From Perception to Semantics: An Environment Representation Model Based on Human-Robot Interactions

Yohan Breux, Sébastien Druon, René Zapata

From Perception to Semantics : An environment representation model based on human-robot interactions.

Yohan Breux[1], Sebastien Druon[1] and Rene Zapata[1]

Abstract— **A robot, in order to be autonomous, needs some kind of representation of its surrounding environment. From a general point of view, basic robotic tasks (such as localization, mapping, object handling, etc.) can be carried out with only very simple geometric primitives, usually extracted from raw sensor data. But whenever an interaction with a human being is involved, robots must have an understanding of concepts expressed in human natural language. In most approaches, this is done through a prebuilt ontology.**

In this paper, we try to bridge the gap between data driven methods and semantic based approaches by introducing a 3-layer environment model based on "instances" : sensor data based observations of concepts stored in a knowledge graph. We will focus on our original object-oriented ontology construction and illustrate the flow of our model in a simple showcase.

I. Introduction and related work

To increase their autonomy and facilitate their interaction with human beings, mobile robots must have an understanding of the world at several levels of abstraction. These are threefold : sensorial perception, object (instance) recognition and semantic knowledge. Past researches mainly focused on low level geometric descriptions and/or vision-based information to develop robots able to move freely in their nearby environment. More recently, semantic information is also taken into account in tasks such as object classification or semantic mapping. However, it is often a shallow use (labeling) limited to some predefined context.

Simultaneous Localization and Mapping (SLAM) has been an important subject of research in the past decade[1][2]. It provides a 2/3D geometric representation of environment based on occupancy grid or structured point cloud. Vision-based manipulation[3][4] is another example of geometric-based modeling. These methods are constrained to fit on predefined models and thus lack genericity.

Efforts have been made to enrich representations with semantic information. The mobile robot system proposed by Sunderhauf et al. [5] categorizes places based on Convolutional Neural Network (CNN)[6]. They focus on semantics related to places and they condition the probability of presence of objects on it. In comparison, our model is centered on **object-related** semantics.

Some frameworks, closer from our work, separate the semantic and sensorial information in several layers. [7][8] build a multi-layer spatial representation of the world. The layer of lower level maintains a metric map. The next layer clusters the map into places with several attached properties such as object class occurrence. The last layer represents a probabilistic chain graph expressing relations between concepts.

Those frameworks leverage both visual and semantic modalities but are mainly centered on the mapping process. They both lack deeper understanding on relation between objects of the scene and are difficult to scale up on an open world.

Up to now we reviewed methods relying mainly on sensorial data with some common-sense knowledge on top of it. Some works are the other way around : their frameworks are grounded on knowledge representation with low-level sensor data used for inference correctness. KnowRob[9] is a task-oriented system leveraging semantic knowledge in the context of human assisting robot. They use a Knowledge base bootstrapped on OpenCyc ontology[10]. Visual inferences are made at run-time by using *computables*, which are called when the attached concept is part of a query. The RoboSHERLOCK system[11] of the same authors fusions possibly contradictory inferences given by different perception algorithms for object classification. For comparison, our work is **task-independent** and focus on general objects understanding.

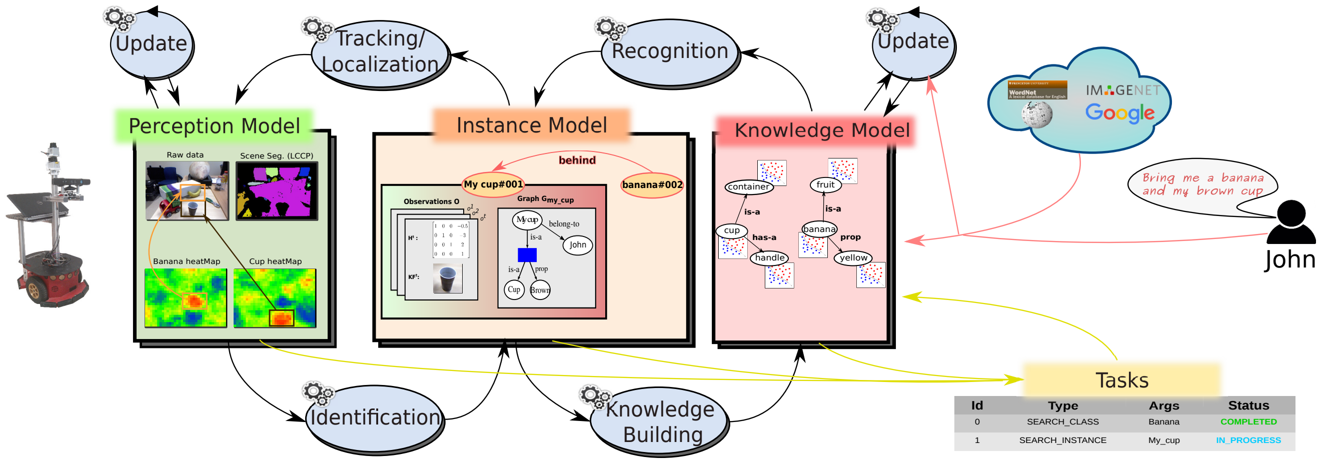
A problem of interest for us is the detection of semantically known but unseen object classes (zero-shot learning). [12][13] propose to train classifiers for attributes as mid-level shared representation of classes. The approach exposed in[14] consists in learning a projection of images into a semantic word vector space learned from co-occurrences in large text corpus. Recently, Akata et al.[15] extend previous work by jointly learning image and label embedding. All those methods rely on a closed set of concepts and may be hard to scale up.

A vast majority of the literature attempts to increase algorithms performance on a close-world assumption through a variety of datasets. Although, in a scenario where robots must evolve in human environment unknown in advance, we also have to reason with generic and adaptable models. Learning that a fork has a high probability to be in a kitchen is good, but knowing *why* provides deeper information. This is the guiding idea behind our model presented in the following section.

II. Our Model

Our model is composed of three main units : Perception, Instance and Knowledge as can be seen in figure 1. The perception model is used to represent low-level sensorial

[1] Yohan Breux, Sebastien Druon and Rene Zapata are with the Computer Science, Robotics and Microelectronics Laboratory (LIRMM), University of Montpellier 161 rue Ada, 34095 Montpellier Cedex 5, France {breux, druon, zapata}@lirmm.fr



Id	Type	Args	Status
0	SEARCH_CLASS	Banana	COMPLETED
1	SEARCH_INSTANCE	My_cup	IN_PROGRESS

Fig. 1. Global representation of our model builded from a user request. Best viewed in color.

information whereas the knowledge model represents high-level semantic relations based on human natural language. The instance model links visual and geometric descriptions of real-world observations to semantic concepts.

A. Perception Model

The perception model takes as input raw data earned from sensors. It is responsible for generic scene segmentation, instance localization/tracking and concept detection. Scene segmentation can be done using unsupervised 3D point cloud segmentation algorithm [16] with saliency-based object detection on RGB images [17]. We choose the normalized output of fully convolutional layer from pretrained convolutional neural network (CNN) (AlexNet[6] and VGG[18]) as our visual descriptor.

This choice is motivated by the recent success of deep features and their good generalization[19] to concepts not necessarily appearing in the training phase. Moreover, the high capacity of such descriptors can also be used to represent individual instances. We modified the original AlexNet/VGG model following[20] to a fully convolutional network (FCN) for class and instance detection. In short, it returns 4096-d vectors associated to an overlapping grid of patches.

B. Knowledge Model

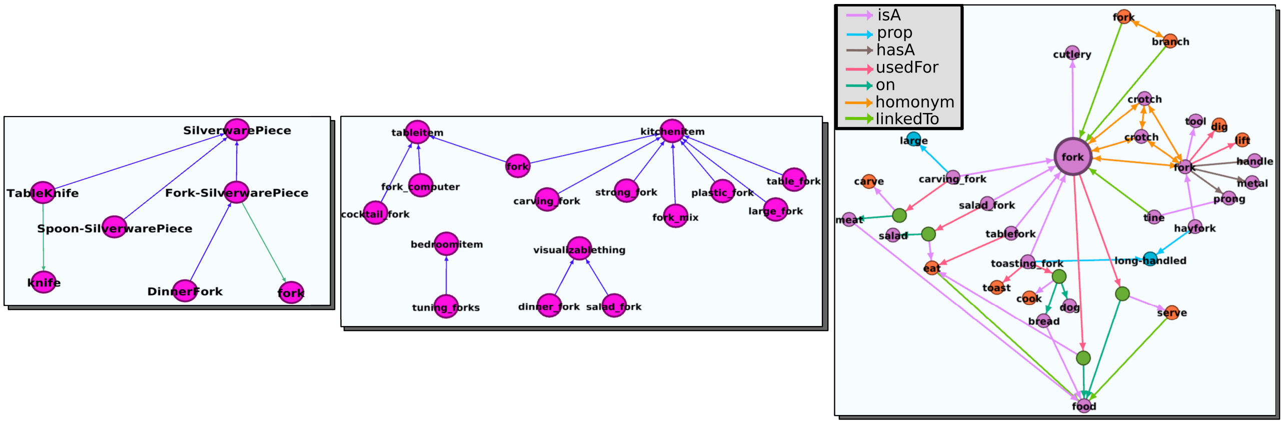
Our Knowledge model is defined as an ontological graph composed of concepts e.g. *fruit*, *cup* and relations such as *is-a* or *has-a*. Its purpose is to give the robot an abstract but yet practical representation of the world he lives in. In general, both manually (OpenCyc[21],KnowRob[9]) or automatically (NELL[22]) built ontologies cover a large spectrum of concepts but most part is not relevant for our robotic application, as for instance *emotion* or *mental event*. On the contrary, it lacks details on object description.

We choose to use a subset of Wordnet [23] to bootstrap our ontology, avoiding categories not relevant to environment understanding eg. *event*, *feeling*. Each concept in it is represented as a group of words (synonyms) identified with a unique id. Homonyms are thus well separated. We observed that, in dictionaries, objects are generally defined based on their appearance or/and by their function. This motivated our automated creation of ontology based on Wordnet glosses.

For instance, *cup* in Wordnet is defined as *"small open container usually used for drinking; usually has a handle"* but *handle* is not marked as a meronym (part) of *cup* and there is no relation with the *drinking* action. To improve upon Wordnet ontology, we automatically parse word definition using the natural language parser SyntaxNet[24]. The parsing result is obtained in a coNLL-X format [25] we use to analyze the definitions and to extract relations between concepts. Unlike knowledge framework such as KnowRob which focuses on leveraging semantic information to execute specifics tasks, we put the emphasis here on the knowledge base construction. As can be seen in figure 2, both KnowRob and NELL ontologies are rather shallow and are not fitted for physical object description. Note that our ontology can highlight object context eg. *food* in the case of *fork*.

Definition of the Knowledge graph : The created Knowledge base is represented by a mixed factor graph $G_K = \{V_K, E_K\}$. $V_K = V_K^C \cup V_K^F$ represents the set of vertices, which can be a concept (V_K^C) or a factorization (V_K^F) of concepts. There are two types of concepts: Object-Property V_K^O or Action V_K^A. Thus we have four kinds of vertex : concept of physical object or property $V_{concept}^O$, concept of actions $V_{concept}^A$, and those used to factorized object V_{factor}^O and action V_{factor}^A concepts.

A physical object is defined the majority of time either by its physical description, its function or both. For instance, a human recognizes a cup by its shape but also by its use ie to drink some liquids. On the other side, a thrash is better described by its function, as its physical appearance has a great variance. Following this statement, we choose to limit relations represented in our model to those related to how human defines physical objects. This naturally leads to make a formal separation between object-property and action concepts. In practice, concept vertices are defined by a unique identifier (Wordnet synsetId).



(a) KnowRob

(b) NELL

(c) Our generated ontology

Fig. 2. Comparison of knowledge subgraph centered on the *fork* concept for (a) KnowRob[9], (b) NELL[22] and (c) ours. Green empty nodes correspond to Factor vertices. Blue edges of (a) and (b) are *isA* relations. Green edges in (a) are *hasValue* relations, equivalent to Wordnet synsets (synonyms).

Factor vertices represent unnamed concepts as combination of another concepts. We can automatically represent specialized subclass of concepts which do not necessarily correspond to a named concept in human language. Doing so, we limit the number of concepts in our Knowledge model. In figures 2 and 3b, factor (resp. action, object, property) vertices are drawn in green (resp. orange, purple, blue).

Semantic relations are usually expressed as a triplet $< Subject, Predicate, Object >$. *Subject*, *Object* are nominal groups and *Predicate* generally a verb. Edges E_K of the ontology graph G_K can be of the following types :

- *isA* : expresses hyper/hyponym relation e.g. *isA(cup, container)*, *isA(paper cup, cup)*.
- *hasA* : represents part of objects e.g. *hasA(cup, handle)*
- *prop* : represents properties of concept, usually expressed by an adjective but can be noun in case of material : *prop(wheel, circular)*, *prop(desk, wood)*
- *usedFor* : represents the function of a concept which is expressed by a verb of action : *usedFor(cup, drink)*
- *on* : if such information is available, used to precise the object on which is applied the action e.g. *on(drink, tea)*.
- *linked-to* : represents other unknown relations between concepts with other concepts such as *linkedTo(hold, hand)*.
- *homonym* : They are obtained by linking concepts defined by a common word or synonym. They are substantial (see Table I).

Note that all edges are directed except for homonym relation. Each relation type corresponds to a subset of predicates e.g. *"have","consist of"* are predicates for the *has-a* relation.

The use of factor vertices is important because it prevents the creation of too much detailed concepts e.g *yellow skin* in *banana has a yellow skin*. Obviously, *yellow* can't be linked directly to *skin* so we use factor vertex as a proxy to link those two concepts. We want a knowledge model as abstracted as possible : we avoid direct representation of overdetailed concepts, keeping a clear border between instances and concepts. In practice, the ontology is stored as a list of Prolog clauses.

Creation of the Knowledge Graph from words glosses As previously said, we bootstrap the ontology graph using Wordnet ontology. We improve it by parsing Wordnet glosses with SyntaxNet and analyzing its output to extract relations on concepts. Our objective is to convert each sentence in the glosses into a formula in first-order logic. We consider *"A banana is a red, yellow or green fruit."* as an example sentence. Note that our model can still be used without Wordnet ontology by using another sources of information such as the first paragraph on Wikipedia pages. The principal advantage of Wordnet is to provide short and similarly structured definitions.

First, we search for Nominal Groups (*NG*) ie nouns with related determinant and adjective(s). We also search for constraints between NG through conjunction such as *"or"* and propagate conjunction repeated by a comma *","* as in *"a red, yellow or green fruit."*. In our *banana* example, we have four NGs : $NG_0 =$ *"A banana"*, $NG_1 =$ *"red fruit"*, $NG_2 =$ *"yellow fruit"* and $NG_3 =$ *"green fruit"*. The last three ones are constrained by a "OR" conjunction. We then link these NGs by constructing Semantic Relation Groups (*SRG*) which are triplet of the form $< NG_{subject}$, *Predicate*, $NG_{object} >$. Here we have $SRG_0 =< NG_0, isA, NG_1 >$, $SRG_1 =< NG_0, isA, NG_2 >$ and $SRG_2 =< NG_0, isA, NG_3 >$. We also propagate the constraint from the NGs to the SRGs, so that the three SRGs in our example are constrained by a "OR" conjunction. Finally we have the sentence representation in first-order logic : $SRG_0 \vee SRG_1 \vee SRG_2$. From each SRG, we construct one factor vertex for the NG subject and the NG object. In case where the NG contains only a noun, we directly construct a concept vertex. The type of edge (*isA, hasA*,...) between these two vertices is defined by the SRG predicate. We also consider verbs for the *usedFor* and *on* relations. An ontology graph build from the fork definition subgraph can be seen in Figure 3b. Table I resumes the global repartition of vertex/edge type in our ontology. We remark that homonymy relations are quite

TABLE I
GENERAL STATISTICS ON OUR ONTOLOGY CREATED FROM WORDNET GLOSSES.

	Type		Count	Ratio(%)
Vertex	*Noun*		23945	58.38
	Factor		9338	22.77
	Verb		5517	13.45
	Adjective		2219	5.41
Edge	*is-a*	Wordnet	21191	20.06
		Auto	15079	14.27
	linked-to		30379	28.76
	prop		17000	16.09
	homonym		13092	12.39
	has-a		3679	3.48
	use-for		3020	2.86
	on		2198	2.08

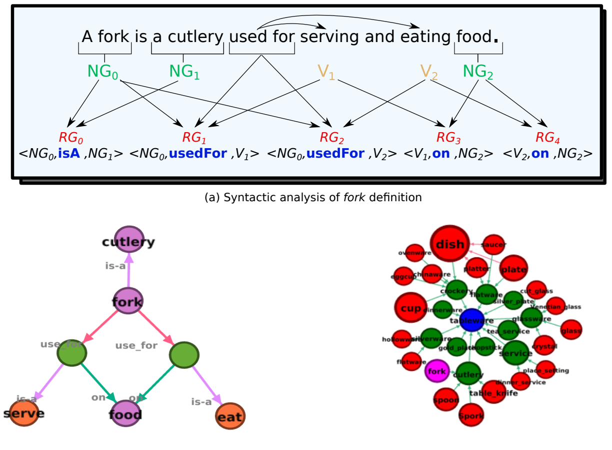
important : it is therefore necessary to keep them in our ontology for sense disambiguation. Besides, more than 40% of *isA* relations comes from our method.

C. Instance Model

Instances are physical realizations of semantic concepts. This model is thus here as a bridge between sensorial low-level data to generic high-level semantic knowledge. It is not only an abstraction bridge, but also a temporal one.

Indeed, unlike the perception model which works at the scale of a frame and knowledge model at an ”infinite” scale, instance model works on a limited slice of time. Hence time-related information can be extracted from the instance model such as kinematic relation between objects [26] or object co-occurrences. Mechanical information is crucial for manipulation and can’t be retrieve with static perception data. Data mining of knowledge base such as Wikipedia can give at best some vague information on the type of kinematic relation e.g. *”Doors normally consist of a panel that swings on hinges on the edge”* but not detailed enough to be usable by the robot. Besides, other works based on multilayer representations (section I) put instances and concepts in the same conceptual layer. Here, we clearly separate concepts and instances because of this fundamental difference : a concept lives ”forever” and is generic whereas an instance lives temporarily and is specific to the current robot context.

Definition : Formally, our instance model is represented by a directed graph $G_I = \{V_I, E_I\}$ where the set of vertices V_I represents the instances and E_I represents relations between them. Each instance $v \in V_I$ is defined by a set of observations $O = \{o^{t_0}, \ldots, o^{t_n}\}$ with $o^{t_i} = \{t_i,\ H^{t_i},\ KF^{t_i}\}$, a graph $G_v = \{V_v, E_v\}$ which links the instance to concepts in the knowledge model i.e. $V_v \subset V_K$. t_i is the time stamp of the observation, $H^{t_i} \in \mathscr{M}_{4,4}(\mathbb{R})$ is the pose matrix at this instant, KF^{t_i} is the corresponding key frame (image patch and visual descriptor). The graph G_v uses the same formalism as the knowledge model. It is extended by taking in account agent ownership through the relation *belong-to* eg. *belong-to(My_key, John)*. Those relations are created from the identity of the interacting user in case of possessive pronoun (*my_cup*) or directly (*the cup of john*). Instances can be created and updated from detections of the perception model but also from information provided by the users. The cup instance showed in the Figure 1 was first created from the semantic information provided by the user request, and its visual properties were updated after feedback from the perception model. Thus, we can represent and reason with unseen instances. In this case, $O = \emptyset$.



(a) Syntactic analysis of *fork* definition

(b) Generated subgraph

(c) Context subgraph from the concept *fork*

Fig. 3. Ontology construction from the Wordnet definition of fork : (a) Syntactic analysis and (b) corresponding ontology subgraph. (c) is the context subgraph of *fork*. We assigned the same color to nodes at the same relative level from the top local node. Red (resp. green) edges correspond to link created by our method (resp. Wordnet).

Currently, edges E_I only represent the following spatial relations between instances (relative to the robot or speaking agent) : $\{$*behind, in front of, to the left/right of, on top of, under, inside*$\}$. Those edges can be created from user-provided information in natural language such as *”My key is in the box”* which translates into the relation $< my\_key,\ inside,\ isA(A, box) >$, even if instances have not been observed yet. It can be also inferred from perception data through their relative positions if both instances have been seen.

We only store relations provided by the user and not from observations as they are time-dependent. We infer it on demand e.g. if user asks the robot ”Where is the key ?”, it will simply infer ”the key is on the table and behind the cup” from the current poses of the instances. This information is stored in a temporary and separate database file, only used by the request which generated it.

Instance creation : At each frame, the perception model outputs visual instances to add to our model. To avoid duplicate, we have to compute correspondences with already known elements. We simply merge them based on their cosine similarity if their visual descriptors are available. When only semantic information such as *isA(my_cup,cup), prop(my_cup,red)* is known, we merge visual instances which comply with the description. Finally, an instance is added if no correspondence has been found.

III. TASK CREATION

On top of the three main units of our model, we have an entity creating and managing tasks generated from user requests. They are syntactically analyzed as previously explained and are converted to dynamic Prolog predicates. Consider the two requests *"Bring me the fork behind the cup"* and *"Bring me the fork behind my cup"*. We suppose that the *my_cup* instance is visually known. Following an adaptation of the Wordnet glosses parsing method explained in section II-B, we generate the following predicates for each request :

$$mp(A,B) :\text{-}\ isA(A,fork),\ isA(B,cup),\ behind(A,B). \tag{1}$$
$$mp(A) :\text{-}\ isA(A,fork),\ behind(A,my\_cup),\ isDet(my\_cup).$$

where the *isDet/1* predicate stands for *isDetected*. In practice, we also take in account subclasses of concept c up to depth N by replacing *isA* with a predicate *isDescendant/3*.

We use here the concept of computable classes/properties from [9] to compute some relations on-demand based on the current state of the robot model. Here, *behind/2* and *isDet/1* are examples of such predicates. Each concept c in the request starts a new detection thread with an independent queue of instances to check. This queue is initialized with previously seen but unidentified instances. It is then filled with instances newly created or being updated with new semantic or visual information. Instances I detected as a concept c are then updated by adding a clause *isA('I',c)* or *prop('I',c)* (depending on the type of concept c) to their semantic representation. Note that if the task can't be solved in the current state, it keeps running in background. Its status is checked up every time a new instance is created or updated.

For instance localization, we add a layer to our FCN which computes the cosine similarities between the instance and each patch of the image. Tracking is simply done by relocating the object at each frame (tracking-by-detection). Note that we also developed a tracking method inferring movement up to a similarity in SIM(2) based on CNN features but it is out of the scope of this paper. Concept detection is done with a *one-vs-all* Random Forest [27] learned for each concept using external data if available eg. ImageNet[28]. We added another layer to our FCN which computes the classifier score at each patch. Corresponding heat map can be seen in the Perception Model representation of the figure 1. Our detection scheme, while giving satisfying results for the proof-of-concept of our model architecture, is not optimized in any way : the focus is put on the exploitation of those detections for the semantic modeling of the robot environment.

The object requested is an hint for the presence of contextually related object. We exploit this by loading upper concepts (currently in the *isA* relation sense) C_n at depth n from c and by extracting the subgraph with each subconcept of C_n at depth at most n. For instance, when $n = 2$, it consists in first taking C_n as the "grand-parent" concepts and then adding their children and grandchildren. An example of such subgraph is given in figure 3c for the concept $fork$. Those concepts are passively searched as a background task. However, even for small n, this graph can be quite huge. To deal with this, we envisage to sort the concept search queue based on another similarity measure based on Glove word vector[29].

IV. Use case

It is rather hard to quantitatively assess the performance of our model, as it depends heavily on the algorithms used for the vision and the semantic analysis. We propose to showcase a simple example of application. We consider a scene consisting of several objects (two cups, one toy truck and a fork) on top of a table. Here, we don't use general scene segmentation algorithm as it requires post-processing and can introduce spurious detection. We detect the table plane using a RANSAC based plane fitting on the 3D point cloud. Objects above it are segmented into instances. The initial 2D object masks are further refined using GrabCut[30]. The procedure is summarized in figure 4. At the beginning, the knowledge model is initialized as explained in section II-B. Instances are created from the scene segmentation with only visual information available.

The user asks the robot the following request : *"Bring me the cup to the left of the truck"*. This starts a new task with the following predicate:

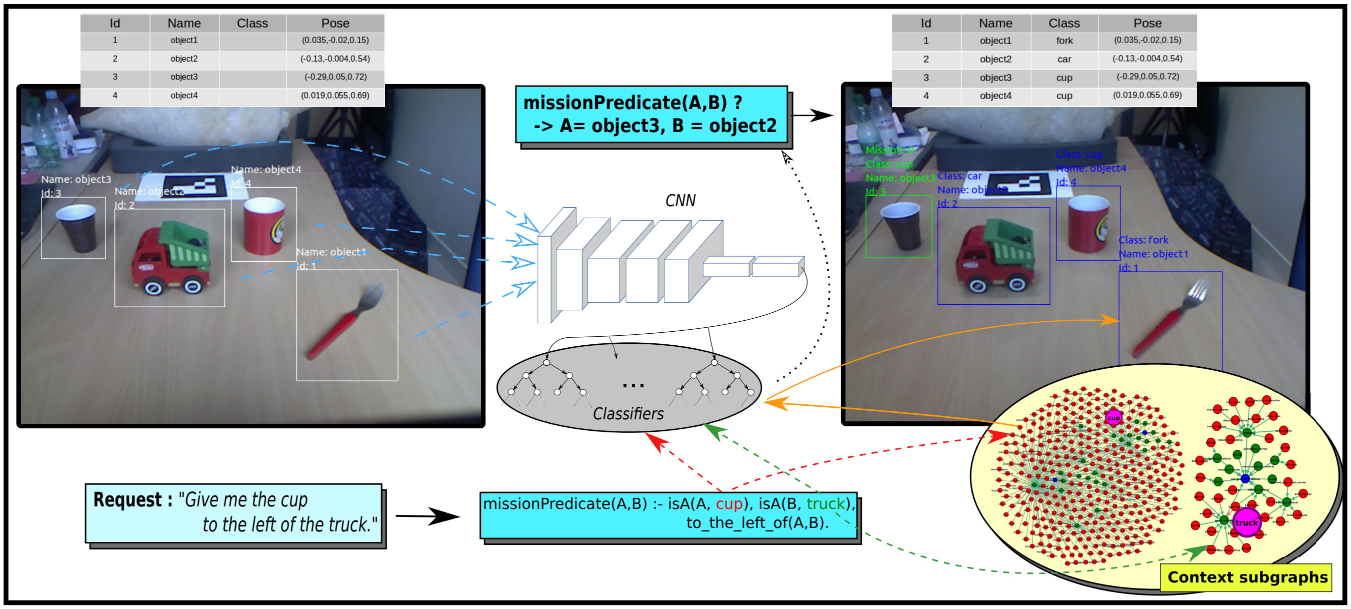
$$mp(A,B) :\text{-}isA(A,cup),\ isA(B,car),\ to\_the\_left\_of(A,B). \tag{2}$$

The car and cup detections are run in two separate threads. For each new detection, the system checks if the task predicate holds for some instances. Finally, after finding the required cup instance, we extract its context subgraph as shown in figure 3c. Note that instances were found through scene segmentation before any user request, so that object detection with FCN was not required.

V. Discussion

We have presented a general multi-modal framework for environment representation/understanding in the context of autonomous robotics. We tackle the problem of semantic knowledge building from a robotic viewpoint. Majority of semantic related researches uses generic ontology built by experts. However, they contain only few detailed descriptions of physical concepts. In particular, our work mainly differs from RoboSHERLOCK and KnowRob frameworks as we put the focus on automatic object-oriented ontology construction. This motivates our use of dictionary e.g. Wordnet as source of semantic information, as it defines concepts with physical description and functions (usage). Besides, definitions are rather principled in their structure making the parsing easier and more reliable. Another advantage is that we can easily integrate in the same way information given by the user in natural language about concepts and instances.

We are currently working on several areas of improvement: integration of feedback with the user when task solution can't be decided, development of method for instance class inference and integration of the system on a mobile robot.



Id	Name	Class	Pose
1	object1		(0.035,-0.02,0.15)
2	object2		(-0.13,-0.004,0.54)
3	object3		(-0.29,0.05,0.72)
4	object4		(0.019,0.055,0.69)

Id	Name	Class	Pose
1	object1	fork	(0.035,-0.02,0.15)
2	object2	car	(-0.13,-0.004,0.54)
3	object3	cup	(-0.29,0.05,0.72)
4	object4	cup	(0.019,0.055,0.69)

missionPredicate(A,B) ?
-> A= object3, B = object2

Request : *"Give me the cup to the left of the truck."*

missionPredicate(A,B) :- isA(A, cup), isA(B, truck), to_the_left_of(A,B).

Fig. 4. Processing of a user request as explained in section IV. We start from the right side with a knowledge model created from our Wordnet glosses analysis. The scene is segmented into four instances. The user request is then analyzed and automatically converted to a dynamic Prolog predicate. This triggers classification of required concepts. Once the task objective found, context subgraphs are created and related concepts searched (here *fork* from the *cup* context).

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, M. Press, Ed. MIT Press, Cambridge, 2005.

[2] T. Whelan, S. Leutenegger, *et al.*, "Elasticfusion: Dense slam without a pose graph," *Proc. Robotics: Science and Systems, Rome, Italy*, 2015.

[3] F. Keith, N. Mansard, *et al.*, "Optimization of tasks warping and scheduling for smooth sequencing of robotic actions," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 1609–1614.

[4] S. Mehta and T. Burks, "Vision-based control of robotic manipulator for citrus harvesting," *Computers and Electronics in Agriculture*, vol. 102, pp. 146 – 158, 2014.

[5] N. Sünderhauf, F. Dayoub, *et al.*, "Place categorization and semantic mapping on a mobile robot," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 5729–5736.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 3515–3522.

[8] D. Lang, S. Friedmann, *et al.*, "Semantic mapping for mobile outdoor robots," in *14th IAPR International Conference on Machine Vision Applications*, 2015, pp. 325–328.

[9] M. Tenorth and M. Beetz, "Representations for robot knowledge in the knowrob framework," *Artificial Intelligence*, vol. 247, pp. 151–169, 2017.

[10] C. Matuszek, J. Cabral, *et al.*, "An introduction to the syntax and content of cyc." in *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006, pp. 44–49.

[11] M. Beetz, M. Tenorth, and J. Winkler, "Open-ease," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1983–1990.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958.

[13] Y. Sun, L. Bo, and D. Fox, "Attribute based object identification," in *ICRA*, 2013.

[14] R. Socher, M. Ganjoo, *et al.*, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, 2013, pp. 935–943.

[15] Z. Akata, F. Perronnin, *et al.*, "Label-embedding for image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

[16] S. C. Stein, M. Schoeler, *et al.*, "Object partitioning using local convexity," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304–311, 2014.

[17] J. Zhang and S. Sclaroff, "Exploiting surroundness for saliency detection : A boolean map approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 889–902, May 2016.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] A. S. Razavian, H. Azizpour, *et al.*, "Cnn features off-the-shelf: an astounding baseline for recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 802–813, 2014.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[21] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[22] T. Mitchell, W. Cohen, *et al.*, "Never-ending learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[23] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[24] D. Andor, C. Alberti, *et al.*, "Globally normalized transition-based neural networks," *arXiv preprint arXiv:1603.06042*, 2016.

[25] S. Buchholz and E. Marsi, "Conll-x shared task on multilingual dependency parsing," in *Proceedings of the Tenth Conference on Computational Natural Language Learning*, 2006, pp. 149–164.

[26] J. Yan and M. Pollefeys, "A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 865–877, 2008. [Online]. Available: