

Learning Context on a Humanoid Robot using Incremental Latent Dirichlet Allocation

Hande Çelikkanat, Güner Orhan, Nicolas Pugeault, Frank Guerin, Erol Şahin, and Sinan Kalkan

Abstract—In this article, we formalize and model context in terms of a set of concepts grounded in the sensorimotor interactions of a robot. The concepts are modeled as a web using Markov Random Field, inspired from the concept web hypothesis for representing concepts in humans. On this concept web, we treat context as a latent variable of Latent Dirichlet Allocation (LDA), which is a widely-used method in computational linguistics for modeling topics in texts. We extend the standard LDA method in order to make it incremental so that (i) it does not re-learn everything from scratch given new interactions (*i.e.*, it is online) and (ii) it can discover and add a new context into its model when necessary. We demonstrate on the iCub platform that, partly owing to modeling context on top of the concept web, our approach is adaptive, online and robust: It is adaptive and online since it can learn and discover a new context from new interactions. It is robust since it is not affected by irrelevant stimuli and it can discover contexts after a few interactions only. Moreover, we show how to use the context learned in such a model for two important tasks: object recognition and planning.

Index Terms—Context, Situated Concepts, Latent Dirichlet Allocation

I. INTRODUCTION

We tackle the problem of using contextual information to improve the performance of a cognitive robot, specifically in perception and planning. We define context as the totality of the information characterizing the situation of a cognitive system; *e.g.*, it can include objects, persons, places, and temporally extended information related to ongoing tasks, but also information not directly related to these tasks [1].

Hande Çelikkanat, Güner Orhan, and Sinan Kalkan are with KOVAN Research Lab, Department of Computer Engineering, Middle East Technical University, Ankara, TURKEY, E-mail: {hande.guner.orhan,skalkan}@ceng.metu.edu.tr

Erol Şahin is with KOVAN Research Lab, Department of Computer Engineering, Middle East Technical University, Ankara, TURKEY, and with Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, E-mail:erol@ceng.metu.edu.tr

Nicolas Pugeault is with College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK E-mail: n.pugeault@exeter.ac.uk

Frank Guerin is with Department of Computing Science, University of Aberdeen, UK, E-mail: f.guerin@abdn.ac.uk

There is ample evidence that natural cognitive systems modulate their response to stimuli depending on a wide range of other, seemingly irrelevant stimuli (context). Yeh and Barsalou [2] demonstrated in a series of experiments that human subjects perform better at a variety of cognitive tasks when taking context into account. This is because context can promote relevant information and behaviors, while suppressing irrelevant ones, based on statistical likelihood of various objects and behaviors in a certain setting. A concept such as a chair does not exist in isolation, but is associated in memory with other concepts that also occurred in the concrete situations where the concept was previously encountered by the system; *e.g.*, the chair’s location, office or living room, but also the actions performed with the chair, such as reclining. These connections between concepts in memory then allow the system, when detecting a concept, to draw inferences about connected concepts; this is illustrated in Figure 1. The activation of a ‘chair’ concept promotes related objects such as ‘table’ and ‘lamp’, and draws inferences on their plausible position. Furthermore, a ‘living room’ concept will promote chair properties such as ‘large’ and ‘soft’, rather than ‘small’ and ‘hard’ (contrary to, *e.g.*, a ‘classroom’ concept). Similarly, actions usually associated with the active concepts, such as ‘sitting’ in our example, are promoted, whereas unlikely actions (‘lifting’) are suppressed. In sum, what forms context depends on the concept of interest, and consists of all other concepts present at the same time. Through experience, a cognitive system forms an interconnected network of related concepts and situations that allows efficient filtering of context and inference.

In this article, motivated by the concept-based nature of human cognition, we formulate context to be the set of active concepts in the scene, rather than relating it directly to raw sensorimotor data. For this, we employ a widely-used topic model in computational linguistics, called Latent Dirichlet Allocation (LDA), and apply it to the active concepts in the scene. For modeling the concepts, we use a concept-web model that we developed using Markov Random Fields in our previous work [3].

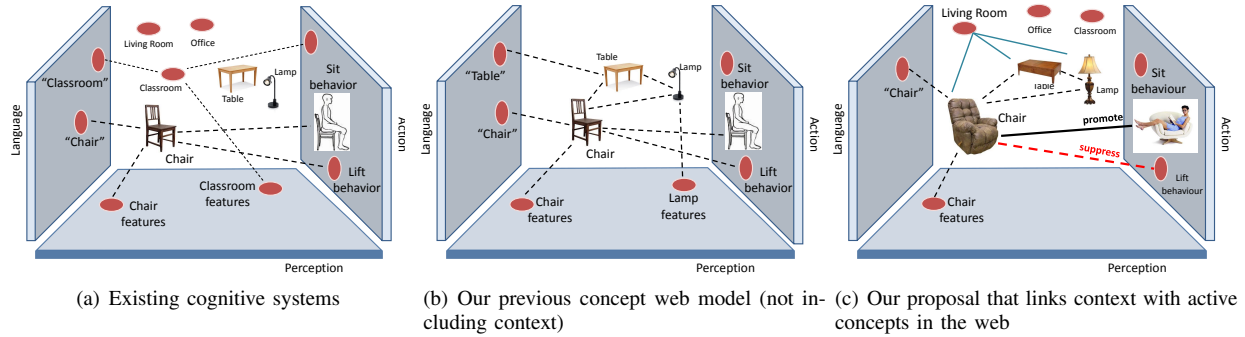


Fig. 1: (a) Existing cognitive systems have concepts which have links to perceptual features and motor actions which were programmed by a designer or trained in context-free environments. (b) The concept web model that we developed in our previous study [3]: A densely connected concept web connecting perception, action and language; however, there is no notion of context in this model. (c) We propose a system that learns in context the links between concepts and sensorimotor primitives, based on the statistics of its interactions in real-life environments. For clarity, only a few links and concepts are shown. [Sub-figures (a) and (c) adapted from [1]]

We demonstrate how context can be learned and used by such a model for several tasks by a humanoid robot.

A. Context in Cognitive Science and Robotics

It is a matter of consensus across fields that context processing is an essential part of embodied cognition (*e.g.*, psychology [2], language [4]–[6], AI [7], robotics [8], [9] and computer vision [10], [11]). Schank and Abelson [7] argued that reasoning about situations in daily life relies on “scripts” that inform reasoners about the prototypical features of these situations. A restaurant, for example, tends to come with a menu, dishes, a waiter, a chef, and so on. This work has gone on to influence today’s formal ontologies. Probably the earliest research on context focused on linguistic phenomena, studying how the understanding of an expression (*e.g.*, a personal pronoun like “it”) is affected by the rest of the sentence or text [5]. Later research applied these ideas to other aspects of communication, including speech (*e.g.*, pitch accent) and body language (*e.g.*, [12]). Even more drastically, the notion of a context has been extended to all symbolic systems (*e.g.*, [13]). Perhaps most notably, McCarthy [9] proposed the rectification of context in classical (logical) AI, arguing that Artificial Intelligence needs to put the notion of a context centre stage. In McCarthy’s view, intelligent machines “must construct or choose a limited context containing a suitable theory whose predicates and functions connect to the machine’s inputs and outputs in an appropriate way” [14]. This work gave rise to a wave of theoretical work focusing on issues like the problem of “lifting” information about one knowledge base to another.

Work in all these traditions continues to inspire Cognitive Science and AI. But times have changed: the rise of embodied cognition theories in the 90’s, for instance, has offered a different perspective on context, based on a perceptual and action-based rather than symbolic approach [8]. This perceptual perspective is particularly relevant for robotics, where contexts typically need to be acquired from perception (*i.e.*, they cannot be programmed in advance). Barsalou, for example, has advocated the necessity for concepts to be situated [2], [15]; in other words, for an abstract concept to be related to concrete contexts. Coventry *et al.* [6] studied the difference between geometric and functional contexts in the use of spatial prepositions (“over” vs. “above”) and of linguistic quantifiers (“few” vs. “many” vs. “several”).

One striking example concerns work on affordances. Until some years ago, behavioral studies on affordances tended to highlight the fact that affordances are automatically activated, independently from the kind of task and context. This was shown through compatibility effects in which, for example, size resulted as a relevant dimension even if the task did not require subjects to judge objects on the basis of their size but, instead, of their category (*e.g.*, [16]). Recent evidence has questioned this view of affordances, showing that the activation of affordances is modulated by the physical and by the social context. A variety of studies have shown that the embedding in a context given by a specific scene (*e.g.*, [17]–[19]) or by the presence of other objects influences affordances activation (*e.g.*, [20]–[22]).

Robotics has achieved significant success in terms of both theory and applications in the past five decades [23]; however, research involving context has focused on the

environmental aspect only, *e.g.*, in scene interpretation [10], urban search for rescue tasks [24], home security [25] and elderly people’s living environments [26], object recognition in daily activities [27], [28], and trying to fulfill possibly incomplete natural language instructions of humans [29].

Of all these works, [27], [29]–[31] stand out for attempting more explicit utilization of contextual information. In an attempt to provide a common representation for multiple agents sharing knowledge, Padovitz *et al.* [30] and Mastrogiovanni *et al.* [31] define context as explicit and crisp conjunction rules of a priori known predicates for each context. However, these representations are therefore both overly restrictive to conjunctive phrases, and also naive in the sense that the programmer is assumed to know how to encode each context rule in conjunctions. The assumption of existing and perfectly known conjunctive rules for each context is indeed a strong one, and would be over-sensitive, for instance, to the failures in the sensing of any one of the necessary premises, or to the emergence of unpredicted contexts. Indeed, the main emphasis in both works is more on providing a common ground for facilitating information sharing among multiple agents, rather than elaborating on contextual representation. Anand *et al.* [27] define and use a more restricted notion of *spatial* context, limited to the spatial relationships between canonical placements of objects in the environment: A computer is usually found *on-top-of* a table, and this information can be used facilitating object search and labeling (see [18], [19] for similar behavior in humans). On the other hand, Misra *et al.* [29] treat context as multiple-choice values of the states of known objects in the environment (*i.e.*, microwave door is *closed* or *open*), used afterwards for completing missing information in natural-language commands of humans. Given an incomplete command from a naive human partner, the robot can therefore use this contextual background to complete missing *implied* links to achieve these commands. An example might be, when commanded to “heap up the milk”, reasoning that “the milk is currently *in the carton*, but it needs to be *on the oven*, and the oven must be in the *on state*.”

In computer vision, the notion of context has grown in prominence over the last decade, both explicitly and implicitly [10]. Explicitly, the study of visual gist [32] showed that holistic encodings of the visual input could carry a large amount of information, allowing scene identification [32], [33], urban scene detection [34], and autonomous navigation [35], as well as action recognition [36], object categorization [37], and detection [11]. Implicitly, the now popular data-driven, machine

learning-based approach to vision led to algorithms that efficiently extract all predictive information from the visual data, making heavy use of context to reach high performance (see [38] for a criticism).

A promising approach for developing an explicit model of context seems to be Latent Dirichlet Allocation (LDA), a hidden topic model developed for categorizing documents of large text corpora [39]. As a robust, unsupervised Bayesian method, it has been utilized as well in a variety of applications, ranging from fraud detection [40] to the identification of functional regulatory networks of miRNA-mRNAs [41]. Since the method provides the statistical tools for discovering hidden topics in unsupervised data, we propose that it can also be used for modeling context. In fact, ours is not the first attempt to use LDA formulation in robotics: It has been utilized successfully for object categorization from multi-modal sensory data [42]–[44], and for autonomous drive annotation [45]. However, our work is the first to use LDA for modeling context in robotics.

B. This Study

We see in existing works various piecemeal efforts to tackle particular facets of context in specific domains. In contrast, following the intuitions of [2], we argue that a principled approach is needed to learn, represent and process context in a developing cognitive system.

We study how we can equip a robot with the ability of detecting and using context, *e.g.*, in object recognition and planning as proof of concept. The novelty and contributions of our approach can be summarized as:

- Formalization of context on a robot using Latent Dirichlet Allocation (LDA). To the best of our knowledge, this is the first time that context is tackled systematically, as a separate entity but also in direct relation with other conceptual entities, in a robotics scenario. In contrast to the attempts of Anand *et al.* [27] and Misra *et al.* [29] for using contextual information, which do not introduce a general model of context, resorting to defining it in terms of predefined geometric relations or object-part states, we formalize context to develop an adaptive system in which contextual information can be extracted, represented, and utilized explicitly.
- We provide an incremental extension of LDA so that (i) it does not re-learn everything from scratch given new sensorimotor interactions (*i.e.*, it is on-line) and (ii) it can discover and add a new context into its representations when necessary.
- Finally, motivated by our findings of the computational advantages in [3], we propose applying

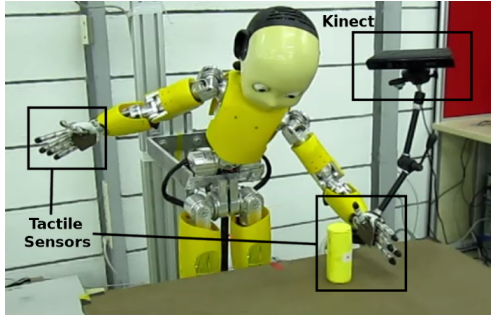


Fig. 2: The setup used in the experiments. iCub senses the environment with tactile sensors, a microphone and a Kinect.

LDA on a concept web representation of the scene, instead of its raw features directly. We subsequently demonstrate how learning context from high-level concepts, instead of raw features, is easier and achieves higher performance.

The current article extends an earlier version of our work [1], where preliminary results on integrating context were presented using the standard LDA with an ad hoc concept web. The current article differs in the following aspects: (i) The LDA is extended in order to make it online and incremental. (ii) The ad hoc concept web is replaced with a formally developed concept-web modeled using Markov Random Fields. (iii) A more extensive evaluation of the system is presented.

The current article uses the concept web model that we developed in [3]. This previous work introduced a concept web model and showed why it is important and useful. However, the current work goes beyond that and integrates context on top of this model, to demonstrate how context can be learned and used by a robot.

II. EXPERIMENTAL SETUP

We conduct our experiments using the iCub humanoid robot platform [46] (Figure 2). iCub has tactile sensors in each fingertip to detect the degree of grasping of an object and collect haptic information about the its hardness. Joint encoder values are used for collecting the proprioceptive information about the hand and the arm status. A Kinect device is used to get 3D visual information from the environment. iCub also has an external microphone to record the sound of objects.

A. Object Set

We have an object set \mathcal{O} of 60 objects, arbitrarily divided into a training set of 45 and a test set of 15 objects. The training objects are labeled via supervision as belonging to one of the 6 noun categories, $\{box, ball,$

$cylinder, cup, plate, tool\}$ (Figure 3), and one of the two adjectives in each 5 dichotomic adjective pair, $\{hard \times soft, noisy \times silent, tall \times short, thin \times thick, round \times edgy\}$ (Figure 4). The mapping between nouns and adjectives is not 1-to-1; e.g., a box can be soft or hard, silent or noisy etc. Table I depicts these co-occurrences.

B. Behaviors

We have a repertoire of 13 behaviors, $\{grasp, push left, push right, push forward, push backward, move left, move right, move forward, move backward, drop, throw, knock down, shake\}$, which are performed via hard-coded scripts on perceived objects with variable positions and poses. To ensure realism, some objects are (assumed to be) fragile and certain behaviors are not applied on them: We prevent iCub from *dropping, shaking, throwing, knocking* and *pushing* plates and cups. We also refrain from *pushing* balls, since they tend to roll down and disappear from the table. Table II shows the allowed behaviors for each noun category.

C. Features and Data Collection

iCub interacts with each object $o \in \mathcal{O}$ as follows:

1. The object o is placed on the table to an arbitrary location.
2. iCub “looks” at the object (*i.e.*, takes a 3D snapshot using the Kinect sensor) and extracts the initial visual features \mathbf{e}_v .
3. For each allowable action on the object (Table II):
 - 3.1. iCub executes the action on the object.
 - 3.2. If the *grasp* behavior is in progress, haptic (\mathbf{e}_h), and proprioceptive (\mathbf{e}_p) features are collected.
 - 3.3. If the *shake* behavior is in progress, audio (\mathbf{e}_a) features are collected.
 - 3.4. iCub takes a second 3D snapshot and extracts the final visual features \mathbf{e}'_v .
 - 3.5. The object is placed to a different initial position (to allow possible variability) by a human supervisor, before proceeding with the next action.

Table III lists the features used by iCub in this study. The first 6 visual features are basic position information and three dimensional properties of the object, and the next 40 features are the zenith and azimuth normal vectors of each point on the object. In addition to the normal information, we use histogram of shape index values. Shape index [47] is essentially a representation of the local surface type, calculated from the maximum and minimum principal curvatures (Q_1, Q_2 , respectively) of the point as follows: $\frac{Q_1 + Q_2}{Q_1 - Q_2}$.



Fig. 3: The objects used in the experiments, divided to each noun category.



Fig. 4: The objects for each adjective category.

TABLE I: The co-occurrences of noun and adjective labels for the dataset. Numbers denote the number of objects (out of 60) belonging to both categories.

	Hard	Soft	Noisy	Silent	Tall	Short	Thin	Thick	Round	Edgy
Box	2	14	2	14	0	16	0	16	0	16
Ball	3	7	7	3	0	10	1	9	10	0
Cylinder	14	0	5	9	10	4	9	5	14	0
Cup	11	0	1	10	0	11	0	11	11	0
Tool	5	0	5	0	5	0	0	5	5	0
Plate	4	0	0	4	4	0	0	4	4	0

TABLE II: The set of behaviors applicable for each object. A: Applicable; NA: Not-Applicable

	Push (Left, Right Forward, Backward)	Move (Left, Right Forward, Backward)	Drop	Grasp	Shake	Knock down	Throw
Box	A	A	A	A	A	A	A
Ball	NA	A	A	A	A	A	A
Cylinder	A	A	A	A	A	A	A
Cup	NA	A	NA	A	NA	NA	NA
Tool	A	A	A	A	A	A	A
Plate	NA	A	NA	A	NA	NA	NA

The following 13 are auditory features (\mathbf{e}_a) used to determine whether an object produces sound when interacted with. We use MFCC (Mel-Frequency Cepstrum Coefficients) on the raw audio data, yielding a set of 13-feature vectors. As features, we use the differences between the maximum and minimum values of each vector.

Haptic and proprioceptive features (\mathbf{e}_h and \mathbf{e}_p) are obtained from the index finger of iCub only. They are collected through the grasping action, and encode the difference between initial and final sensor readings for haptic/proprioceptive data, the minimum and maximum readings, and also the mean, variance, and the standard deviation values.

The concatenation of these features ($\mathbf{e}_v, \mathbf{e}_a, \mathbf{e}_h, \mathbf{e}_p$) is called an *entity feature vector* and is denoted by \mathbf{e} . Each object is described by an entity feature vector. For describing behaviors, we use *effect feature vectors*, denoted by \mathbf{f} , capturing the effect of a behavior on an object. They give the difference between the visual feature of the object before and after a behavior is applied, obtained by $\mathbf{f} = \mathbf{e}'_v - \mathbf{e}_v$. See Figure 5 for an illustration.

TABLE III: The visual, audio, haptic and proprioceptive features extracted from the interactions of the robot.

Feature Type	Feature	Position
Visual (\mathbf{e}_v)	Position:(x, y, z)	1-3
	Object dimensions:($width, height, depth$)	4-6
	Normal zenith histogram bins	7-26
	Normal azimuth histogram bins	27-46
	Shape index histogram bins	47-66
Audio (\mathbf{e}_a)	13 bins of MFCC (max - min)	67-79
Haptic (\mathbf{e}_h)	Change for index finger	80
	Min values for index finger	81
	Max values for index finger	82
	Mean for index finger	83
	Variance for index finger	84
	Standard deviation for index finger	85
Proprioceptive (\mathbf{e}_p)	Change for index finger	86
	Min values for index finger	87
	Max values for index finger	88
	Mean for index finger	89
	Variance for index finger	90
	Standard deviation for index finger	91

D. Contextual Setting

Our experimental setting is comprised of three contexts, *Kitchen*, *Playroom*, and *Workshop*. Some concepts in our framework occur in certain contexts, such as plates and cups existing in a Kitchen, balls and boxes occurring in a Playroom, as so on. Notice that this tendency is mostly a characteristic of noun concepts, which have more clear-cut divisions into contexts. On the contrary,

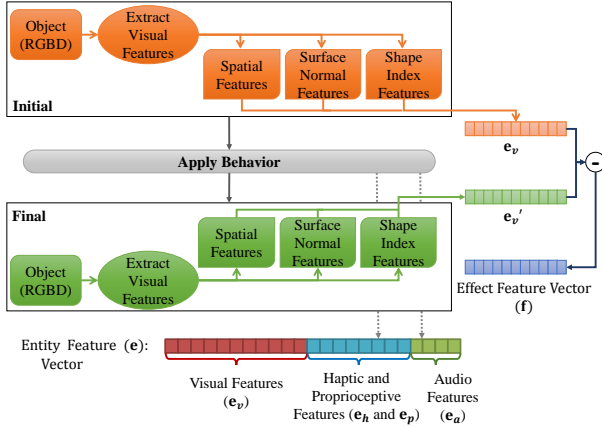


Fig. 5: Extraction of entity (\mathbf{e}) and effect (\mathbf{f}) feature vectors. \mathbf{e}_v and \mathbf{e}'_v are the initial and final visual features of the object before and after a behavior’s execution. $\mathbf{f} = \mathbf{e}'_v - \mathbf{e}_v$.

some concepts are so general that they do not have such clear-cut divisions. This is a characteristic of most adjective concepts in our setting: Adjectives such a round or tall are so generic that they are not limited to certain contexts. Table IV summarizes the prevalent concepts of the three contexts.

III. A CONCEPT WEB USING MARKOV RANDOM FIELD

In our system, context is formalized over a set of concepts that are extracted from the scene, and represented in a densely-connected web structure, called the concept web [3]. Since this web is central to our model, before continuing with the exact formalization and use of context in the system, we briefly describe the extraction of relevant concepts from a scene, and the formation of the concept web. We describe a framework consisting of three kinds of concepts: Noun concepts $\mathbb{N} = \{box, ball, cylinder, cup, plate, tool\}$, adjective concepts $\mathbb{A} = \{hard \times soft, noisy \times silent, tall \times short, thin \times thick, round \times edgy\}$, and verb concepts $\mathbb{V} = \{grasp, push left, push right, push forward, push backward, move left, move right, move forward, move backward, drop, throw, knock down, shake\}$.¹ Before evaluating each scene in terms of its context, the robot views and possibly interacts with the objects, makes initial predictions about the concepts associated with the scene, and then builds a web of these concepts to make use of their related semantics.

¹Note that “verb concepts” do not have to correspond to the behavior set in a 1-1 manner: A verb concept can be associated with multiple behaviors, for instance, provided that all of these behaviors produce the same effect [48], although this is not the case in this study.

TABLE IV: Used contexts and their prevalent concepts.

Kitchen			Playroom			Workshop		
cup	short	thin	ball	edgy	silent	tool	edgy	tall
plate	hard	thick	box	soft	thick	cylinder	hard	thin
round	silent		round	noisy		round	silent	thick

A. Reasoning with Individual Concepts

The initial task of the robot is to predict the individual concept(s) that are related to an object in its environment. This mapping of the world from raw features to a concept can be learned in a variety of manners, *e.g.*, using Support Vector Machines, k-Nearest Neighbors, Neural Networks, etc. In this work, we adopt a prototype-based approach [49], [50] following previous work [48], [51]; however, this choice is not central to the rest of the article; any method that provides a measure of similarity to a category from raw features is sufficient for this part. For a review of alternative representation schemes, the reader may for instance refer to [52]–[54].

In our framework, we describe the noun (\mathbb{N}), adjective (\mathbb{A}) and verb (\mathbb{V}) concepts in terms of their prototypes, which we learn from accordingly labeled interactions during the training phase (Figure 6(a), see Appendix for more detail). These prototypes are learned from the entity feature vectors \mathbf{e} and effect feature vectors \mathbf{f} during training, and summarize which features are highly relevant for the concept (need to be strongly positive, strongly negative, or close to zero), and which are irrelevant for the concept. During execution, an incoming instance is compared with the concept taking into account the concept prototype: The Euclidean distance $D(c, \mathbf{x})$ between the incoming feature vector \mathbf{x} and the concept prototype of c is calculated by disregarding the irrelevant features of the concept:

$$D(c, \mathbf{x}) = \frac{1}{|\mathcal{R}_c \setminus \mathcal{R}_c^*|} \sqrt{\sum_{i \in \mathcal{R}_c \setminus \mathcal{R}_c^*} (\mathbf{x}^i - \mu_c^i)^2}, \quad (1)$$

where \mathcal{R}_c^* is the set of indices that are not relevant for concept c ; \mathbf{x}^i is the i^{th} dimension of \mathbf{x} , and μ_c is the prototype of concept c . The complete procedure of prototype extraction and concept assignment is provided in Appendix. Interested readers can also refer to [3], [48].

B. From Individual Concepts to a Densely Connected Web

In [3], we show how concepts can be extracted and represented in a densely connected web based on Markov Random Fields (MRF) [55], providing greater robustness of reasoning than considering concepts in isolation. For the sake of completeness, we describe the method here briefly.

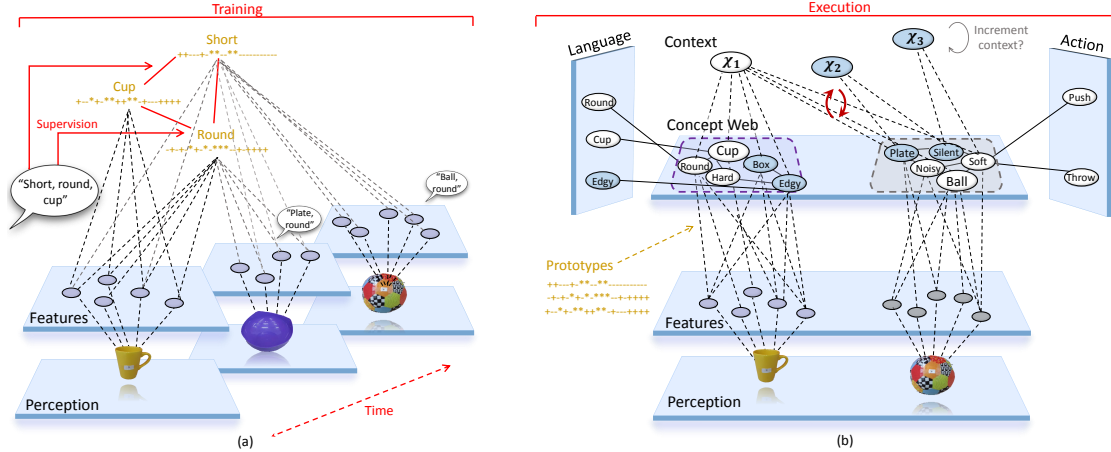


Fig. 6: The schematic presentation of the whole system. (a) The training phase is the only phase that includes supervision. Features are detected from instances automatically, while simultaneously the instances are labeled by a human partner. Prototypes are extracted from these two types of information. (b) The execution phase is fully autonomous. Information can flow in from the perception space, through a feature extraction mid-level, where they are compared against previously extracted prototypes, or from the language and action. Detected concepts effect each other in a concept web to converge to a common interpretation. (A number of nodes are randomly illustrated with white color to exemplify active concepts.) The concept webs of each object in the scene are then analyzed to elicit the contextual setting. The contextual information in turn is fed back to the concept webs to guide their activation. There is constant information flow between these two kinds of nodes.

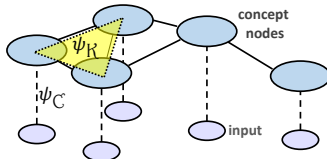


Fig. 7: Visualization of the MRF-based model. Initial predictions are used to initialize concept node probabilities. Conformance to these are maintained by minimizing the unary potential functions ψ_C . Clique potentials are initialized from the co-occurrence information of the training data, and conformance to this is maintained by minimizing the clique potentials ψ_K .

A Markov Random Field is an undirected graph of random variables, over which inference is often carried out by a minimization of a predefined energy function. In the energy function, the consistency of the categories (*i.e.*, the nodes) with the input (by the “data term” in MRF) and the consistency between the categories (by the “smoothness term”) are specified. By minimizing this energy function, an MRF finds the most likely categories for an input, satisfying also the regularization constraints specified in the smoothness term.

In our representation of the concept web as an MRF, the nodes correspond to concepts, and commonly co-occurring concepts are connected via edges. With $\mathbb{C} = \mathbb{N} \cup \mathbb{A} \cup \mathbb{V}$ being the set of all concepts, the concept web W is defined as a graph, $W = G(\mathbb{C}, \mathbb{E})$, with each

concept $c \in \mathbb{C}$ being a node in W , and edge $\epsilon_{ij} \in \mathbb{E}$ meaning that concepts c_i and c_j have co-occurred in the training set. In other words, the edges between the nodes (concepts) are learned from the training data, which is composed of observations of individual objects and behaviors executed on them.

What happens when a new observation arrives is depicted with a schematic representation in Figure 7. The connections from the input to the nodes correspond to the data term (represented with ψ_C , unary potentials), and the connections between the nodes model the smoothness term (represented with ψ_K , clique potentials). The energy $U(\omega)$, of a given MRF configuration ω , is then:

$$\begin{aligned}
 U(\omega) &= U_{data}(\omega) + U_{smooth}(\omega) \\
 &= \sum_{c \in \omega} \psi_c(c) + \sum_{\mathcal{K} \in \mathbb{K}} \psi_K(\mathcal{K}, \omega), \quad (2)
 \end{aligned}$$

where the first term, *i.e.*, the data term, is a summation of the unary potentials for each active concept c in ω , and the second term is the smoothness term, as a summation of clique potential functions. The unary potential function denoted by ψ_C is defined as:

$$\psi_C(c) := D(c, \mathbf{x}), \quad (3)$$

with \mathbf{x} being the instantaneous observation, $D(c, \mathbf{x})$ its distance to concept c (Equation 1). The potential function

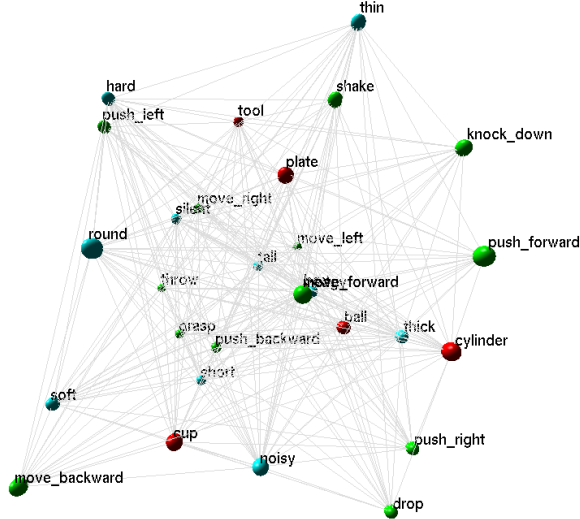


Fig. 8: A sample concept web constructed by the iCub. Noun, adjective, and verb concepts are indicated with red, blue, and green respectively. Connections between concepts are shown with gray. Ubigraph 3D visualization library [56] is used for displaying the graph, presented here as a projection on the 2D plane. [Taken from [3], best viewed in color]

for cliques, denoted by ψ_K , is defined as:

$$\psi_K(\mathcal{K}, \omega) := \mathcal{V}_{\mathcal{K}}(\mathbf{x}_{\mathcal{K}}) := \sum_{x_i \in \mathbf{x}_{\mathcal{K}}} |val(x_i) - E(x_i | \mathbf{x}_{\mathcal{K}-i})| \quad (4)$$

where $\mathcal{V}_{\mathcal{K}}(\mathbf{x}_{\mathcal{K}})$ is defined as an (abused) shorthand notation for the potential of a clique node consisting of active variables $\mathbf{x}_{\mathcal{K}}$, x_i is the i^{th} variable in the clique, $\mathbf{x}_{\mathcal{K}-i}$ are the variables in the clique excluding the i^{th} variable, $val(x_i)$ is the current value assignment of the variable x_i , $|\cdot|$ is the absolute value function, $E(\cdot)$ is the expected value function, and $E(x_i | \mathbf{x}_{\mathcal{K}-i})$ is the expected value of the i^{th} variable given the values of the remaining variables in the clique.

The energy function in Equation 2 must be minimized to find the most likely configuration $\omega^* = \arg \min_{\omega} (U(\omega))$. We use the Loopy Belief Propagation (LBP) algorithm [57], [58] designed specifically for cyclic MRFs, for this energy minimization. A schematic depiction of the complete system is presented in Figure 6(b), showing the information flow from perception space through a feature-extraction mid-layer, as well as from language, and action spaces. The finalized concept web has all the relevant concepts in the active state (indicated with white color), and connected to their relevant counterparts in the three spaces. A sample concept web that is constructed is shown in Figure 8.

IV. MODELING CONTEXT USING INCREMENTAL LATENT DIRICHLET ALLOCATION

In our framework, context is linked to the set of concepts that the robot perceives from its immediate environment. We use Latent Dirichlet Allocation (LDA) to detect the latent (unobserved) context(s) of the scenes. Initially, the scene is represented as a concept web (Section III), which is then used as an input to contextual analysis. The detected contexts are in turn fed back to the concept web to guide its reasoning. In this section, we provide the details of these steps.

A. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [39] is a method for modeling topics of documents in large text corpora. Assuming a document $d \in \mathbb{D}$ is a set of words $\{w_1, \dots, w_N\}$ drawn from a fixed vocabulary ($w_i \in \mathbb{W}$, vocabulary size is $|\mathbb{W}|$, $|\cdot|$ denotes set cardinality), LDA posits a finite mixture over a fixed set of topics $\{z_1, \dots, z_k\}$ ($z_t \in \mathbb{Z}$, $|\mathbb{Z}| = K$ is the topic count). Then, a document can be described by its probabilities of being related to each of these topics, $P(z_t | d_i)$. Meanwhile, a topic is modeled by its probability of producing each word in the vocabulary, $P(w_j | z_t)$. LDA tries to infer these document and topic probability distributions, given a corpus \mathbb{D} .

Being a generative model, LDA assumes that the corpus had previously been generated by choosing a Dirichlet prior α , and a $K \times |\mathbb{W}|$ matrix, called β , that contains the probabilities of each word given each topic, *i.e.*, with entries $\beta_{jk} = P(w_j | z_k)$. Furthermore, it assumes that every document $d \in \mathbb{D}$ had been generated by first choosing a probability distribution of topics for this document, $\theta \sim \text{Dir}(\alpha)$, followed by, for each word location n in the document, choosing a topic $z_n \sim \text{Discrete}(\theta)$, and eventually a word w_n , given the chosen topic z_n and the β matrix denoting $P(w_n | z_n, \beta)$.

LDA effectively tries to estimate the unknown α and β parameters from the given corpus, through which it is possible to infer any other parameter. This problem, however, is famously intractable [39]. There are various solutions though, including a variational inference method [39], a collapsed Gibbs sampling solution [59] and collapsed variational inference approach [60].

The strengths of LDA are two-fold: First, it is a generative model. There exists other powerful, non-generative models for topic analysis (for instance, see [61]), however, being a generative model, LDA can assign probabilities to documents that have not been seen before. Second, it allows non-strict memberships of words to topics: A word may be generated by multiple

Algorithm 1 Batch Gibbs sampling algorithm [59]. Algorithm formulation adapted from [62].

```

initialize  $\vec{z} = [z_1, \dots, z_N]$  randomly from the set  $\{1, 2, \dots, K\}$ 
while not converged do
  choose a word index  $j$  from  $\{1, 2, \dots, N\}$ 
  sample  $z_j$  according to  $P(z_j | \vec{z}_{\setminus j}, \vec{w}_N)$  (Equation 5)
end while

```

topics, and according to which document it occurs in, considering the topic probability distribution of the document, a different topic might be assigned to the different occurrences of the word.

Batch Gibbs Sampling Approach for Solving LDA Introduced by Griffiths and Steyvers [59], the Batch Gibbs Sampling Approach (Algorithm 1) is a ‘‘collapsed’’ method for solving the LDA problem, because it integrates out the Dirichlet parameters and instead directly samples the topic variables $\vec{z} = [z_1, \dots, z_N]$ for every word position $n \in \{1, \dots, N\}$. The algorithm starts by randomly assigning \vec{z} , and then until convergence samples the topic assignment z_j for the word w_j in document d , according to the instantaneous state:

$$P(z_j | \vec{z}_{\setminus j}, \vec{w}_N) \propto \frac{n_{z_j, \setminus j}^{w_j} + \xi}{n_{z_j, \setminus j} + |\mathbb{W}| \xi} \times \frac{n_{z_j, \setminus j}^d + \alpha}{N_{\setminus j}^d + K \alpha}, \quad (5)$$

where $(\cdot)_{\setminus j}$ notation stands for all items excluding the currently considered index j , therefore letting $\vec{z}_{\setminus j}$: the vector of all topics except z_j , \vec{w}_N : the vector of all words, $n_{z_j, \setminus j}^{w_j}$: the number of times that word w_j has been assigned to topic z_j except at index j , $n_{z_j, \setminus j}$: the number of times that any word has been assigned to topic z_j except at index j , $n_{z_j, \setminus j}^d$: the number of times that any word in document d has been assigned to topic z_j except at index j , $N_{\setminus j}^d$: the total number of all words in document d except at index j , $|\mathbb{W}|$: the size of the vocabulary set, and K : the topic count. The approach assumes symmetric Dirichlet priors α and ξ , *i.e.*, that they are vectors with the same value in all entries. The α vs. ξ trade-off controls the compromise between having few topics per document, vs. having few topics per word.

B. Modeling Contextual Information with LDA

We now describe how we model our robotics scenario within the Latent Dirichlet Allocation framework. The components of our system correspond to the specific LDA terms as follows:

1. Each scene the robot encounters is represented as an LDA document. In our concept web-based model, this scene/document is then a set of active concepts.
2. The sum of all the encountered scenes is analogous to the corpus \mathbb{D} .

TABLE V: The correspondence between the LDA terms and the notation used in this work

LDA	Our Notation
document $d \in \mathbb{D}$	a single scene (<i>i.e.</i> , the set of active concepts in the scene)
corpus \mathbb{D}	all encountered scenes
word $w_i \in \mathbb{W}$	an active concept c_{act} in the concept webs (can be a noun, adjective, or verb: $c_{act} \in \mathbb{C} = \mathbb{N} \cup \mathbb{A} \cup \mathbb{V}$)
topic	a ‘context’, either Kitchen, Playroom, or Workshop

3. Each active concept c_{act} in this scene corresponds to a word w_i in the document.
4. Finally, the ‘‘context’’s that we are trying to discover correspond to the latent topics of LDA.

Our aim is to associate each scene with the relevant contexts. Table V summarizes the correspondence between the LDA terms, and the notions in our robotics scenario. Also note that LDA works on the bag-of-words assumption that the order of the words in a document is not important, which is compatible with our scenario: Indeed, concepts exist or do not exist in a scene, there is no ordering between them. That is, the *probabilities* of the existence of concepts are not dependent on an *order* of appearance, in contrary to, for instance, certain natural language processing scenarios.

C. An Incremental and Online Version: Incremental-LDA

Since a robot operates in a dynamic world, it needs to be able to discover newly emerging contexts with new interactions. To truly comply with developmental principles, the robot not only needs to estimate itself the ideal number of contexts, but also to validate its own prediction continuously and revise and update it if necessary; we cannot foresee this for it (for a very good discussion on what makes a system developmental, see [63]).

One limitation of LDA is that it requires a fixed number of topics. This requirement is characteristic of the parametric approaches, where the parameters of the solution are defined a priori and do not change no matter how many training examples are encountered. Although they are very widely used and successful in general (among well-known examples are regression, Fisher’s discriminant analysis, Bayesian graphical methods), the necessity of predefining parameters can be restrictive. In the case of latent feature models, different methods have been proposed for dealing with an unknown number of clusters, focusing especially on the Dirichlet process and Bayesian solutions [64]–[66]. Targeting specifically the LDA problem, Teh *et al.* [67] proposed a Hierarchical Dirichlet Process framework which can start with infinitely many possible topics, and settle on the likeliest

Algorithm 2 The proposed Incremental-LDA algorithm

```
initialize context count  $K \leftarrow 1$ .
for all encountered scenes do
  run K-Incremental Gibbs sampler with  $K$ 
  while  $\mathbb{C}_{low} \neq \emptyset$  do
    increment context count  $K \leftarrow K + 1$ 
    run K-Incremental Gibbs sampler with  $K$ 
  end while
  output converged context assignments  $\bar{z}_N$  for the scene
end for
```

number of topics itself. Wang *et al.* [68] developed an online solution for this hierarchical setting.

Since the previously proposed variations are either batch or parametrically dependent on the number of topics K , we enhance the original LDA methodology with a simple mechanism that allows both online learning, and dynamic updating of the ideal K value over time. This new variant, henceforth called Incremental-LDA, does not need the number of contexts to have been predefined, starting instead with the most general case of $K = 1$, and increasing the context count when necessary.

Incremental LDA Incremental-LDA (Algorithm 2) decides on K dynamically, starting the with most general case, $K = 1$, and incrementing the context count as necessary. For deciding when to increase K , we define and use \mathbb{C}_{low} , the set of words whose confidence values for contextual assignments are lower than a threshold value τ . If there exists such words with low confidences, *i.e.*, $\mathbb{C}_{low} \neq \emptyset$, Incremental-LDA attempts to increase their confidences by incrementing the context count.

K-Incremental Gibbs Sampling Standard batch Gibbs sampler proposed by Griffiths and Steyvers [59] is not suitable for use with Incremental-LDA, because it needs to start from scratch each time the context count K is incremented. The previous solution is forgotten completely, whereas parts of it would still be applicable. This is especially true for the parts of the previous solution that exhibited high enough confidence. Therefore we introduce K-Incremental Gibbs Sampling (Algorithm 3) as an incremental variant: When the context count is incremented to K , K-Incremental Gibbs Sampling resumes its search from the previously converged solution for $K - 1$ contexts, conducting a local search in the close vicinity. This is done by retaining the previous assignments of the high-confidence terms, while initializing low-confidence terms (\mathbb{C}_{low}) to the newest context id K . Effectively, the highly confident part of the solution is reused. Note that for escaping possible local minima, a high-confidence term can also be reassigned to the new context with a low probability $\delta \ll 1$.

D. Making Use of Context: Feeding the Contextual Information back to the Concept Web

Since the system does not employ an attentional mechanism, it focuses on each object in the scene one by one, identifying the concepts related to each one with a concept web. The set of all these active concepts for all objects is then used for deducing the context of the scene. After determining the context, the probabilities of concepts are updated with the conditional likelihood of concepts in that context:

$$P(c)^* = \sigma \times P(c) + (1 - \sigma) \times P(c|\chi), \quad (6)$$

where $c \in \mathbb{C} = \mathbb{N} \cup \mathbb{A} \cup \mathbb{V}$ is a concept, $P(c)$ is the MRF-decided probability of the concept c , χ is the context, $P(c|\chi)$ is the probability of the concept given the context (decided by Incremental-LDA), and $P(c)^*$ is the updated value of the concept probability. The whole system, which consists of (1) reiteration of the object concept webs, (2) context deduction, and (3) probabilistic update of concept webs according to the context, is then repeated until the convergence of the individual concept webs and context analysis. See Figure 6(b) for a schematic visualization.

σ in Equation 6 is responsible with regulating the strength of contextual feedback in our world, with $\sigma = 0$ corresponding to using only contextual information, and $\sigma = 1$ corresponding to pure concept web decision. An average log likelihood \hat{l} is calculated over the test set as follows and depicted in Figure 9:

$$\hat{l} = \frac{1}{N|\mathbb{C}^{n+}|} \sum_{i=1}^N \sum_{c \in \mathbb{C}^{n+}} \log P(c|x_n, \sigma), \quad (7)$$

with N denoting the observation count, x_n being the n^{th} observation, \mathbb{C}^{n+} with cardinality $|\mathbb{C}^{n+}|$ being the set of concepts *related with* the n^{th} observation, and $P(c|x_n, \sigma)$ denoting the probability of obtaining the related concept c given observation x_n , under the setting σ . The results estimate a reasonable interval between $[0.4, 0.5]$; from this interval, we select σ as 0.5. Note that the convergence of \hat{l} for $\sigma \geq 0.7$ corresponds to the contextual feedback being too weak to affect concept

Algorithm 3 The K-Incremental Gibbs sampling approach we propose as a companion to Incremental-LDA

```
initialize  $\bar{z}_N$  from the previous solution for  $K - 1$  contexts
 $\forall$  context  $t \mid c_t \in \mathbb{C}_{low}$ , initialize  $z_t \leftarrow K$ 
 $\forall$  context  $t' \mid c_{t'} \notin \mathbb{C}_{low}$ , reassign  $z_{t'} \leftarrow K$  with prob.  $\delta \ll 1$ 
while not converged do
  choose a concept index  $j$  from  $\{1, 2, \dots, N\}$ 
  sample  $z_j$  according to  $P(z_j|\bar{z}_{N \setminus j}, \bar{w}_N)$  (Equation 5)
end while
```

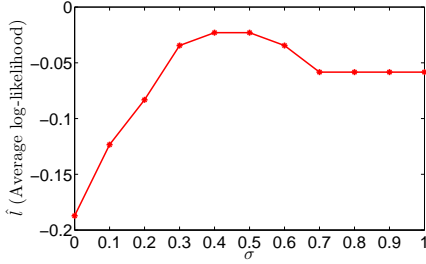


Fig. 9: Average log likelihood \hat{l} for varying σ (Equation 6, $\sigma = 0$: Pure contextual information, $\sigma = 1$: Pure concept web decision). The interval $[0.4, 0.5]$ is depicted as maximizing \hat{l} .

web decision at all, therefore the average log-likelihood does not vary in this region.

E. Entropy-Based Evaluation of the System

We define an entropy-based metric of disorder to evaluate the performance of the system, with two terms:

$$\tilde{H} = \rho \times H(C|X) + (1 - \rho) \times H(X|S), \quad (8)$$

where $H(\cdot)$ is the entropy function, C , X , S are random variables denoting concepts, contexts, and scenes respectively, $H(C|X)$ is the conditional entropy of concepts given the context, $H(X|S)$ is the conditional entropy of contexts given the scene, and ρ is a parameter determining the relative importance of the two terms (set to 0.25 experimentally). These two terms stem from two possibly opposing targets: We would like as few contexts as possible assigned to a scene, giving us more specific “documents”; and at the same time as few concepts as possible associated with a context, thereby more specific “topics”. A combination of the two terms is expected to give us the most specific contextualization of the scene².

V. EXPERIMENTS AND RESULTS

We now evaluate our framework and assumptions from three different aspects:

1. We first test whether Incremental-LDA can determine the optimal number of contexts; *e.g.*, if it stops adding new contexts at the optimal point. We also test if reusing partial solutions in K-Incremental Gibbs sampler leads to better performance.
2. Then we compare extracting context *directly* from raw features of the scene, against modeling it on top of the concept web.
3. Finally, we demonstrate how contextual information can improve reasoning, in three different scenarios:

²Similar multi-objective optimization of these two metrics can be found in the literature, for instance see [59].

- (1) scene interpretation, (2) object recognition, and (3) planning.

The training and test scenes in the experiments can belong to 3 different contexts (*Kitchen*, *Playroom*, and *Workshop*). Unless explicitly mentioned, a scene is a *pure* context scene, *i.e.*, contains elements of a single context. A scene can also contain elements from multiple contexts, in which case it is denoted as a *mixed* context scene. For generating each scene in the set, a context is decided randomly and then the scene is populated with randomly chosen objects that have the noun, adjective, and verb attributes related to the selected context.

A. Performance of Incremental-LDA and K-Incremental Gibbs Sampling

First, we analyze the dynamics of Incremental-LDA under two variables: One is a varying number of encountered scenes, in which we hope to detect the correct number of contexts as soon as possible, and the second is the varying number of contexts K , in which we look for a preference for the expected number of contexts, *i.e.*, $K = 3$ for our case. Figure 11 depicts the number of highly uncertain concepts ($|\mathcal{C}_{low}|$) and the entropies (\tilde{H} , Equation 8) of the system for different configurations. Note that, left alone, Incremental-LDA would itself converge to a certain K setting, which is ideally $K = 3$ here, however, for the sake of comparison, we force varying K values in these experiments.

For each configuration, we use 10 test sets of $|\mathbb{D}|$ scenes with random contexts. The number of encountered scenes in a test set, $|\mathbb{D}|$, is one of the free variables. Each scene $d \in \mathbb{D}$ is populated with 3-5 random objects of the randomly selected context. Figures 11(a) and 11(b) show that with a reasonable number of scenes ($|\mathbb{D}| \geq 3$), $|\mathcal{C}_{low}|$ remains positive until K reaches 3, and then diminishes. For cross-check of the results, Figures 11(c)

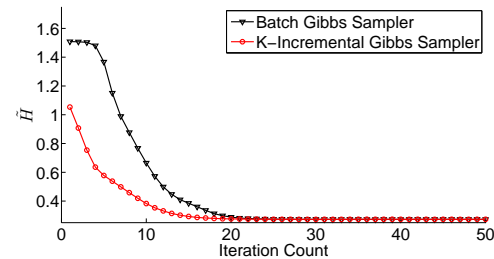


Fig. 10: A comparison of the entropy (\tilde{H}) evolution (Equation 8) of K-Incremental Gibbs solver, versus the standard batch Gibbs solver. The K-Incremental Gibbs solver is fed a partial solution for 2 contexts and then run for $K = 3$ contexts. The batch Gibbs sampler is directly run for $K = 3$ contexts.

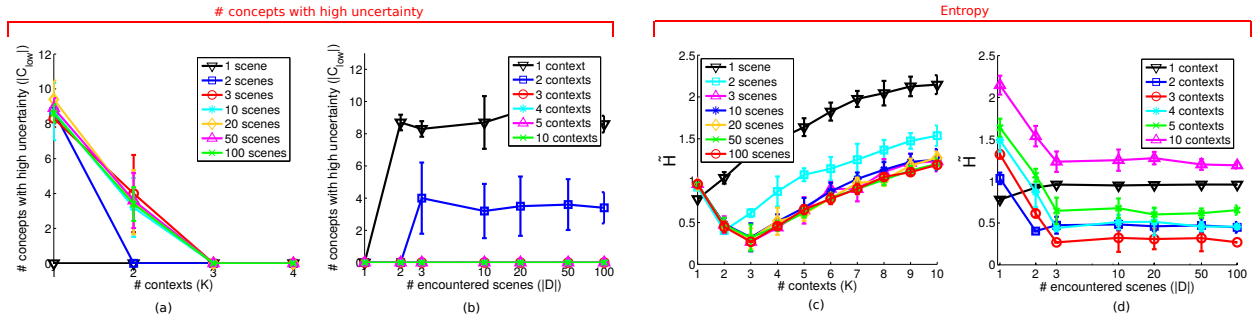


Fig. 11: The effect of encountered scene counts and varying context counts K . Note that Incremental-LDA would itself stop at $K = 3$, however we force increasing K for the sake of comparison. (a) Effect of increasing K on the number of uncertain concepts, $|C_{low}|$, for varying number of scenes. By $K = 3$ contexts $|C_{low}|$ diminishes to 0, therefore Incremental-LDA would stop adding new contexts at this point. (b) Effect of encountered scenes on the number of uncertain concepts, $|C_{low}|$, for different context counts. (c) Effect of increasing K on the entropy of the system, \tilde{H} , for varying number of scenes. (d) Effect of encountered scenes on the entropy of the system, \tilde{H} , for different context counts. In all the experiments, 10 test sets of $|\mathbb{D}|$ scenes each are used. The mean values for the 10 test sets are plotted, while the standard deviations are indicated with error bars. In (b) and (d), the x-axis is in log-scale. [Best viewed in color]

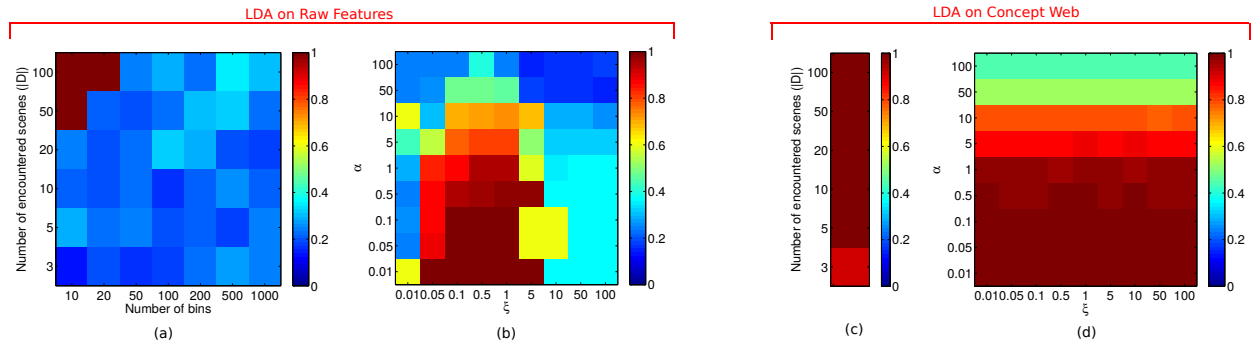


Fig. 12: The performances of LDA over raw features only, versus of LDA over MRF-based concept web, presented as prediction accuracies scaled to $[0,1]$. Presented values are the predicted likelihoods of “correct contexts” in each corresponding case. For evaluation, the ground truth data of the expected contexts were extracted via supervision. α and ξ are the trade-off parameters from Equation 5. (a) Using only the raw features as input to LDA, for varying discretization bin counts and increasing numbers of encountered scenes ($\alpha = 0.1$, $\xi = 0.1$). (b) Using only the raw features as input to LDA, for varying settings of α and ξ (50 scenes, 10 bins). (c) Using the concept web as input to LDA, for increasing numbers of encountered scenes. ($\alpha = 0.1$, $\xi = 0.1$. Discretization is not necessary, therefore the result vector is 1-dimensional.) (d) Using the concept web as input to LDA, for varying settings of α and ξ (50 scenes). [Best viewed in color]

and 11(d) presents the change of the entropy of the system, \tilde{H} , for varying K and scene count. The mean and standard deviation values for the 10 test sets are indicated with error bars. We hope to achieve as early convergence as possible to the correct context count, which is duly achieved by the $|\mathbb{D}| = 3$ scenes mark. Note that for reasonable numbers of scenes, the lowest possible entropy values are achieved when $K = 3$, which conforms our expectations since, in our experiments, since we truly have three contexts, namely the *Kitchen*, *Playroom*, and *Workshop* contexts.



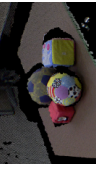



In all four cases, the system converges with about 3 encountered scenes, and shows preference (in terms

of minimal entropy and minimal number of highly uncertain concepts) at $K = 3$ contexts. Since this is also the point at which $|C_{low}|$ reaches to zero, Incremental-LDA then stops adding new counts, correctly deducing the minimum entropy setting of our system.

Next, we compare the performance of K-Incremental Gibbs sampling with batch Gibbs sampling. The question is whether reusing the previous partial solution leads to faster convergence times for K-Incremental Gibbs sampler. Our test set includes 100 scenes.

Figure 10 presents the results over this test set that conform with our expectations: Using a partial solution for 2 contexts, K-Incremental Gibbs sampler converges

TABLE VI. Prediction of context for a few example scenes where confidences are indicated in parentheses. Bold text indicates correct decisions. [Best viewed in color]

Pure Contexts			Mixed Contexts		
Scene	Existing Objects	Predicted Context (% contribution)	Scene	Existing Objects	Predicted Context (% contribution)
	2 cups, 2 plates (100% Kitchen)	Kitchen (100%)		3 boxes, 1 ball, 1 cylinder, 1 tool (66% Playroom, 33% Workshop)	Playroom (72.59%) Workshop (26.23%) Kitchen (1.18%)
	2 boxes, 2 balls (100% Playroom)	Playroom (100%)		2 plates, 2 cup, 1 ball, 1 box (66% Kitchen, 33% Playroom)	Kitchen (62.04%) Playroom (37.14%) Workshop (0.82%)
	2 tools, 3 cylinders (100% Workshop)	Workshop (100%)		1 tool, 1 cylinder, 1 plate, 1 cup (50% Kitchen, 50% Workshop)	Kitchen (46.67%) Workshop (51.56%) Playroom (1.77%)

faster compared to the batch solver. We measure the convergence of the system in terms of its entropy³.

Note that K-Incremental Gibbs Sampling is fundamentally a variant of the standard Gibbs sampler employing an informed initialization, which has been successful in various challenging problems with very high number of contexts (topics), e.g., in [59], which extracts ≈ 300 “hot” scientific topics over 28,154 abstracts published in PNAS between 1991 and 2001. Therefore, in spite of the physical limitations on the data set used in this study, resulting in a modest number of concepts and contexts, it is reasonable to expect that K-Incremental

³Also note that the entropy value eventually reached by the two solvers is indeed the expected minimum entropy value for these environmental conditions.

Gibbs sampler will also be able to scale up for a high number of contexts as well.

B. Context from the Concept Web against Context from Raw Features

Next we evaluate how useful the concept web is in guiding contextualization. Figure 12 shows the comparison of LDA on concept web versus LDA on raw-features-only. First, we contrast how the two schemes fare in case of insufficient scene encounters. Concurrently, we also investigate to what degree the discretization of the raw-features is necessary, if at all. In the second type of tests, we conduct a grid parameter search in the LDA space, to decide the best parameter settings for the two algorithms, as well as their sensitivity level to the changes in these parameters. Note that these two sets of experiments must be thought of in unison, in the sense that we have iteratively updated the parameters used in one set according to the best results of the other set, therefore we hope to present meaningful results in both sets. In the figures, we present the predicted likelihoods assigned by these algorithms to the contexts that we “know” to be true. The correct contexts have been decided through supervision for evaluation purposes only.

Figures 12(a-b) depict the contextualization performance on raw features directly, with Figures 12(c-d) showing the performance of the concept web. Figure 12(a) versus Figure 12(c) compare the results of the first set, *i.e.*, the effects of scene count and discretization (with the trade-off parameters α and ξ from Equation 5 both set to 0.1) An important result that pops out is that the raw features approach needs 50 scenes to settle on a meaningful partitioning, while the concept web method manages to converge with an impressive speed at as few as 3-5 scenes. Even at 50 scenes, the raw features approach needs to be supported by coarse discretization of the features (*i.e.*, being divided into 10 bins at most), since LDA is unable to locate statistically significant co-occurrences otherwise. For other settings, the decisions of the raw-features approach are at chance level: 33.3% for a 3-way decision.

Figures 12(b) and 12(d), on the other hand, present the results of the grid search in the α - ξ space (with 50 scenes, 10-bins of discretization). Once again, we see that LDA-on-raw-features is more fragile against parameter changes, while the concept web method proves robust under most settings. Indeed, even for the worst parameter settings, notice that the concept-web case provides confidences of over 50%, which are sufficient for correct decision making, and are well over the chance level of 33.3%.

TABLE VII: Object recognition in context. Prediction confidences are indicated in parentheses. Bold text indicates correct decisions whereas stroked text indicates wrong decisions. [Best viewed in color]

Objects	Perception only		Concept Web		Context	In Context	
	Predicted Nouns (% confidence)	Predicted Adjectives (% confidence)	Predicted Nouns (% confidence)	Predicted Adjectives (% confidence)		Predicted Nouns (% confidence)	Predicted Adjectives (% confidence)
	ball (8%) box (13%) cup (43%) cylinder (20%) plate (9%) tool (7%)	edgy (34%) hard (71%) noisy (42%) short (54%) thick (47%)	ball (0%) box (0%) cup (100%) cylinder (0%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (0%) short (100%) thick (100%)		ball (0%) box (0%) cup (100%) cylinder (0%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (0%) short (100%) thick (100%)
	ball (12%) box (13%) cup (17%) cylinder (29%) plate (12%) tool (17%)	edgy (45%) hard (56%) noisy (58%) short (41%) thick (40%)	ball (0%) box (0%) cup (0%) cylinder (100%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (100%) short (0%) thick (0%)		ball (0%) box (0%) cup (0%) cylinder (100%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (100%) short (0%) thick (100%)
	ball (14%) box (17%) cup (19%) cylinder (26%) plate (13%) tool (11%)	edgy (42%) hard (60%) noisy (42%) short (45%) thick (40%)	ball (0%) box (0%) cup (100%) cylinder (0%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (0%) short (100%) thick (0%)		ball (0%) box (0%) cup (0%) cylinder (100%) plate (0%) tool (0%)	edgy (0%) hard (100%) noisy (0%) short (100%) thick (0%)

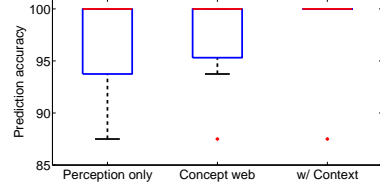


Fig. 13: The combined results of object recognition in context, over all 15 objects in the test set. The prediction accuracies over all determined noun and adjective concepts, using (i) only perceptual features, (ii) the concept web, and (iii) contextual information are compared. In the plot, the red lines denote the median values, the boxes denote the data that fall between the 25th and 75th percentiles, the whiskers cover the extreme data that are not outliers, and stars indicate the outliers.

The results confirm that learning context from concepts is better than learning them from raw features in two aspects: (i) Learning converges faster, and is therefore more reliable even after as few as 3-5 scene encounters, and (ii) It is less sensitive to the model parameters, which increases the robustness of learning without needing a careful tuning of parameters.

C. Using Context, Part 1: Making Sense of Pure- and Mixed-Context Environments

Now we demonstrate how our context model can be utilized in reasoning and decision making. The first scenario is designed for assessing how successful our model is in recognizing contexts of scenes. The robot encounters six different scenes, three of which are composed of items of a single context, and the remaining three of multiple contexts. Table VI demonstrates the predicted context(s), showing that the robot can distinguish between pure and mixed-context scenes correctly, and decide on the correct components in case of a mixed-context scene. These results are important because they demonstrate that our interpretation of the scene context is correct, regardless of the scene being composed of a single context or multiple contexts. Therefore, we obtain justification for our next step of using this contextual interpretation for guiding reasoning in other cognitive tasks.

D. Using Context, Part 2: Object Recognition in Context

The second scenario considers the effect of context on object recognition. Table VII demonstrates the recognition results for seven sample objects that are either (i) individually perceived (columns 2-3), (ii) assessed in an

individual concept web (columns 4-5), or (iii) evaluated in context⁴ (columns 7-8).

The results show that concept web itself can correct certain mistakes of the perception-only assessment, while also boosting confidences of guesses to 100% certainty. However, it is not flawless and is also prone, albeit in a lesser amount, to errors (see rows 2-3 in the table). In such cases, it is especially difficult to correct these errors, due to the (unfounded) high confidence associated with them. Contextual information can be beneficial in these settings.

Remembering our fundamental assumption that related objects occur together in context (which allowed us to develop an LDA-based model in the first place), the system can use context to revise and correct its previous judgments. The loop of (a) context deduction, (b) probabilistic update of concept web, and (c) reiteration of MRF, as described in Section IV-D and Equation 6, also visualized in Figure 6, is utilized for refining predictions in context. Combined results for all 15 test objects are demonstrated in Figure 13, which also show an improvement of performance for the context-guided recognition.

In all these results, however, the individual predictions made solely using prototypes are quite good already, thereby making it difficult to adequately estimate the benefits of using context. Hence we have conducted an additional set of experiments, depicted in Figure 14, under artificial noise specifically added to the prototype

⁴The objects are given in pure-context environments, for the sake of easy analysis.

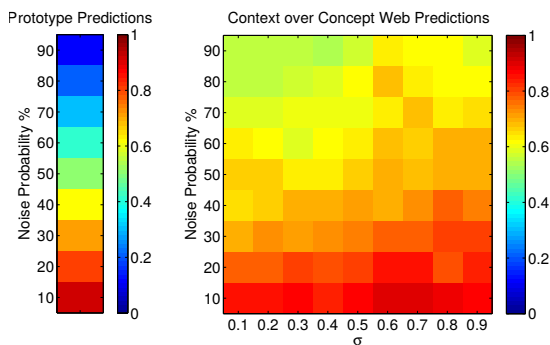


Fig. 14: The performance of the individual prototype-based predictions, versus context enhanced concept web predictions, under artificially added noise, presented as prediction accuracies scaled to $[0,1]$. The noise probability denotes the probability of artificial noise being added to each single concept, via reversing its prototype-predicted probability from $p\%$ to *reversed* to $(100 - p)\%$. σ refers to the trade-off parameter in Equation 6.

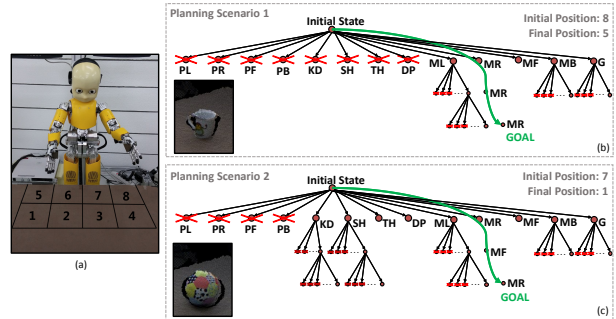


Fig. 15: Pruning of forward planning trees by integrating contextual information. (a) iCub’s workspace. (b) First planning scenario. iCub is expected to move a cup from position 8 to position 5. Since pushing and knocking actions are dangerous in the kitchen context, these nodes are pruned without further expansion. Pruned branches are indicated with crosses. (c) Second scenario. iCub must bring a ball from position 7 to 1. Pushes are pruned, since pushing a ball causes it to roll down from the table. PX: Push left/right/forward/backward, MX: Move left/right/forward/backward, KD: Knock down, SH: Shake, TH: Throw, DP: Drop, G: Grasp.

predictions. An average of the prediction accuracies (scaled to $[0, 1]$) over 15 random trials over the test set are shown. A noise probability parameter is determined in the range $[10\%, 90\%]$, and this parameter defines the probability of selection of each concept for addition of artificial noise. In case a concept is selected, its prototype-predicted probability $p\%$ is *reversed* to $(100 - p)\%$. The σ trade-off parameter of Equation 6 is varied in the range $[0.1, 0.9]$. Each noise probability vs. σ combination is repeated 15 times⁵ and the average performance results are presented in Figure 14. The system is shown to be quite resilient under increasing artificial noise: Combining information from many sources all of which contributes to the contextual analysis, the system is able to detect the context correctly and thereby correct individual wrong predictions using the majority vote.

E. Using Context, Part 3: Planning in Context

Finally, we show how contextual information can be useful in a planning task. It is known that humans hugely rely on contextual information for planning their actions [69]–[73], possibly due to a severely restricted working memory capacity [74], [75], which results in efficient day-to-day planning, but maybe less-than-favorable performances in chess. The robots would also benefit from similar contextual guidance in planning.

⁵Each trial is a random one, due to the probabilistic selection of the reversed concepts, with probability equal to the momentarily utilized noise probability parameter.

To show how context can be used similarly in a robotic planning scenario, we provide two simple situations as proof-of-concept: The robot has to move two objects over a table (Figure 15(a)) from an initial to a goal position. Since the robot has learned the effect features of behaviors on training objects, it is theoretically able to expand a planning tree starting from the initial state and expanding behavior nodes until the goal condition is reached. These scenarios are simulated; however, the decisions of the robot are based on real world data: The robot plans according to the expected results of actions as learned by the verb prototypes. Although it does not physically *move* to perform the plan, theoretically the plans are executable. In other words, we are interested not in the physical success of the plans, but in the computational efficiency of producing these plans.

In the first scenario, Figure 15(b), the robot is asked to move a cup from position 8 to position 5. This goal can be achieved with three consecutive *move right* actions in our setting. A fully-expanded tree, therefore, would consist of three levels, and with a branching factor of 13, it will consist of $13^0 + 13^1 + 13^2 + 13^3 = 2380$ nodes. However, given the contextual information of the scene, which is the *Kitchen* context, the robot can refrain from expanding the inappropriate behaviors in a Kitchen⁶, leaving only the *move left*, *move right*, *move forward*, *move backward* and *grasp* as possible actions to be expanded. Such an elimination gives a drastic reduction in the size of the planning tree, resulting in $5^0 + 5^1 + 5^2 + 5^3 = 156$ nodes instead of 2380.

Figure 15(c) shows another scenario in the *Playroom* context. This time, the robot refrains from applying the *push* actions on associated objects, since balls, which are also in this context, tend to roll down and fall from the table when pushed. Therefore, the *push* nodes are pruned, leaving $9^0 + 9^1 + 9^2 + 9^3 = 820$ nodes in the tree. We use a breadth-first forward planning scheme subject to context-dependent pruning.

Figure 16 compares un-pruned and pruned node counts for 10000 random scenarios in the move-over-the-table scenario presented above, presented for the three contexts separately. Each scenario is prepared by randomly determining a context, as well as initial and goal positions on the table environment, and then asking the robot to plan a behavior sequence from the initial to the goal position in this contextual background. Note that the amount of node reduction in these experiments depend on the randomly chosen target position. If the goal position is very close to the initial position, then relatively little reduction is possible, since the height

⁶Assuming we do not want to, for instance, *shake* a full cup.

of the planning tree will already be fairly shallow even in the unpruned case. However, if the random target is chosen sufficiently far from the initial position, which would normally require a very deep and wide planning tree, significant pruning is possible. The outliers in the graph correspond to such points. Note that the amount of pruning in the Kitchen case is greater than the Playroom case, since potentially greater number of actions are non-applicable in the Kitchen case. In the Workshop case, where all actions are applicable, there are no possible reductions.

The reductions shown here are only provided as proof-of-concepts, but it is clear how important it is for a robot to learn to prune its search trees in a real world setting. For a very limited robot of a small, or maybe even intermediate set of actions, considering each action for every situation might be an option, but for any robot who aims to operate in the real world, the actions will be so varied and planning chains will necessarily be so long that even most basic reductions (*i.e.*, no need to consider opening the kitchen door for heating a glass of milk) will be of critical importance.

F. Running Time Performance of the System

The whole system is able to work close to real-time: 10 test runs with non-optimized code on a standard desktop PC (i5 core, 8GB RAM) provided an average running time of $209.82ms \pm 3.38ms$ for the detection of context with Incremental LDA, and $1395.22ms \pm 15.31ms$ for the convergence of the concept web.

VI. SUMMARY AND DISCUSSION

In the article, we studied how a humanoid robot can model, learn and use context. For modeling context,

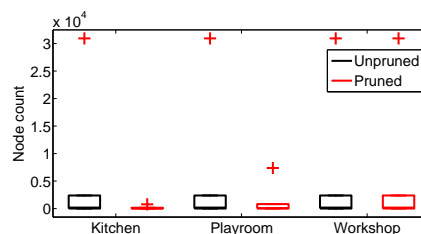


Fig. 16: The node counts of *unpruned* vs. *pruned* planning trees of 10000 random scenarios, grouped by their contexts. The Kitchen context is subject to more pruning, as expected, due to a large number of *NA* behaviors. The Workshop context, on the other hand, is not subject to any pruning, since all behaviors are potentially applicable. In the plot, the boxes denote the data that fall between the 25th and 75th percentiles, and stars indicate the outliers. [Best viewed in color]

we employed and extended Latent Dirichlet Allocation (LDA), a widely-used topic model in the computational linguistics literature. Unlike the existing applications of LDA in robotics for, *e.g.*, word learning, where LDA is directly applied onto low-level sensorimotor data, we were motivated by the concept web hypotheses in humans and its computational advantages to apply LDA onto a concept web model that we developed in our previous work using Markov Random Fields.

We demonstrated the following important aspects:

- In an unsupervised fashion, the robot can learn context even if the number of contexts is not given. By using an online version of the Gibbs sampler proposed in the article, the robot can work online to process new observations and can tackle new contexts. By a systematic analysis, we show that the model finds the *correct* number of contexts in different settings.
- The robot can use the learned contexts to improve its performance in cognitive tasks. In the article, we showed this aspect for object recognition and planning.
- Finally we show how learning context over a web of abstracted concepts is *easier* and provides better performance for an LDA-based architecture, which deals with the sensorimotor complexity of real world better than raw features themselves.

Below, we discuss several aspects of our design.

A. Basing Context on the Concept Web

Basing context on a concept web has significant computational advantages: In [3], we show how concept web enables a superior performance of object recognition and conceptualization as compared to a raw-feature based scheme. In this work, we provide further evidence regarding the performance of concept web for LDA-based conceptualization. We demonstrate how the concept web provides better performance with significantly fewer training examples, as well as reduced sensitivity against system parameters. These advantages are due to its abstraction capability: The real world presents an overwhelming amount of complex information, which needs some structure to be imposed before statistically significant relations can be discovered. This is argued to be the driving reason of conceptualization in humans as well (*e.g.*, [76]–[82], for a slightly different but interesting argument, see also [83].)

B. Planning in the Real World

Bylander [84] and Chapman [85] show that planning is intractable in the general sense, unless it is restricted

severely, for instance, to propositional planning with strictly positive preconditions and exactly one postcondition. Such restricted cases can be defined to reduce the planning problem to a polynomial-time subset; however, small deviations make the problem intractable again: *e.g.*, the NP-hard problem of allowing two postconditions along with one precondition, or the NP-complete problem of one strictly positive postcondition along with one precondition. As Bylander [84] and Hendler [86] note, it is difficult to describe any interesting world in propositional logic, let alone such restrictions for the sake of tractability. We have to find a workaround. We propose that this workaround can be, and for humans is, context [69]–[72].

Also supporting our hypothesis is the work of Siegler, *e.g.*, [87], who, from a developmental point of view, stresses how important context is in helping children choose which skill or problem solving strategy to apply in a certain situation. So important is this process of choosing, he claims, that the question is not “whether children ‘have’ a concept or strategy or theory at a given age”, but it is rather “the set of conceptualizations and strategies and theories that children know and the mechanisms by which they choose among them” [87].

C. Limitations and Future Work

Overall, we provide promising results that a learning scheme which *includes* background information, instead of leaving it out, is feasible *and* useful for a robot when dealing with the real world. Our work can be extended in several directions.

The experiments were performed on real objects, although the settings are not realistic. This limitation was due to the interaction capabilities of iCub: iCub cannot walk and is confined to a table-top environment. Moreover, due to its delicate hands and the limited precision of the touch sensors on the hands, the range of objects that can be interacted with was limited to light-weight and convex objects. This also restricted us in the varieties of contexts. However, LDA is shown to scale up extremely well in natural language processing settings, where it could be tested with huge corpora (*e.g.*, [39], [59]) as well in a number of other complicated real-life scenarios including functional miRNA–mRNA regulatory modules identification [41] and fraud detection [40]; therefore, we believe that our framework will scale well in realistic robotics settings.

In Incremental-LDA, we assumed that the number of contexts can only *increase* in the environment, and therefore it is not necessary to check if the context count K can go down. We observe similar assumptions in the

literature, *e.g.*, [67], where the number of topics can only *increase* in time. We believe that there is no reason for a biological cognitive agent to remove learned contexts from its system; although they might be merged as new contexts or split into sub-contexts, the only case where the number of contexts might decrease is when the agent forgets learned associations.

It should also be noted that, although our current concept web is composed of noun, adjective, and verb concepts, a cognitive model should include spatial, temporal, adverb, and social concepts as well. With the incorporation of these types of concepts in our concept web, contexts related to their semantics will also be able to manifest themselves in our model.

Another plausible extension is regarding the concept web: The current concept web is a model of long-term memory only, with links holding information about the robot’s experiences about the world. This long-term memory is activated based on the current perception, yet, there is no clear separation between short-term and long-term memory akin to humans.

ACKNOWLEDGMENTS

We would like to thank Angelo Cangelosi, Anna Borghi and Honghai Liu for fruitful discussions on integrating context into cognitive systems. For the experiments, we acknowledge the use of the facilities provided by the the Modeling and Simulation Center of METU (MODSIMMER). This work is funded by the Scientific and Technological Research Council of Turkey (TÜBİTAK) through project no 111E287, and partially supported by the Marie Curie International Outgoing Fellowship titled “Towards Better Robot Manipulation: Improvement through Interaction” (FP7-PEOPLE-2013-IOF- 628854) awarded to Erol Şahin.

APPENDIX

We use prototypes to represent the noun (N), adjective (A) and verb (V) concepts, the extraction of which is illustrated in Figure 6(a). The noun and adjective concepts are related to the object entities, while verb concepts are related to the changes induced on the objects by the behaviors. Therefore, the prototypes of the noun and adjective concepts are obtained from the entity feature vectors \mathbf{e} , while the verb concept prototypes are obtained from effect feature vectors \mathbf{f} . Each object in the training set is labeled beforehand by supervision to denote the concepts it is associated with: Each training object is strictly labeled with 1 noun concept (out of 6) and 5 adjective concepts (one from each of the 5 dichotomic pairs). In addition, every applicable behavior

is applied to each training object, and the interactions are labeled with strictly 1 verb concept.

During training, the entity and effect feature vectors are collected from the training objects, and divided according to the labeled concepts. For each concept, every feature is assessed in terms of its contribution to the concept: If the feature has a highly positive contribution to the concept, it is indicated with a ‘+’ in the concept prototype. ‘-’ denotes a negative contribution, and ‘*’ denotes inconsistent contribution. These contributions are decided by clustering the features, using Robust Growing Neural Gas (RGNG) clustering algorithm [88], in a two dimensional space of means and variances: The mean axis denotes the amount of the contribution, while the variance axis denotes the consistency. Features with positive mean and low variance are marked with ‘+’; negative mean and low variance with ‘-’; and high variance with ‘*’. Of special interest are the features marked with ‘*’s, which effectively distinguishes *irrelevant* features, that can be disregarded from comparisons regarding the concept.

Prototypes for the verb concepts are extracted in a similar manner, except that (1) they are calculated over the effect features \mathbf{f} , and (2) they include a ‘0’ character for features that are unaffected by the behavior.

Eventually, we obtain 29 prototypes in total; 6 for nouns, 10 for adjectives, and 13 for verbs. The prototypes of the noun and adjective concepts are of length 91, the same with the length of an entity feature vector \mathbf{e} , containing 66 visual, 13 audio, 6 haptic and 6 proprioceptive features. The prototypes of the verb concepts are composed of 66 characters, and denote visual features only. The prototypes used in this study are shown in Table VIII.

When a new object is encountered, its entity feature vector \mathbf{e} is compared against the noun and adjective prototypes. Similarly, if a behavior has been applied, the effect feature vector \mathbf{f} is compared against the verb concept prototypes to recognize the behavior. This comparison consists of finding the concepts that minimize the Euclidean distance between the object’s feature vector and the concept mean vector (Equation 1). The *irrelevant features* of each concept, marked with ‘*’ in the concept prototype, are excluded from this calculation.

REFERENCES

- [1] H. Celikkanat, G. Orhan, N. Pugeault, F. Guerin, E. Sahin, and S. Kalkan, “Learning and using context on a humanoid robot using Latent Dirichlet Allocation,” in *IEEE ICDL-EpiRob*, 2014, pp. 201–207.
- [2] W. Yeh and L. W. Barsalou, “The situated nature of concepts,” *The American journal of psychology*, pp. 349–384, 2006.

- [42] T. Nakamura, T. Nagai, and N. Iwahashi, "Grounding of word meanings in multimodal concepts using LDA," in *IEEE/RSJ IROS*, 2009, pp. 3943–3948.
- [43] T. Nakamura, T. Araki, T. Nagai, and N. Iwahashi, "Grounding of word meanings in Latent Dirichlet Allocation-based multimodal concepts," *Advanced Robotics*, vol. 25, pp. 2189–2206, 2011.
- [44] T. Araki, T. Nakamura, T. Nagai, S. Nagasaka, T. Taniguchi, and N. Iwahashi, "Online learning of concepts and words using multimodal LDA and Hierarchical Pitman-Yor Language Model," in *IEEE/RSJ IROS*, 2012, pp. 1623–1630.
- [45] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Automatic drive annotation via multimodal latent topic model," in *IROS*, Nov 2013, pp. 2744–2749.
- [46] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.
- [47] J. J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales," *Image and vision computing*, vol. 10, no. 8, pp. 557–564, 1992.
- [48] S. Kalkan, N. Dag, O. Yürüten, A. M. Borghi, and E. Sahin, "Verb concepts from affordances," *Interaction Studies*, vol. 15, no. 1, pp. 1–37, 2014.
- [49] E. H. Rosch, "Natural categories," *Cognitive psychology*, vol. 4, no. 3, pp. 328–350, 1973.
- [50] P. Gärdenfors, *Conceptual spaces: The geometry of thought*. MIT press, 2004.
- [51] G. Orhan, S. Olgunsoylu, E. Sahin, and S. Kalkan, "Co-learning nouns and adjectives," in *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob 2013)*, 2013, pp. 1–6.
- [52] L. Gabora, E. Rosch, and D. Aerts, "Toward an ecological theory of concepts," *Ecological Psychology*, vol. 20, no. 1, pp. 84–116, 2008.
- [53] J. K. Kruschke and M. K. Johansen, "A model of probabilistic category learning," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 25, no. 5, p. 1083, 1999.
- [54] Y. Rossee, "Mixture models of categorization," *Journal of Mathematical Psychology*, vol. 46, no. 2, pp. 178–210, 2002.
- [55] R. Kindermann, J. L. Snell *et al.*, *Markov random fields and their applications*. American Mathematical Society Providence, RI, 1980, vol. 1.
- [56] T. Veldhuizen, "Ubigraph: Free dynamic graph visualization software," 2007.
- [57] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields," in *Computer Vision—ECCV 2006*. Springer, 2006, pp. 16–29.
- [58] T. Heskes *et al.*, "Stable fixed points of loopy belief propagation are minima of the bethe free energy," *Advances in neural information processing systems*, vol. 15, pp. 359–366, 2003.
- [59] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. of the Nat. Acad. of Sci.*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.
- [60] Y. W. Teh, D. Newman, and M. Welling, "A collapsed variational Bayesian inference algorithm for Latent Dirichlet Allocation," in *Advances in neural information processing systems*, 2006, pp. 1353–1360.
- [61] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999, pp. 50–57.
- [62] K. R. Canani, L. Shi, and T. L. Griffiths, "Online inference of topics with Latent Dirichlet Allocation," in *International conference on artificial intelligence and statistics*, 2009, pp. 65–72.
- [63] A. Stoytchev, "Some basic principles of developmental robotics," *Autonomous Mental Development, IEEE Transactions on*, vol. 1, no. 2, pp. 122–130, 2009.
- [64] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The annals of statistics*, pp. 1152–1174, 1974.
- [65] T. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.
- [66] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the american statistical association*, vol. 90, no. 430, pp. 577–588, 1995.
- [67] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, vol. 101, no. 476, 2006.
- [68] C. Wang, J. W. Paisley, and D. M. Blei, "Online variational inference for the hierarchical dirichlet process," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 752–760.
- [69] O. Lindemann, P. Stenneken, H. T. Van Schie, and H. Bekkering, "Semantic activation in action planning," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 32, no. 3, p. 633, 2006.
- [70] M. van Elk, H. van Schie, and H. Bekkering, "Action semantics: a unifying conceptual framework for the selective use of multimodal and modality-specific object knowledge," *Physics of life reviews*, 2013.
- [71] S. H. Creem and D. R. Proffitt, "Grasping objects by their handles: a necessary interaction between cognition and action," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 27, no. 1, p. 218, 2001.
- [72] S. L. Friedman and E. K. Scholnick, *The developmental psychology of planning: Why, how, and when do we plan?* Psychology Press, 2014.
- [73] N. L. Stein and T. Trabasso, "What's in a story: An approach to comprehension and instruction," *R. Glaser (Ed.), Advances in instructional psychology*, pp. 213–267, 1981.
- [74] N. Cowan, *Working memory capacity*. Psychology Press, 2004.
- [75] J. D. Gabrieli, R. A. Poldrack, and J. E. Desmond, "The role of left prefrontal cortex in language and memory," *Proc. of the Nat. Acad. of Sci.*, vol. 95, no. 3, pp. 906–913, 1998.
- [76] G. C. Oden, "Concept, knowledge, and thought," *Annual Review of Psychology*, vol. 38, no. 1, pp. 203–227, 1987.
- [77] U. Hahn and N. Chater, "Concepts and similarity," *Knowledge, concepts and categories*, pp. 43–92, 1997.
- [78] J. Kim, "Concepts of supervenience," *Philosophy and Phenomenological Research*, pp. 153–176, 1984.
- [79] T. Deacon, "The symbolic species: the co-evolution of language and the human brain," 1997.
- [80] A. Klippel and D. R. Montello, "Linguistic and nonlinguistic turn direction concepts," in *Spatial information theory*. Springer, 2007, pp. 354–372.
- [81] S. Timpf, G. S. Volta, D. W. Pollock, and M. J. Egenhofer, "A conceptual model of wayfinding using multiple levels of abstraction," in *Theories and methods of spatio-temporal reasoning in geographic space*. Springer, 1992, pp. 348–367.
- [82] J. A. Hampton, "Conceptual combination," *Knowledge, concepts, and categories*, pp. 133–159, 1997.
- [83] A. H. Hastorf and H. Cantril, "They saw a game; a case study," *The Journal of Abnormal and Social Psychology*, vol. 49, no. 1, p. 129, 1954.
- [84] T. Bylander, "Complexity results for planning," in *IJCAI*, vol. 10, 1991, pp. 274–279.
- [85] D. Chapman, "Planning for conjunctive goals," *Artificial intelligence*, vol. 32, no. 3, pp. 333–377, 1987.
- [86] J. A. Hendler, A. Tate, and M. Drummond, "Ai planning: Systems and techniques," *AI magazine*, vol. 11, no. 2, p. 61, 1990.
- [87] Z. Chen, R. S. Siegler, and M. W. Daehler, "Across the great divide: Bridging the gap between understanding of toddlers

and older childrens thinking,” *Monographs of the Society for Research in Child Development*, vol. 65, no. 2, 2000.

- [88] A. K. Qin and P. N. Suganthan, “Robust growing neural gas algorithm with application in cluster analysis,” *Neural Networks*, vol. 17, no. 8-9, pp. 1135–1148, 2004.



Hande Çelikkanat is currently pursuing her Ph.D. degree in cognitive and developmental robotics in KOVAN Lab., Department of Computer Engineering, Middle East Technical University. She also holds B.Sc. and M.Sc. degrees from the same department, with the M.Sc. thesis titled *Control of a Mobile Robot Swarm via Informed Robots*. Her research interests include the neurological and psychological bases of cognition, especially as related to the development of

language, and modeling of these in robots.



Güner Orhan received a B.Sc. degree in the Department of Computer Engineering, Middle East Technical University, 2012. He completed his M.Sc. degree in KOVAN Research Lab., Department of Computer Engineering, METU, 2014. The title of the thesis is “Building a Web of Concepts on a Humanoid Robot”. He is currently pursuing PhD. in the Faculty of Electrical Engineering, Mathematics and Computer Science, Formal Methods and Tools group at the University of Twente, the Netherlands. His research interests are Developmental Robotics, Cognitive Robotics, Parallel Programming, Computer Vision, Image Processing, Schedulers and Software Product Line Engineering.



Nicolas Pugeault received the M.Sc. from the University of Plymouth, Plymouth, U.K., in 2002, the engineering degree from the Ecole Supérieure d’Informatique, électronique, Automatique, Paris, France, in 2004, and the Ph.D. degree from the University of Göttingen, Göttingen, Germany in 2008. He is currently a lecturer at the College of Engineering, Mathematics and Physical Sciences, at the University of Exeter, Exeter, U.K. His research interests include cognitive systems,

machine learning, and computer vision.



Frank Guerin obtained his Ph.D. degree from Imperial College, London, in 2002. Since August 2003, he has been a Lecturer in Computing Science at the University of Aberdeen. He is interested in infant sensorimotor development, especially meansend behaviour and precursors to tool use. Dr. Guerin is a member of The Society for the Study of Artificial Intelligence and Simulation of Behaviour, where he has served as a committee member and co-chair of the annual

convention.



Erol Şahin has a B.Sc. in Electrical and Electronics Engineering from Bilkent University, Turkey, in 1991, an M.Sc. in Computer Engineering from Middle East Technical University (METU) in 1995, and a Ph.D. in Cognitive and Neural Systems from Boston University, USA, in 2000. He is working as an Assistant Professor at the Dept. of Computer Engineering at METU and is heading the KOVAN Research Laboratory. His work on cognitive systems focuses on how the

notion of affordances can be used at different levels of autonomous robot control, and how the notion can be linked to mirror and canonical neurons for developing concepts that correspond to verbs and nouns in language (through the ROSSI project). In 2009, he has been awarded one of the free iCub humanoid robot platforms by the RobotCub project, to carry on his research on the topic. Dr. Sahin has also been working on swarm robotics, and has edited two books and two special issues on the topic.



Sinan Kalkan received his M.Sc. degree in Computer Engineering from Middle East Technical University, Turkey in 2003, and his Ph.D. degree in Informatics from the University of Göttingen, Germany in 2008. After working as a postdoctoral researcher at the University of Göttingen and at Middle East Technical University, he is an assistant professor at Middle East Technical University since 2010. Sinan Kalkan’s research interests include biologically motivated Computer Vision and Image Processing, and Developmental Robotics.