Predicting humans: a sensor-based architecture for real time Intent Recognition using Problog

Gennaro Daniele Acciaro[1], Fabio Aurelio D'Asaro[2] and Silvia Rossi[1]

[1]DIETI, University of Naples Federico II, Italy
[2]Logic Group, Department of Philosophy, University of Milan, Italy

Abstract

In a world where the population is aging, products that improve living comfort will have more importance in people's lives. These products must interpret the intentions of those who live in the house to provide them with assistance in their daily tasks. Motivated by these issues, we present an architecture for real-time *Intention Recognition*. We demonstrate it with a kitchen use-case, where the agent prepares a meal. Our goal is to recognize what type of meal the agent intends to prepare. The architecture consists of two layers, namely the "Classification Layer" and the "Problog Layer". The Classification Layer recognizes the environment through sensors and classifiers, and passes the information to the Problog Layer, which uses Problog to infer the intention. The Problog Layer consists of two Knowledge Bases: the "Static KB" and the "Dynamic KB". The former axiomatically describes the intentions we want to recognize, while the latter is generated at runtime using information from the Classification Layer.

Keywords

Intention Recognition, Problog, Knowledge Representation, Smart Technologies

1. Introduction

The latest version of "World Population Ageing" - an annual report of the United Nations[1] - outlines two meaningful statistics: in 2020, people over 65 are 727 million and are expected to increase to 1.5 billion by 2050. This same report mentions that, in most developed countries, these people will manage to live without a caregiver's external support, mainly thanks to good welfare and healthcare system. For these reasons, we can assume that soon it will be necessary to understand these people's intentions in an automated way to provide them with better comfort in a home environment through smart-home products designed to help these people in their daily tasks.

This paper focuses precisely on this aspect, presenting a logic-based architecture for intention recognition, which we demonstrate through a proof of concept. The use-case is that of smart kitchen environment, where our automated system aims to recognize what the human intends to cook – which is a particular instance of an *Intention Recognition* problem. According to [1],

WOA 2021: 22nd Workshop From Objects to Agents, September 01–03, 2021, Bologna, Italy
fabio.dasaro@unimi.it (F. A. D'Asaro); silrossi@unina.it (S. Rossi)
http://www.fabiodasaro.com (F. A. D'Asaro)
0000-0002-2958-3874 (F. A. D'Asaro); 0000-0002-3379-1756 (S. Rossi)
© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/undesa_pd-2020_world_population_ageing_highlights.pdf

CEUR-WS.org/Vol-2963/paper5.pdf

Intention Recognition is the process of becoming aware of the intention of another agent and, more technically, inferring an agent's intention through its actions and their effects on the environment. Hence, an intention is inferred from a sequence of actions. In this paper, we detect an action as the combination of two parts, namely the object the human is working with (e.g., milk, orange, knife) and the human's pose, which we consider in order to disambiguate what the agent is currently doing with the object (e.g., cutting, taking or pouring). In our architecture, two different Machine Learning-based classifiers detect these two parts of an action. We then feed the classifiers' outputs to a Problog architecture.

Since we are mainly concerned with sequences of actions, we chose to use a popular temporal ontology known as the Event Calculus [2, 3] which allows for the definition of events occurring along an explicit timeline. Moreover, given our Machine Learning classifiers and actions' probabilistic nature, we found it natural to use a probabilistic extension of this language. Among the possible choices [4, 5, 6] we picked the Problog-based system Prob-EC [6] as this has been successfully applied to similar use cases such as Event Recognition. It is worth noting here that, unlike the Event Recognition task, in Intention Recognition one aims to detect what the agent intends to do in the near future (e.g., "prepare a salad"), rather than an activity that is currently being performed (e.g., "two people are meeting each other").

It is worth noting that although in this paper we present a specific use-case of our architecture, this proof of concept may serve as a blueprint for applications in very different domains. For instance, a conversational chat-bot may want to track user activity in the calendar over multiple days to infer long-term intentions, e.g. it may deduce that the user intends to lose weight from the fact s/he has been exercising a lot and s/he's been buying low-fat foods for the last two weeks. It may also e.g. be employed by shopping centers as an anti-theft system that processes CCTV footage in real time. Furthermore, it could be used to provide both long and short-term assistance to the elderly, e.g. by understanding their intention and providing adequate assistance to finalize them. These use-cases are all very different from each other. However, as we will show in the following sections, they share a common structure: they all make use of time-stamped multimodal data which must be processed in order to deduce some form of user intention. This is precisely the type of problem our architecture aims to tackle. Given the agnostic nature of the building blocks of our architecture, we claim that our work can be readily generalized from our simple proof of concept to more complex domains.

This paper is organized as follows. In section 2 we shortly review related work. In section 3 we provide an overview of the architecture. In section 4 we explain in detail the technologies used to create the proposed architecture. In section 5 we demonstrate the architecture in our specific kitchen use-case. In Section section 6 we present some tentative conclusions and hint at future work.

2. Related Work

The problem of Intention Recognition is a significant one in the field of Human-Computer Interaction. It applies to a wide variety of tasks, ranging from smart homes [7] to neurosciences [8]. Intent Recognition has become an increasingly important field of research in recent years, and several papers have been published proposing different techniques and technologies to

approach it.

As Charniak indicated in 1993 [9], the nature of this discipline must be probabilistic. Bayesian networks are often used when working with uncertainty. Nazerfard and Cook [10] use Bayesian Networks with a continuous normal distribution to predict when the next intended action will occur. Pereira and Han [11] propose the use of Casual Bayesian Networks with plan generation techniques to predict hidden actions and unobservable effects. To a similar aim, Muncaster and Ma [12] propose Dynamic Bayesian Networks. However, in the context of Intent Recognition, Bayesian Networks have two particular problems:

- They do not allow for an explicit representation of a timeline,
- It is difficult to track the sequence of actions, which is central to the very nature of intentions.

For these reasons, we preferred a Probabilistic Event Calculus approach over Bayesian Networks.

Vilain [13] proposes using the analysis of an acyclic Context-Free Grammar to interpret sequences of steps, using a deductive process. The use of Spatial-Temporal And-Or Graphs (ST-AOG) was proposed in [14] and [15]. The ST-AOGs define the sub-activities constituting the final intention. In this paper, we predict agent intentions in Problog through probabilistic rules that correspond to the sub-tasks of an intention.

As we describe in the remainder of this paper, this paper uses two fundamental technologies: Problog and convolutional neural networks (CNN). Problog [16] is a Probabilistic Logic Programming Language with a Prolog-like syntax. Clauses can be decorated with a probability $p \in [0, 1]$ according to the following syntax:

$$p :: \mathit{Head} \text{ :- } \mathit{Body}.$$

The Problog Layer of our architecture implements a probabilistic variant of the Event Calculus known as Prob-EC [6]. It consists of two Knowledge Bases: a Static KB (SKB) and a Dynamic KB (DKB). The SKB contains the domain independent axioms of Prob-EC and general static information about actions and intentions. The DKB is updated at runtime by translating classifier data into probabilistic events whenever the secondary server receives an action. The Problog Layer computes the probability of observing the sequence of action given each possible intention by querying the SKB and the DKB, and eventually outputs the intention(s) maximizing the corresponding likelihood.

As a running example, throughout the paper we use a set $\mathcal{I} = \{\mathit{Breakfast}, \mathit{Pesto\ Pasta}, \mathit{Tomato\ Pasta}, \mathit{Fruit\ Salad}, \mathit{Fish}\}$ of possible intentions. The Problog Layer calculates the likelihood of these activities when a sequence of actions is performed (e.g. take milk, pour it, and take cookies), and then informs the Main Server about the intention that maximizes such likelihood (e.g., preparing breakfast), which in turn displays it on the screen. In the case of a tie, we display all activities with maximal likelihood. A Convolutional Neural Network (CNN) is a typology of neural network able to perform classifications based on the operation of convolution between matrices.

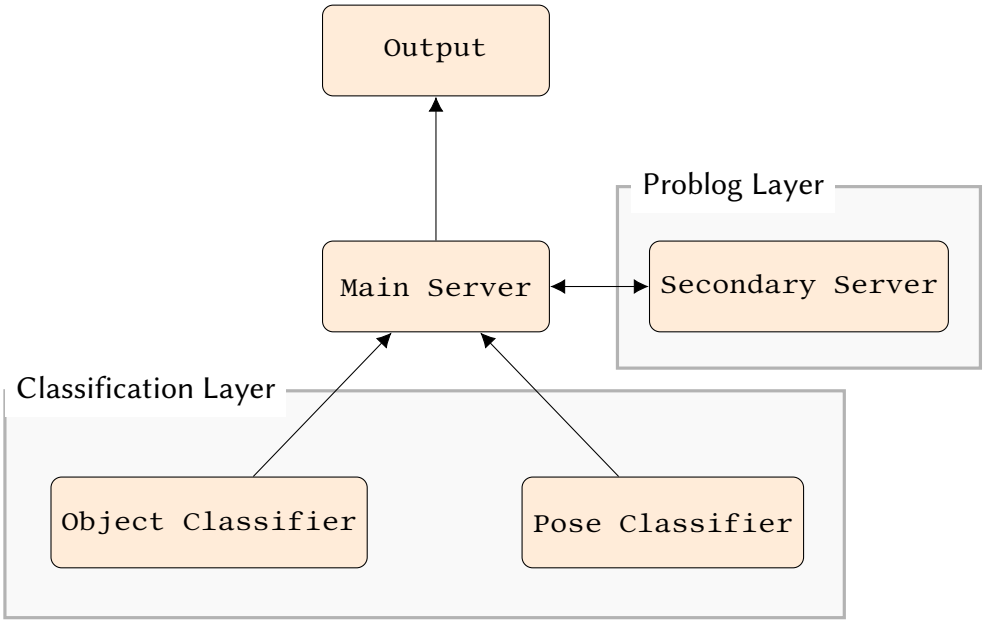

Figure 1: System Architecture

3. Architecture

Our simple proposed architecture for real-time intention recognition is shown in Figure 1. Our intention recognition system consists of two cameras. One camera, located near the human agent, is devoted to recognizing objects, with the other one installed further away from the human to recognize its pose. The Classification Layer of our architecture processes the video stream captured from the cameras. It extracts the object the human agent is currently using, and her pose. Then, it forwards this information to the main server. To prevent flooding, we set a minimum delay of 0.5 seconds between requests to the main server.

The main server stores the data in a buffer, and when the buffer is full, it selects the most frequent action in the buffer and sends it to a secondary server. When the secondary server receives an action, it translates the action into a Problog probabilistic fact and adds it to the (Dynamic) Knowledge Base. It then compiles the whole script and queries it to figure out the most probable intention. We implemented communication between layers through HTTP calls. In particular, the output of the secondary server is a JSON created from the output values that the script in Problog returns. In the remainder of this section, we discuss each component of our system in greater detail.

3.1. Physical Setup

Our controlled environment consists of a video camera (used by the object classifier) on the working table, facing the agent. The other video camera (used for pose recognition) is on the

working table, approximately at 1.8 meters from the agent.

3.2. Classification Layer

The Classification Layer consists of two classifiers that receive the video stream from the two cameras. One of the classifiers aims to recognize objects, with the other one recognizing the agent's pose.

The object classifier is based on MobileNet [17], a Convolutional Neural Network that uses a technique called "Depthwise Separable Convolution" to reduce the computational cost of convolution [17]. Two hyperparameters allow one to further improve MobileNet's computational efficiency, namely the Width Multiplier α and Resolution Multiplier ρ, that optimize the model according to the context.

We used PoseNet [18] to perform the pose recognition task. It supports recognition algorithms both for a single person and for several people simultaneously. PoseNet recognizes 17 key points corresponding to important points of the human skeleton. It associates spatial coordinates to each keypoint, which it then further processes to classify the user's pose. We chose these two models due to their simplicity in performing class training. Nonetheless, the architecture proposed in this paper is also scalable with respect to several other technologies or alternatives that perform the same purposes of object and pose recognition.

4. Implementation

As mentioned above, the Problog script consists of two knowledge bases: the SKB and DKB.

The DKB gets updated every time the classifiers detects an object or a gesture. For instance, if the pose classifier detects that the human agent is performing the gesture *take* at time t, we augment the DKB with the following probabilistic fact:

$$p :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), t)$$

where *happensAt* is a standard Prob-EC predicate to handle event occurrences, and p is the recognition probability associated to the gesture *take* by the pose classifier. Similarly, if the object classifier detects that the human agent is interacting with the ingredient *apple* at time t, we translate this to:

$$p :: \mathit{happensAt}(\mathit{ingredient}(\mathit{apple}), t)$$

On the other hand, the SKB defines how the probability of an intention increases as the result of recognizing an object and/or an action, as in the following example:

$$0.2 :: \mathit{initiatedAt}(\mathit{breakfast} = \mathit{true}, T) \text{ :- } \mathit{take}(\mathit{milk}, T).$$

$$0.5 :: \mathit{initiatedAt}(\mathit{breakfast} = \mathit{true}, T) \text{ :- } \mathit{takeAndPour}(\mathit{milk}, T).$$

where *initiatedAt* is a standard Prob-EC predicate to quantify how an event occurrence affects the probability of a *fluent*, i.e., a property of the world, which in our example is the intention

breakfast to be recognized. The two predicates *take* and *takeAndPour* are abbreviations defined as follows:

$$take(Obj, T) \text{ :- } happensAt(gesture(take), T),\ happensAt(ingredient(Obj), T)$$

$$takeAndPour(Obj, T) \text{ :- } take(Obj, Tprec),\ pour(Obj, T),\ Tprec < T$$

Finally, in order to query the likelihood of preparing *breakfast* at time t we use the in-built Problog predicate *query* as follows:

$$query(holdsAt(breakfast = true, t))$$

We query the SKB and the DKB in order to get the intention that maximizes the likelihood, and pass it on to the main server, which displays it on the screen. In the case of a tie, we display all activities with maximum likelihood.

5. Demonstration

In this section, we demonstrate how our architecture behaves in a few controlled experiments. We first set up the Problog SKB with reasonable probabilities associated with actions and intentions. Then, we let a human agent perform a series of actions. The system analyzed the video streams as outlined in section 3 and the Problog Layer produced the corresponding DKB. In each of the following subsections, we focus on specific experimental runs, by providing the DKB and showing how the probability of intentions evolves over time.

5.1. Equally likely intentions

Figure 2 shows one case in which the architecture is unable to disambiguate between two possible intentions up until time point 4. Note that in this example, the ingredients and their associated actions are sequences of actions that may constitute an intent. As you can see by observing "Tomato pasta" our probabilities are monotonous because we do not exclude the ambiguous case in which the user wants to return to an intention previously started and not concluded.

In this experiment, the sequence of events was as follows: the human agent took the water at times 0 and 1, poured it at time 2, and then took the pasta at time 3. This narrative is captured by the events generated by the Problog Layer, which in this case are as follows:

$$1 :: happensAt(gesture(take), 0).$$
$$1 :: happensAt(ingredient(water), 0).$$
$$1 :: happensAt(gesture(take), 1).$$
$$1 :: happensAt(ingredient(water), 1).$$
$$1 :: happensAt(gesture(pour), 2).$$
$$1 :: happensAt(ingredient(water), 2).$$

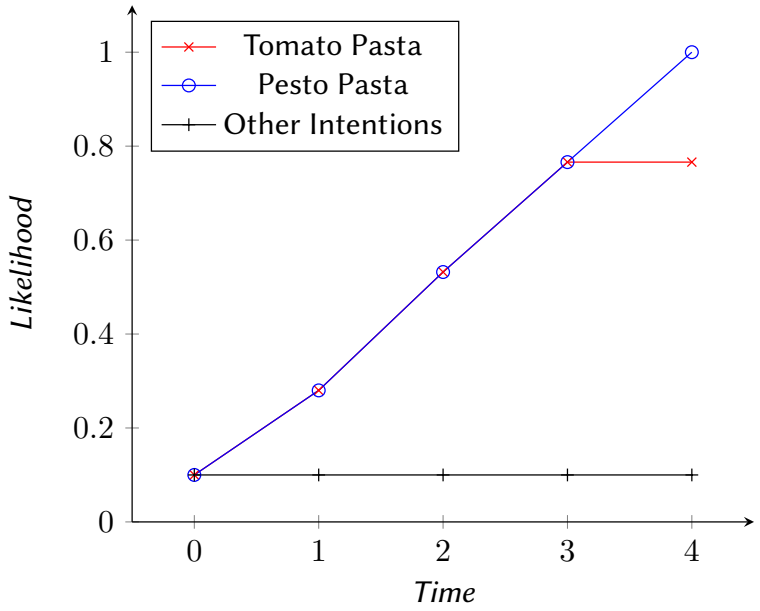

Figure 2: At time-points 0, 1, 2 and 4 the human agent performs actions that are compatible both with the intention of preparing *Tomato Pasta* and *Pesto Pasta*. However, at instant 4 the agent *takes* the ingredient *pesto*, making *Pesto Pasta* the most likely intention. All other intentions are considered to be very unlikely at all time points.

$$1 :: happensAt(gesture(take), 3).$$
$$1 :: happensAt(ingredient(pasta), 3).$$
$$1 :: happensAt(gesture(take), 4).$$
$$1 :: happensAt(ingredient(pesto), 4).$$

At time 3, the system is unable to figure which type of pasta the agent intends to prepare. This can be clearly seen from the figure, which shows the systems assigns equal likelihood to the intention of preparing *Tomato Pasta* and *Pesto Pasta*. However, as soon as the human agent took the pesto (time 4), the system was able to determine that her intention is that of preparing *Pesto pasta*.

5.2. Time factor

Figure 3 shows how Prob-EC allows us to overcome one of the problems affecting Bayesian Networks in an Intention Recognition setting, i.e. the management of the temporal factor. In this example, the human agent has an interaction with ingredient *milk* lasting 4 time points. This is encoded in the following DKB:

$$1 :: happensAt(gesture(take), 0).$$
$$1 :: happensAt(ingredient(milk), 0).$$

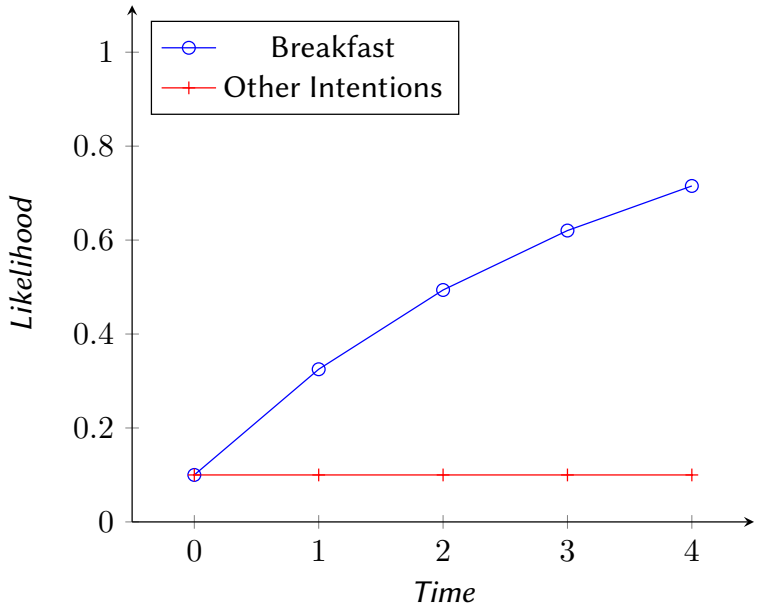

Figure 3: In this example, the agent interacts with the ingredient *milk* at instants 0, 1, 2 and 3. As the *Breakfast* intention becomes more likely, the other intentions remain unlikely as they are incompatible with the use of *milk*.

$$1 :: happensAt(gesture(take), 1).$$
$$1 :: happensAt(ingredient(milk), 1).$$
$$1 :: happensAt(gesture(take), 2).$$
$$1 :: happensAt(ingredient(milk), 2).$$
$$1 :: happensAt(gesture(take), 3).$$
$$1 :: happensAt(ingredient(milk), 3).$$

In this case it is reasonable that the longer the agent interacts with the milk lasts, the greater its intention to have breakfast. Our system behaves accordingly, as shown in fig. 3.

5.3. The complete use case

In previous examples, we had 100% recognition accuracy attached to all events. This was to show how our system behaves when classifiers do not have an associated classification accuracy. We now look at a case where the probability of facts may vary according to classification accuracy, as in the case of our specific system.

In the following experiment, we asked the human agent to perform actions as she normally would when preparing breakfast. She held the milk for two time points (with actions recognized with 85% and 96% accuracy, respectively). Due to a classification problem, the system recognized an orange (78% accuracy) at time 2. Then, she poured the milk (93% accuracy), and then temporarily abandoned his main intention to read the expiration date of a jar of pesto

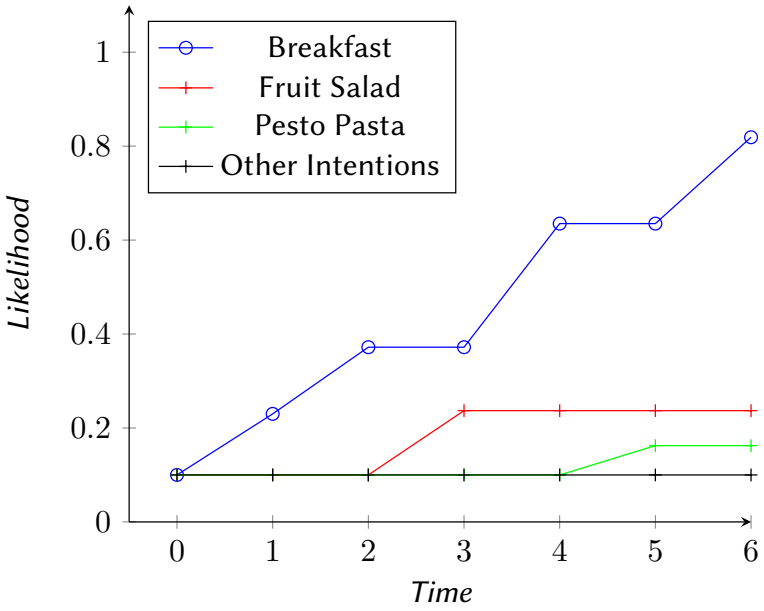

Figure 4: In this example, we show how the architecture behaves in a more realistic use case. The intention of preparing *Breakfast* is correctly recognized at all time points, in spite of a classifier error and the user temporarily performing another task.

(68% accuracy) before grabbing cookies (91% accuracy) to finalize the intention of preparing breakfast. The associated DKB was as follows:

$$
\begin{aligned}
&0.84 :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), 0).\\
&0.86 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{milk}), 0).\\
&0.92 :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), 1).\\
&0.98 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{milk}), 1).\\
&0.78 :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), 2).\\
&0.81 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{orange}), 2).\\
&0.93 :: \mathit{happensAt}(\mathit{gesture}(\mathit{pour}), 3).\\
&0.89 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{milk}), 3).\\
&0.68 :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), 4).\\
&0.76 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{pesto}), 4).\\
&0.91 :: \mathit{happensAt}(\mathit{gesture}(\mathit{take}), 5).\\
&0.94 :: \mathit{happensAt}(\mathit{ingredient}(\mathit{cookies}), 5).
\end{aligned}
$$

Figure 4 shows the results in this case. Note that the intention of preparing *Breakfast* is correctly recognized at all time points.

6. Conclusion and Future Work

This paper proposes the application of probabilistic logic-based architectures, more specifically Problog and Prob-EC in our case, to the task of Intention Recognition. As demonstrated in our example, we believe such tools may prove highly effective and impactful. Although similar approaches have been proposed for Event Recognition, using Event Calculus based architectures for real-time recognition of agents' intention may open up new possibilities and overcome some difficulties with other techniques. Our proposed architecture for a use-case of a smart kitchen can be seen in fig. 1. It includes two main layers: the Classification Layer, sensing the environment, and the Problog Layer, which performs logic-probabilistic inference to derive the most likely intention of the user. In this work, we provide a proof of concept that mainly shows how our architecture works in a series of controlled experiments. Nonetheless, this very architecture may be generalized to other use-cases. The next step of this research will involve human judgment to systematically evaluate the detection accuracy of intention. Furthermore, we foresee that such an architecture might suit the task of learning and predicting complex intentions that were not described a priori. In the future, we aim to further explore these applications and extensions. Finally, we aim to extend the use case to other objects and poses in order to be able to evaluate the performance of the system with respect to the classification of intentions.

References

[1] T. A. Han, Intention Recognition, Commitment and Their Roles in the Evolution of Cooperation From Artificial Intelligence Techniques to Evolutionary Game Theory Models, Springer Berlin Heidelberg, 2013.

[2] R. Kowalski, M. Sergot, A logic-based calculus of events, in: Foundations of knowledge base management, Springer, 1989, pp. 23–55.

[3] R. Miller, M. Shanahan, Some alternative formulations of the event calculus, in: Computational logic: logic programming and beyond, Springer, 2002, pp. 452–490.

[4] A. Skarlatidis, G. Paliouras, A. Artikis, G. A. Vouros, Probabilistic event calculus for event recognition, 2013. arXiv:1207.3270.

[5] F. A. D'Asaro, A. Bikakis, L. Dickens, R. Miller, Probabilistic reasoning about epistemic action narratives, Artificial Intelligence 287 (2020). URL: https://www.sciencedirect.com/science/article/pii/S0004370219300906. doi:https://doi.org/10.1016/j.artint.2020.103352.

[6] A. Skarlatidis, A. Artikis, J. Filippou, G. Paliouras, A probabilistic logic programming event calculus, Theory and Practice of Logic Programming 15 (2015) 213–245. doi:10.1017/S1471068413000690.

[7] J. Rafferty, C. D. Nugent, J. Liu, L. Chen, From activity recognition to intention recognition for assisted living within smart homes, IEEE Transactions on Human-Machine Systems 47 (2017) 368–379. doi:10.1109/THMS.2016.2641388.

[8] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J. C. Mazziotta, G. Rizzolatti,

Grasping the intentions of others with ones own mirror neuron system, PLoS Biology 3 (2005). doi:10.1371/journal.pbio.0030079.

[9] E. Charniak, R. Goldman, A bayesian model of plan recognition, Artificial Intelligence 64 (1993) 53–79. doi:10.1016/0004-3702(93)90060-O.

[10] E. Nazerfard, D. Cook, Using bayesian networks for daily activity prediction, AAAI Workshop - Technical Report (2013) 32–38.

[11] L. Pereira, T. A. Han, Intention recognition via causal bayes networks plus plan generation, volume 5816, 2009, pp. 138–149. doi:10.1007/978-3-642-04686-5_12.

[12] J. Muncaster, Y. Ma, Activity recognition using dynamic bayesian networks with automatic state selection (2007). doi:10.1109/WMVC.2007.5.

[13] M. Vilain, Getting serious about parsing plans: A grammatical analysis of plan recognition, in: Proceedings of the Eighth National Conference on Artificial Intelligence - Volume 1, AAAI'90, AAAI Press, 1990, p. 190–197.

[14] S. Holtzen, Y. Zhao, T. Gao, J. B. Tenenbaum, S.-C. Zhu, Inferring human intent from video by sampling hierarchical plans, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016). doi:10.1109/iros.2016.7759242.

[15] S. Qi, S. Huang, P. Wei, S.-C. Zhu, Predicting human activities using stochastic grammar, 2017 IEEE International Conference on Computer Vision (ICCV) (2017). doi:10.1109/iccv.2017.132.

[16] L. De Raedt, A. Kimmig, H. Toivonen, Problog: A probabilistic prolog and its application in link discovery., in: IJCAI, volume 7, Hyderabad, 2007, pp. 2462–2467.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. arXiv:1704.04861.

[18] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, Towards accurate multi-person pose estimation in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4903–4911.