33Dinfh7_0Y* (please make sure the address is properly typed).

5 Conclusions

W proposed an algorithm to infer a speaker's intention from utterances and context. The proposed method is based on the classification of the utterances into PDDL actions, followed by a plan recognition algorithm using classical planning. Matching of parts of the utterance to actions and parameters is done using semantic role labeling. Recognized utterances are used to infer the partial plan and goal of the speaker, or to guide execution of actions when part of the information is missing. The proposed system allows to utilize utterances in a contextual way, and depending on the state of the planning instance they lead to different inferred intentions. In our HRI experiments the robot reacts to the user utterances by simply telling the goal it inferred. More complex type of reactions are also possible and are left for future research. The major benefit with our approach is that the intentions do not have to be hardcoded for combinations of a large number of contextual states, but is rather intelligently inferred by the robot in a way that scales both with number of possible intents and contextual variables.

We discuss the issue that when instantiating robot commands all of required parameters must be present in order for the commands to be executed. With the support of a planning domain, partially instantiated actions allow instead to take advantage of the principle of rational action, thus inferring missing parameters as the ones that would yield the most optimal intention. This method is in contrast with other approaches where the combinations of available objects are exhausted or searched in order to find the best match.

Evaluation showed how partially instantiated actions positively contribute to inference of the correct goal. For complex scenarios they yield a fair accuracy only when present in fairly large numbers. Additionally, the system was implemented in an HRI setting using a Pepper robot, and we verified its correct operation in several simplistic but relevant experiments.

Future research include incorporation of a dialogue manager to create/mediate intentions, of multiple agents in the inferred intentions, and collection of a structured knowledge-base for planning domains and annotations, possibly testing grounding algorithms that generalize over them.

Acknowledgments This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 721619 for the SOCRATES project.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of the 17th international conference on Computational linguistics-Volume 1. pp. 86–90. Association for Computational Linguistics (1998)

2. Bastianelli, E., Castellucci, G., Croce, D., Iocchi, L., Basili, R., Nardi, D.: Huric: a human robot interaction corpus. In: LREC. pp. 4519–4526 (2014)
3. Bensch, S., Jevtić, A., Hellström, T.: On interaction quality in human-robot interaction. In: International Conference on Agents and Artificial Intelligence (ICAART). pp. 182–189 (2017)
4. Breazeal, C., Aryananda, L.: Recognition of affective communicative intent in robot-directed speech. Autonomous robots **12**(1), 83–104 (2002)
5. Chakraborti, T., Kambhampati, S., Scheutz, M., Zhang, Y.: Ai challenges in human-robot cognitive teaming. arXiv preprint arXiv:1707.04775 (2017)
6. Chen, H., Tan, H., Kuntz, A., Bansal, M., Alterovitz, R.: Enabling robots to understand incomplete natural language instructions using commonsense reasoning. CoRR (2019)
7. Demiris, Y.: Prediction of intent in robotics and multi-agent systems. Cognitive Processing **8**(3), 151–158 (Sep 2007)
8. He, L., Lee, K., Lewis, M., Zettlemoyer, L.: Deep semantic role labeling: What works and what's next. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 473–483 (2017)
9. Kelley, R., Browne, K., Wigand, L., Nicolescu, M., Hamilton, B., Nicolescu, M.: Deep networks for predicting human intent with respect to objects. In: 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp. 171–172 (March 2012)
10. McDermott, D.: Pddl-the planning domain definition language (1998)
11. Ramírez, M., Geffner, H.: Probabilistic plan recognition using off-the-shelf classical planners. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
12. Rani, P., Liu, C., Sarkar, N., Vanman, E.: An empirical study of machine learning techniques for affect recognition in human–robot interaction. Pattern Analysis and Applications **9**(1), 58–69 (2006)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
14. Schaefer, K.E., Chen, J.Y., Wright, J., Aksaray, D., Roy, N.: Challenges with incorporating context into human-robot teaming. In: 2017 AAAI Spring Symposium Series (2017)
15. Sukthankar, G., Geib, C., Bui, H.H., Pynadath, D., Goldman, R.P.: Plan, activity, and intent recognition: Theory and practice. Newnes (2014)
16. Teixeira, A.: A critical analysis of speech-based interaction in healthcare robots: Making a case for the increased use of speech in medical and assistive robots. Speech and automata in health care pp. 1–29 (2014)
17. Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)
18. Thomas, B.J., Jenkins, O.C.: Roboframenet: Verb-centric semantics for actions in robot middleware. In: 2012 IEEE International Conference on Robotics and Automation. pp. 4750–4755. IEEE (2012)
19. Tomasello, M., Carpenter, M., Call, J., Behne, T., Moll, H.: Understanding and sharing intentions: The origins of cultural cognition. Behavioral and Brain Sciences **28**(5), 675–691 (2005)
20. Wei, J.W., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)